Center for Computational Engineering Science Mathematics Division

RWTH AACHEN UNIVERSITY

Diese Arbeit wurde vorgelegt am
Lehrstuhl für Mathematik (MathCCES)

Beschleunigung numerischer Berechnung der Grundzustandsfunktion einer durch einen Laser angeregten Kette von Rydbergatomen

Accelaration of numerical calculation of the ground state of an by laser excited chain of rydberg atoms

Bachelorarbeit
Physik

November 2021

Vorgelegt von Presented by	Philipp Rosendahl Roermonder Straße 112a Matrikelnummer: 378092 philipp.rosendahl@rwth-aachen.de
Erstprüfer First examiner	Prof. Dr. Benjamin Stamm Lehrstuhl für Mathematik (MathCCES) RWTH Aachen University
Zweitprüfer Second examiner	Prof. Dr. Stefan Weßel Institut für theoretische Festkörperphysik RWTH Aachen University
Koreferent Co-supervisor	Dr. Michael Herbst Lehrstuhl für Mathematik (MathCCES) RWTH Aachen University

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Stellen meiner Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht. Dasselbe gilt sinngemäß für Tabellen und Abbildungen. Diese Arbeit hat in dieser oder einer ähnlichen Form noch nicht im Rahmen einer anderen Prüfung vorgelegen.

Aachen, im November 2021

Philipp Rosendahl

Contents

List of Figures

List of Tables

1. Abstract

This thesis discusses an algorithm to calculate the ground state of a chain of rydberg atom coherently excited by laser. The Hamiltonian solved is parameterized by two parameters. The first parameter is setting out the coupling strength of the atoms and the other sets out the deviation of the laser's frequency from the resonance of one Rabi oscillator modelling the "natural" oszialltion between the excited and ground state of one atom. To reduce the complexity of the calculation we use a reduced basis approach. We are able to calculate ground states of a chain with up to 17 and depicted the arrangement of excited and unexcited atoms on 200×200 parameter grid. We reproduced that the ground states are made up by specific translationally symmetric wave functions up to $\approx 85\%$ as found by Nelles [18]. This calculation took 11442 seconds by maintaining almost exact eigenvectors [1]. Our calculation is as fast as the algorithm developed by Nelles, but much more accurate.

2. Introduction

Artificial quantum systems are used to build hardware platforms for quantum computing. A good experimentalist can control the parameterised Hamiltonian to prepare a quantum state needed for an arbitrary calculation. Therefore one needs to know how the system behaves on a specific parameter set and its surrounding to have a reliable way of preparing the machine for the calculation. Rydberg atoms for example have already been used to simulate Ising Models and some other two level systems following fermionic statistics [18]. The study of artificial quantum systems is a relatively new field of research, because it was not of interest so far to build a system with adaptable external parameters, because those are needed for the new field of quantum technology.

It is crucial to have a method of calculating the groundstate of a system very accurately. Good approximations allow one to analyse complex systems without having to consider every detail of the system, but it is always important to compare ones approximations to a more realistic model to check if and where the approximation fails. This is needed to estimate the order of magnitude neglected by the approximations to gain insight into the validity of those. Our motivation is to find an algorithm that provides accurate results in an efficient algorithm to validate Nelles approximation and lay the foundation for the validation of other approximation made in this context.

Current state of solid state research normally is limited by the shape of the Hamiltonian. Mahler [16] for example was able to calculate the magnetisation of a grid of 48×48 spins by using stochastic row expansion [10]. Unfortunately this is no purposeful ansatz for us, because our system can be considered being frustrated. Frustration in this context refers to signs of the permutation under which the coupling is symmetric. Because in the chain of Rydberg atoms all atoms couple with each other the even and uneven permutations are needed to describe the symmetry, so stochastic row expansion will fail in this system.

[1] limited by the precision of LOBPCG with 200 iterations

The typical ansatz, if one does not exploit the structure of the system, is to choose a reduced basis and solve the Hamiltonian projected into the subspace and use the Ritz vectors as approximation. Our proposal is to use such Ritz vectors as first guess for an algorithm capable to exploit the structure of the Hamiltonian. This results in an efficient algorithm having relatively accurate results.

Let us now draw the attention to the system we are analysing shorty before lining out the structure of the thesis. Rydberg atoms are atoms with an highly excited electron with a large quantum number n. Those atoms are really big with a diameter of about 1mm [11]. Because of the highly excited state the electron can be considered as far away from the nucleus and that results in a electric dipole. Those dipoles couple via Van Der Vaals interaction. The system consists of a few of those atoms in a chain. The system will be explained in detail in section 4.

To present our research we will set the mathematical foundation needed to understand the algorithms used first. After that we will introduce one into the physical model of a chain of rydberg atoms. Then we will give a little introduction to the LOBPCG and Lanczos algorithm. Having set the foundations we will elaborate our algorithm. After that we will compare our work to the results of Nelles.

3. Mathematical Modelling

3.1. Vector Spaces, Linear Maps, Dual space

Definition 1 (Vector Space)**.** *Let $\mathbb{F}$ be field a set $(\mathcal{V}, + : \mathcal{V} \times \mathbb{V} \to \mathcal{V}, \cdot : \mathcal{F} \times \mathcal{V} \to \mathcal{V})$ is called a $\mathbb{F}$-Vector Space if*

$$\forall a, b \in \mathbb{F} \text{ and } \forall \left|v\right\rangle, \left|w\right\rangle, \left|z\right\rangle \in \mathcal{V} \tag{1}$$
$$\left|v\right\rangle + \left|w\right\rangle \in \left|V\right\rangle \tag{2}$$
$$\left|v\right\rangle + \left|w\right\rangle = \left|w\right\rangle + \left|v\right\rangle \tag{3}$$
$$(\left|v\right\rangle + \left|w\right\rangle) + \left|z\right\rangle = \left|v\right\rangle + (\left|w\right\rangle + \left|z\right\rangle) \tag{4}$$
$$\forall \left|v\right\rangle \exists \left|u\right\rangle : \left|v\right\rangle + \left|u\right\rangle = \left|0\right\rangle \in \mathcal{V} \tag{5}$$
$$\left|v\right\rangle + \left|0\right\rangle = \left|v\right\rangle \tag{6}$$
$$a \cdot \left|v\right\rangle \in \mathcal{V} \tag{7}$$
$$a \cdot (\left|v\right\rangle + \left|w\right\rangle) = a \cdot \left|v\right\rangle + b \cdot \left|w\right\rangle \tag{8}$$
$$a \cdot (b \cdot \left|v\right\rangle) = (ab) \cdot \left|v\right\rangle \tag{9}$$
$$1 \cdot \left|v\right\rangle = \left|v\right\rangle \tag{10}$$

Elements from a vector space are called vectors.

Definition 2 (Linear Independence)**.** *Let $\mathcal{V}$ be a $\mathbb{F}$-Vector Space. Then the vectors $\left|v_1\right\rangle, ..., \left|v_n\right\rangle \in \mathcal{V}$ are called linear independent, if*

$$\text{Let } \lambda_1, ..., \lambda_n \in \mathbb{F} \tag{11}$$
$$\lambda_1 \cdot \left|v_1\right\rangle + ... + \lambda_i \cdot \left|v_i\right\rangle + ... + \lambda_n \cdot \left|v_n\right\rangle = \left|0\right\rangle \tag{12}$$

$$\implies \lambda_1 = \lambda_2 = ... = \lambda_i = ... = \lambda_n = 0 \tag{13}$$

Having set the basic definition of vector spaces and linear independence we want to introduce the bases. They are important because they allow to write down a representation of a vector and do some calculation. A vector is not an array of numbers, but an element of vector a vector space that can be written as an array of number to a specific basis.

Definition 3 (Basis)**.** *Let $\mathcal{V}$ be a $\mathbb{F}$-Vector Space. A set $B = \{ |b_1\rangle, ..., |b_n\rangle \} \in \mathcal{V}$ is called basis, if $\forall |v\rangle \in \mathcal{V} \exists \lambda_1, ..., \lambda_n \in \mathbb{F} : \sum_{i=1}^{n} \lambda_n * |b_n\rangle = |v\rangle$ and B linear independent.*

Theorem 1 (Steinitz Exchange Lemma [17])**.** *Let $\mathcal{V}$ be a $\mathbb{F}$-Vector Space and let B be a basis of $\mathcal{V}$ and $Y = (|y_1\rangle, ..., |y_s\rangle) \in \mathcal{V}^s$ linear independent. Then $s \leq d = \#B$ and after proper rearranging of B $(|y_1\rangle, ..., |y_s\rangle, |b_{s+1}\rangle, ..., |b_d\rangle)$ is a basis of V.*

Later this theorem will be used to extend a useful base of a Vector Subspace to a base of the original Vector Space. As one will read in 3.3 one can select a Vector Subspace with the most physical important behaviour and compare it to the full space. Therefore ones needs to know, that the Vector Space does not change, if one uses a different basis.

Definition 4 (Dimensions)**.** *Let $\mathcal{V}$ be a $\mathbb{F}$-Vector Space with base B then the cardinal number $d = \#B$ is called the dimension of $\mathcal{V}$.*

The dimension of the a vector space is needed, because the complexity of later calculations depend on it. We also need to know if the dimensions of a vector space are well defined so we know that we cannot reduce complexity by choosing another base.

Corollary 1 (Dimensions are well defined)**.** *Let $\mathcal{V}$ be a $\mathbb{F}$-Vector Space with two different dimensions and bases $d = \#B$, $d' = \#B'$, w.o.l.g $d' > d$ then $\forall |b'\rangle \in B'$ is a linear combination of B.*

$$\forall \lambda'_i : |0\rangle \neq \sum_{i=1}^{d'} \lambda'_i \cdot |b\rangle'_i \tag{14}$$

$$= \lambda'_{d'} \cdot |b'_{d'}\rangle + \sum_{i=1}^{d'-1} \sum_{j=1}^{d} \lambda'_i \cdot \lambda_j \cdot |b_j\rangle \tag{15}$$

$$\text{define } \tilde{\lambda}_j = \lambda_j \sum_{i=1}^{d'-1} \lambda'_i \tag{16}$$

$$= \lambda'_{d'} \cdot |b'_{d'}\rangle + \sum_{j=1}^{d} \tilde{\lambda_j} \cdot |b_j\rangle \tag{17}$$

Choose $\lambda'_{d'} = 1$ Hence B is a base, there exists $\tilde{\lambda}_j$ so that $\sum_{j=1}^{d} \tilde{\lambda_j} \cdot b_j = -b'_{d'}$. So d' is no dimension of V, because B' is not a basis.

In quantum physics operators are vector endomorphisms over a Hilbert space. In easy words linear maps from a Hilbert space to itself. That is why we want to draw the attention to basic properties of linear maps. The fact that all of quantum mechanics used in this thesis is a eigenproblem 59 with linear maps is the reason why we recapitulate linear maps.

Definition 5 (Linear Map)**.** *Let $\mathcal{V}, \mathcal{W}$ be $\mathbb{F}$-Vector Spaces, the a map $M : \mathcal{V} \to \mathcal{W}$ is called linear, if $\forall a \in \mathbb{F}, |v\rangle, |(\rangle u) \in \mathcal{V} : M(a \cdot |v\rangle + |u\rangle) = a \cdot M(|v\rangle) + M(|u\rangle)$*

Corollary 2 (The Set $M_{\mathcal{V}\to\mathcal{W}} = \{ \mathcal{V} \to \mathcal{W} \mid linear \}$ is a $\mathbb{F}$-Vector Space)**.** *Let $\mathcal{V}, \mathcal{W}$ be $\mathbb{F}$-Vector Spaces, then define the $(M_{\mathcal{V}\to\mathcal{W}}, +, \cdot)$*

$$\forall a \in \mathbb{F}, |v\rangle \in \mathcal{V}, \phi, \eta \in M_{V\to W} \tag{18}$$
$$(\phi + \eta)(|v\rangle) = \phi(|v\rangle) + \eta(|v\rangle) \tag{19}$$
$$(a \cdot \phi)(|v\rangle) = a \cdot \phi(|v\rangle) \tag{20}$$

Definition 6 (Dual Space)**.** *Let $\mathcal{V}$ be a $\mathbb{F}$-Vectorspace, then the $\mathcal{V}^* = \{ \mathcal{V} \to \mathbb{F} \mid linear \}$ is called dual space. $\mathcal{V}^*$ also is a $\mathbb{F}$-Vectorspace with the same dimension, because the vector space can be constructed similar like the scalar multiplication and vector addition in endomorphism rings.*

Definition 7 (Hermitian sesquilinear maps)**.** *Let $\mathcal{V}$ be a $\mathbb{C}$-vector space, then the linear map Φ is called sesquilinear if*

$$\forall a, b \in \mathbb{C}, |v\rangle, |w\rangle, |u\rangle \in \mathcal{V}$$
$$\Phi(|v\rangle, a|w\rangle + b|u\rangle) = a\Phi(|v\rangle, |w\rangle) + b\Phi(|v\rangle, |u\rangle)$$
$$\Phi(a|v\rangle + b|u\rangle, |w\rangle) = \bar{a}\Phi(|v\rangle, |w\rangle) + \bar{b}\Phi(|u\rangle, |w\rangle)$$

A sesquilinear map is called hermitian, if $\Phi(|v\rangle, |w\rangle) = \overline{\Phi|w\rangle, |v\rangle}$

For quantummechanics one formulate the states using a complex Hilbert space. So one has a vector space with a norm induced by hermitian sesqulinear map Φ. For using the Dirac formalism one calls endormorphisms operators and use the Braket notation to tell vectors and dual vectors apart. Therefore the dual vectors are chosen that the dual vector $\langle\psi'|$ applied on the vector $|\psi\rangle$ is the same like $\Phi(|\psi'\rangle, |\psi\rangle) = |\psi|^2$.

Corollary 3 (For a given $|\psi\rangle$ exists a well defined $\langle\psi|$)**.** *The existence of the $\langle\psi|$ can be proven by using the scalar product of the Hilbert space by using checking linearity of the functional $\langle v|(w) = \int_H \bar{v}(x)w(x)dh$ which is certainly given. Let $\langle\psi|, \langle\psi|'$ are two different dual vectors*

$$\langle\psi|\psi\rangle = \langle\psi|'|\psi\rangle = \Phi(\langle\psi|, \langle\psi|) \tag{21}$$
$$\implies \langle\psi|\psi\rangle - \langle\psi|'|\psi\rangle = (\langle\psi| - \langle\psi|')|\psi\rangle = 0 \tag{22}$$
$$\implies |\psi\rangle \in Core(\langle\psi| - \langle\psi|') or \langle\psi| - \langle\psi|' = 0 \tag{23}$$

Now we will take a closer look to the first case, because the second case is contradictory to the assumption, that $\langle\psi|$ and $\langle\psi|'$ are different. Now considering the first so case: The term decays into one term covered by the second case (the parallel parts) and the orthogonal parts which are definitely fulfil the first condition. Those orthogonal parts need to be zero: choose $|\phi\rangle$ with $\Phi(|\phi\rangle, |\psi\rangle) = 0$. Then the existing dual vector $\langle\psi|$ added to $\langle\psi|$ $\langle\psi|^p rime = \langle\psi| + \langle\phi|$ would fulfil the first condition, but there is a contradiction considering

$$0 = \Phi(|\psi\rangle', |\phi\rangle) = \langle\psi|' |\phi\rangle = \langle\psi| |\phi\rangle + \langle\phi| |\phi\rangle = |\phi|^2 \tag{24}$$

So the $\langle\psi|$ is well defined.

Definition 8 (Vector Subspace)**.** *Let $(\mathbb{V}, +, \cdot)$ be a $\mathbb{F}$-Vector Space. Then $(\mathcal{U} \subset \mathcal{V}, +, \cdot)$ is called a Vector Subspace, if $(\mathcal{U}, +, \cdot)$ is a $\mathbb{F}$-Vector Space on its own.*

So one has to check if all axioms of a Vector Space are fulfilled. The easiest way to construct a Vector Subspace $U \subset V$ is to chose $n < d$ linear independent vectors $v \in V$ and create all linear combinations.

Definition 9 (Projector)**.** *A projector is linear map $\pi : \mathcal{V} -> \mathcal{V}$ with $\pi^2 = \pi$. With $\mathcal{V}$ being a $\mathbb{F}$-vector space.*

Corollary 4 (Domain of projector is a vector subspace)**.** *The properties of a vector space are inherited by the properties of linear maps. Those can be checked be individually.*

The next thing we need to know to proof the formalism of Ritz's method later 3.2 is that, if $\mathcal{U} \subset \mathcal{V}$ are $\mathbb{F}$-vector spaces and $|u_i\rangle$ is basis of $\mathcal{U}$,

Corollary 5 ($\sum_i |u_i\rangle \langle u_i|$ is a projector)**.**

$$(\sum |u_i\rangle \langle u_i|)^2 := \pi^2 = \sum_i \sum_j |u_i\rangle \langle u_i| |u_j\rangle \langle u_j| \tag{25}$$

$$= \sum_i \sum_j |u_i\rangle (\langle u_i| |u_j\rangle) \langle u_j| \tag{26}$$

$$= \sum_i \sum_j |u_i\rangle (\delta_{i,j}) \langle u_j| \tag{27}$$

$$= \sum |u_i\rangle \langle u_i| = \pi \tag{28}$$

Since we know how to construct a projector to an arbitrary subspace, we know how to project a vector into such a vector subspace. Moreover the a vector $|\Psi\rangle = \lambda |u_i\rangle$ can be interpreted as a vector in the subspace $\mathcal{U}$ with u_i as basis, or as element of $\mathcal{V}$ if one interprets $|u_i\rangle$ as a set of linear independent vectors of the vector space $\mathcal{V}$. Those interpretations do not contradict themselves. They are just different representation of the same vectors. The difference of those concepts will be more intuitive after introducing the Ritz-method in the next chapter 3.2. We can project from the vector space $\mathcal{V}$ in the subspace $\mathcal{U} \subset \mathcal{V}$ by

$$|u\rangle = U |v\rangle = (|u_1\rangle, ... |u_m\rangle) |u\rangle \tag{29}$$

3.2. Ritz method

Ritz's method for solving the eigenvalue problem 59 is following. One selects a subspace $\mathcal{U}$ of approximations reducing the residue . The Ritz method is the general of reduced basis approach [2].

$$r(\theta, |x\rangle) = A\,|x\rangle - \theta\,|x\rangle \tag{30}$$

being the same like the eigen problem 59 but without the requirement of being equal to zero. Then we choose an orthonormal basis $Q_n = (q_1, ..., q_n)$ and project the matrix A into $\mathbb{L}$ by using.

$$B_n = Q_n^\dagger A Q_n \tag{31}$$

$$B_n\,|s\rangle = \theta\,|s\rangle \tag{32}$$

Stretching the arc back to 4 one will realise, that the $|s\rangle$ shares similarities with the full eigenproblem 59. If one chooses a special subspace like one generated by a singular value dcomposition 10 or a Krylov subspace 67, one gets either a good approximation of $|s\rangle$ in general or of $|s\rangle$ being one eigenvector. The case of Krylov subspaces will be used later by the Lanczos algorithm 0.

3.3. Singular Value Decomposition

Definition 10 (Singular Value Decomposition)**.** *Let $A \in \mathbb{R}^{n,m}$ then there exists a decomposition $A = U \cdot \Sigma \cdot V^\dagger$ with $U \in \mathbb{R}^{n,n}, \mathcal{V} \in \mathbb{R}^{m,m}$ unitary, $\Sigma \in \mathbb{R}^{n,m}$ diagonal with entries greater or equal 0.*

First we will check, if the decomposition exists, so we will construct the decomposition. Referring to [14] the Matrix U needs to be symbolically:
$U = (\ \text{Column Space of } A \mid \text{Core of } A^\dagger\ )$ If the column space of A is not empty, let the first singular value be $\sigma_1 = |A|$. So there exists a vector $|u_1\rangle$ with $|\,|u_1\rangle\,| = 1$ and $|A\,|u_1\rangle\,| = \sigma_1$. Then for every $i \in \{\,2, ..., \text{column rank of A}\,\}$ construct a subspace O_i orthogonal on $Span\,\{\,|u_1\rangle\,, ..., |u_{i-1}\rangle\,\}$. Then choose $\sigma_i = |\,|O_i\rangle\,\langle O_i|\,A\,|O_i\rangle\,\langle O_i|\,|$. Then $|u_i\rangle$ is constructed like $|u_1\rangle$. Then construct a full orthonormal base by extending $Span\,\{\,|u_1\rangle\,, ..., |u_{\text{column rank A}}\rangle\,\}$ with the core of $A^\dagger$ with corresponding singular values 0. Because the matrix norm is well-defined the singular values σ_i are also well defined. The column vectors in the unitary matrix U are not well-defined, because the singular values are not necessarily different in pairs like the degeneration of eigenspaces. If A has full column rank and the core of $A^\dagger$ is empty and all if all singular values are pairwise different, also the matrix U is well-defined. So we have to construct the matrix V. Therefore look at the definition [3]

$$A = U \cdot \Sigma \cdot V^\dagger \tag{33}$$

$$\implies U^\dagger A = \Sigma V^\dagger \tag{34}$$

$$\implies \forall i \in \{\,1, ..., m\,\}\,(U^\dagger A)_{i,\_} = (\Sigma V^\dagger)_{i,\_} \tag{35}$$

[2] see chapter 3.3
[3] With $A_{i,\_}$ we mean the the i-th row of A

$$\implies (U^\dagger A)_{i,\_} = \sigma_i V^\dagger_{i,\_} \tag{36}$$

If σ_i bigger than 0 the row vector $|v_i\rangle = V^\dagger_{i,\_}$ is well defined. Check orthonormality for all $|v_i\rangle, |v_j\rangle$ with $i, j < rank(A)$, defined above.

$$\langle v_i, v_j\rangle = \frac{1}{\sigma_i\sigma_j}\langle (A^\dagger U)_{\_,i}, (A^\dagger U)_{\_,j}\rangle \tag{37}$$

$$= \frac{1}{\sigma_i\sigma_j}\langle A^\dagger u_i, A^\dagger u_j\rangle \tag{38}$$

$$\text{If } i = j \ = \frac{|A^\dagger u_i|^2}{\sigma_i^2} = \frac{\sigma_i^2}{\sigma_i^2} = 1 \tag{39}$$

$$\tag{40}$$

Because $\forall i, j < rank(A)$ are not in the core of A, A needs to bijective in this subspace. Lets proof that orthogonality is conserved on a set of vectors when an bijective map is applied. Therefore let W be an C vector space with orthonormal base e_i and B an bijective W endomorphism.

$$\sum_i \lambda_i B\,|e_i\rangle = 0 \tag{41}$$

$$\implies B(\sum_i \lambda_i\,|e_i\rangle) = 0 \tag{42}$$

$$\text{because the core of } B = \{\,0\,\} \text{ and } |e_i\rangle \text{ are orthogonal} \tag{43}$$

$$\implies \forall i \lambda_i = 0 \tag{44}$$

So if i and j are not equal, $\langle v_i, v_j\rangle = 0$. So the matrix V needs to be unitary.

3.4. Reduced Basis Approach

First of all we have to state, that one can express a linear map between finite vector spaces as a matrix. Therefore one can need to express the vector spaces as linear combination of their bases. Then one can identify the vectors as column array of field elements equal to the factors before the base elements. If one wants express a linear map $L : \mathcal{W} \to \mathcal{V}; \mathcal{W}, \mathcal{V}\mathbb{F} - \text{Vector space with basis } v_i, w_i$ as matrix one only has to write the matrix $L(w_1), ..., (L_i), ..., L_{dim(W)}$. Now we want to approximate a linear map or matrix with another map between Vector spaces of lower dimension to reduce computation time and warrant the best approximation of that map. In that context the residue norm should be as small as possible. So let $\mathcal{V}$ be $\mathbb{R}$ Hilbert space and H be an vector space endomorphism from $\mathcal{V}$ into itself with $dim_\mathcal{V}$ eigenvectors $H\,|e_i\rangle = \lambda_i\,|e_i\rangle$. Then we want to find a Subvector space $\mathcal{U} \subset \mathcal{V}$, with orthonormal basises $\{\,|u_i\rangle\,\} \subset \{\,|v_i\rangle\,\}$ so that for every eigenvector $|e_i\rangle$ the residue norm $Max_i|\,|e_i\rangle - \sum_j |u_j\rangle\,\langle u_j|\,|e_i\rangle\,|$ is minimal. The assumption is that that is true, if the $|u_j\rangle\,\forall j \in \{\,1...dim_(U)\,\}$) are equal to the first $dim_\mathcal{U}$ column vectors of the singular value decomposition $USV = H$. For the actual computation the Hamiltonian will be represented in diagonal form. Considering

the research of the stability and symmetry of the ground state of the coupled array of Rydberg atoms, because those vectors for the reduced basis are required anyways.

Best approximation by using reduced base by singular value decomposition.

$$Max_i|e_i - \sum_j |u_j\rangle \langle u_j| e_i| \tag{45}$$

$$= Max_i \frac{1}{\lambda_j}|He_i - \sum_j |u_j\rangle \langle u_j| He_i| \tag{46}$$

$$= Max_i \frac{1}{\lambda_j}|He_i - \sum_j |u_j\rangle \langle u_j| He_i| \tag{47}$$

$$= Max_i \frac{1}{\lambda_j}|USVe_i - \sum_j |u_j\rangle \langle u_j| He_i| \tag{48}$$

$$= Max_i \frac{1}{\lambda_j}|\sum_{j>dim_U} |u_j\rangle \langle u_j| SVe_i| \tag{49}$$

$$> Max_i \frac{1}{\lambda_j}|\sum_{j>dim_U} |u_j\rangle \langle u_j| \,||\sum_{j>dim_U} S_{j,j}||V||e_i| \tag{50}$$

Now considering the construction of the Singular value decomposition one can see easily, that the approximation is fulfils the demand of the approximation □

Considering the fact that the research question deals with different operators which are no linear maps for the parameter set $\frac{\Delta}{\Omega}$, n_s we have to have justify the approximation for the use case. If investigate this parameter set via perturbation theory we will find, that that approximation works, because any point on the plane can be described by a linear combination of the operators nearby. So we now know, that a reduced base approach via singular value decomposition works quite well for only one operator and for an operator field like here, we can use the approximation, by simply find an base vector like by ignoring the fact we are using different operators. This will of course mess up the behaviour of the residue norm.

3.5. Hilbert spaces

A Hilbert space is a $\mathbb{C}$-vector space with a sesquilinear form ϕ used as skalar product $\forall\, |v\rangle, |w\rangle \in$ Hilbert space$\phi(|v\rangle, |w\rangle) = \langle v, w\rangle$ positive definite and $||v\rangle| = \sqrt{\langle v, v\rangle}$. Moreover a Hilbert space is closed referencing the norm $|\cdot|$, meaning if a sequence converges the limit is element of the Hilbert space.

3.6. Tensor Product Spaces

Tensor product spaces are a relatively complex topic whole libraries can be filled with that topic. So we will focus on the relevant parts for this work. In many particles physics one uses tensor spaces product spaces over the single particle Hilbert spaces.

There are many tensor products that are all isomorph to them, because of the universal property of tensor product spaces. One tensor product over finite vector spaces is the Kronecker product.

Definition 11. *Kronecker Product*

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \otimes \begin{pmatrix} E & F \end{pmatrix} = \begin{pmatrix} AE & BE & AF & BF \\ CE & DE & CF & DF \end{pmatrix} \tag{51}$$

Like Nelles we will the Kronecker product as tensor product for the calculation. The Kornecker product has a useful structure to extract the dominant part of the spectrum (see paragraph 6.1.3)

4. Chain of N Coupled Rydberg Atoms

4.1. Physical Foundation

For the Hamiltionian we will refer to Nelles [18]. Rydberg states are highly excited electronic states. Hence the energy of one rydberg state is approximately $E_n = R_y/n^2$ [11] with very big n. Considering Fermis Golden Law [19] which says, that two states n_1, n_2 need a transition time of about $t \approx \frac{h}{E_{n_2} - E_{n_1}} \approx \frac{hR_y}{n_2^2 - n_1^2}$. That means, that the transition time increases for a bigger difference between atomic states. So for a high excitation the election oscillates quickly back and forth around the expected value $R_y + \Delta$. The transition between the excited and ground state is almost is because of the very big expected n almost continuous. So Nelles used Rabi-Oscillators is a two niveau system with an excitation and time independent Schrödinger-Equation:

$$\begin{pmatrix} i\dot{a}_1 \\ \dot{a}_2 \end{pmatrix} = \begin{pmatrix} 0 & \frac{\Omega^*}{2} \\ \frac{\Omega}{2} & -\Delta \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \frac{2\pi}{h} \hat{H} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \frac{2\pi}{h} \left(\frac{\Omega}{2} \sigma_x - \Delta \hat{n} \right) \tag{52}$$

$$\text{with } \sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ and } \hat{n} := \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{53}$$

Where a_1 is ground state of the atom and a_2 the niveau of the approximated rydberg state. Now we will consider an array of rydberg atoms. Like always the space of the array with n atoms is just a Tensor Product Space that is generated by the span of n times the single particle Space. The interaction between those atoms is though the van-der-Vaals coupling so the strength decays by $\frac{C}{distance^6}$ where C is the coupling strength. Using $hatV$ is the interaction of the atoms is given by the two particle operator in $\hat{V}\hat{n}_i\hat{n}_j$ in the summation over all i, j with $i \neq j$ there occur term like $\hat{V}(\hat{n}_i\hat{n}_j + \hat{n}_j\hat{n}_i)$. Luckily two occupation operators do commute so we only have to sum over $i > j$. So the hamiltionian for the whole system is given by [18].

$$\frac{2\pi}{\Omega} H = \frac{1}{2} \sum_i \sigma_{x,i} - \frac{\Delta}{\Omega} \sum_i \hat{n}_i + \sum_{i>j} \frac{n_s}{i-j}^6 \hat{n}_i \hat{n}_j \tag{54}$$

The notation $operator_i$ means that the i-th component of the tensor product state is the operator with all other factors being 0. One can understand n_s as the interaction strength between those atoms witch the distance of the atoms and coupling through van-der-Vaals already included and $\frac{\Delta}{\Omega}$ is set by the deviation of the frequency of the laser from the resonance frequency of the Rabi oscillator.

4.2. Structure of Operator parts

Now let us draw the attention to the structure of the Operators. One can see in figure 1 the sparsity pattern of the operators for a chain of eleven rydberg atoms. If we would present figures for other chains one would see similar sparsity patterns. For such sparse occupied matrices a sparse matrix data structure (chapter 5) would be purposeful. As

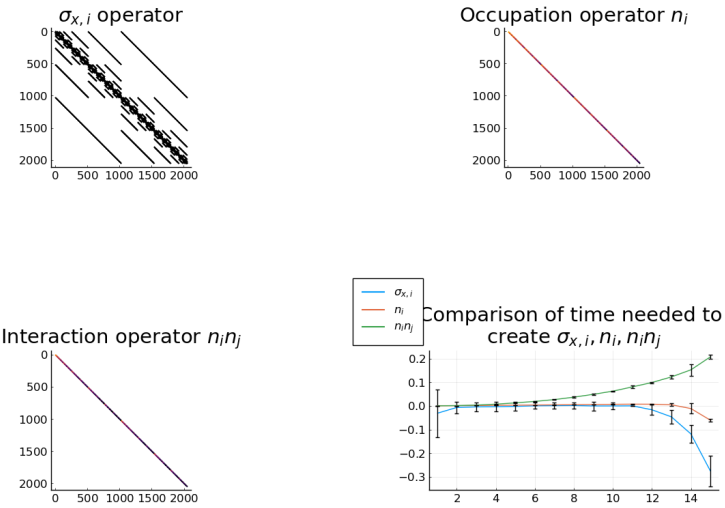

Figure 1: Spy Figures of operator parts for 11 atoms and time needed to create them for 1 to 14 Atoms compared to the time Nelles needed.

one sees in figure 1 the hamiltonians are either in diagonal or sparse shape. Considering the n_i operators for d atoms:

Theorem 2 (Shape of n_i)**.** *Let d be the number of coupled atoms, then $n_i \in \mathbb{C}^{2^d,2^d}$ in diagonal shape with 2^i blocks of the length 2^{d-i} and alternating diagonals with 0 and 1.*

Shape of n_i. To start the proof let $d = 2$ then

$$n_1 = \hat{n} \otimes \mathbb{1}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{55}$$

$$n_2 = \mathbb{1}_2 \otimes \hat{n} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{56}$$

To complete the proof by the induction principle consider the case $d+1$ and let theorem be fulfilled in the case d. For all n_i^{d+1} there exists a n_j^d so that either $\mathbb{1}_2 \otimes n_j^d$ or $n_j^d \otimes \mathbb{1}_2$ equals n_i^{d+1}. Then:

$$n_i^{d+1} = \mathbb{1}_2 \otimes n_j^d = \begin{pmatrix} 1n_i^d & 0 \\ 0 & 1n_i^d \end{pmatrix} \tag{57}$$

which is a matrix with in diagonal shape because n_i^d is in diagonal shape and has $2 \cdot 2^j$ blocks with length 2^{d-1-j} and alternating 0 and 1. In this case $j = i - 1$. This j exists for all $i > 1$ In the other case $i = 1$:

$$n_1^{d+1} = \hat{n} \otimes \mathbb{1}_{2^d} = \begin{pmatrix} 0\mathbb{1}_{2^d} & 0 \\ 0 & 1\mathbb{1}_{2^d} \end{pmatrix} \tag{58}$$

Which is diagonal matrix with 2 of length $2^d = 2^{d+1-i}$. □

Theorem 3 (shape of $\sigma_{x,i}$)**.** *The Matrix $\sigma_{x,i}$ consists of 2^{i-1} side diagonals of length $l_i = 2^{d-i}$. The upper side j-th diagonal begins at $2l(j-1)+1, l*(2j-1)+1$. A rule for the lower side diagonals can be deduced by transposing the indices from the upper side diagonal.Now we will discuss how to speed up the generation of the hamiltonian 54.*

The proof of that theorem is very similar to the proof 4.2 and will be skipped.

5. Iterative Methods

5.1. Sparse Matrices

Let us take a look to the sparse matrix data type first. We will need it later to create the Hamiltonians. Using Sparse Arrays format allows us to store the Hamiltonians without filling up the RAM with zeros. The CSC (compressed sparse column) format is superior to the CSR (compressed sparse row) format when a multiplication with a vector is performed, because the structure of data type is optimised for row slicing. And the Julia Programming Language used a column first format so we have to CSC anyways. Every row is stored as a pointer to a tuple of column indices and the actual non zero values.

Example 1 (CSC - compressed sparse column [8])**.** *In this example indices start by 1.*

$$M = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 0 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$
\begin{aligned}
Values &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix} \\
Columns &= \begin{pmatrix} 1 & 3 & 3 & 1 & 2 & 3 \end{pmatrix} \\
Row &= \begin{pmatrix} 1 & 3 & 4 \end{pmatrix}
\end{aligned}
$$

The number of the column is used to define the beginning of the compressed row indices. Naturally the number of the next row minus one marks the end of the column array. [20] To insert a value manually can be a very time consuming because it requires a lot copying. If one wanted to insert 7 with the index $1, 1$, the library used had to copy every value column pair to a higher part of memory to insert the new value. After that it had to increment the Row pointers. Compared to a single copy call needed for dense structure this a tremendous overhead. The reason we need to understand this structure is we need to fill in the row pointer and column value pairs manually avoid the extensive usage of copying.

For sparse matrices it is always useful to use iterative eigensolvers. Those normally only need a matrix free representation of the linear map, so they can be implemented without row or column slicing and bypass the drawback of CSC-structures.

5.2. Lanczos

The Lanczos [13] algorithm is one of the oldest iterative algorithms introduced 1950 by Cornelius Lanczos. The algorithm became popular with the usage of electronic computers. First we will introduce some preliminary concepts needed to understand the algorithm

Let $\mathcal{V}$ be an $\mathbb{R}$-vector space, $A : \mathcal{V} \to \mathcal{V}$ be an symmetric matrix of rank n and let $|x\rangle$ be an arbitrary vector. The eigenvector equation is given by

$$A|x\rangle - \lambda|x\rangle = 0. \tag{59}$$

One can see that this equation as solutions, only if $\chi_A(\lambda) = det(A - \lambda\mathbb{1}) = 0$. Krylov used the **Krylov** subspaces $\mathbb{K}^m = span\{|x\rangle, A|x\rangle, ..., A^{m-1}|x\rangle\}$ to have a useful subspace to solve the characteristic polynomial $\chi_A(\lambda)$. A Krylov subspace is a cyclic generated sub vector space. We will proceed now describing the Lanczos algorithm. Therefore we denote the eigenvalues $\lambda_1 < \lambda_2 < ... < \lambda_n$, $|e_1\rangle, |e_2\rangle, ..., |e_n\rangle, |e_i\rangle = A^i|x\rangle$ the unit vectors of a Krylov basis and $|q_1\rangle, ..., |q_n\rangle$ an orthonormal basis.

Theorem 4 (In the basis $|q_1\rangle, ..., |q_m\rangle$ the matrix A is in tridiagonal shape)**.** *Consider the matrix element*

$$\forall i, j \in 1, ..., m \, \langle q_i A q_j \rangle : \tag{60}$$

$$|q_i\rangle \in \mathbb{K}^i(|x\rangle), A|q_j\rangle \in \mathbb{K}^{j+1}(|x\rangle), |q_i\rangle \perp \mathbb{K}^{i-1}(|x\rangle) \quad \textit{Follows from orthonormal basis} \tag{61}$$

Since $A|q_j\rangle \in \mathbb{K}(j+1)(|x\rangle)$ *and we know that* $|q_{j+1}\rangle \perp \mathbb{K}^j$ *we know that* $\forall j \forall k > j+1 \, \langle q_k | A | q_j \rangle = 0$*. Because the matrix* A *is symmetric it needs to stay symmetric in the orthonormal basis* $|q_1\rangle, ..., |q_m\rangle$*. So the matrix* A *is tridiagonal in the orthonormal Krylov subspace. We call* $\langle q_i| A |q_{i+1}\rangle = \beta_i$ *and* $\langle q_i| A |q_i\rangle = \alpha_i$

Hence an orthonormal basis is required we need to take a close look at the vector $A\,|q_j\rangle$ and refine $|r_j\rangle$ to find an iterative way of constructing α_i, β_i.

$$A\,|q_j\rangle = \beta_{j-1}\,|q_{j-1}\rangle + \alpha\,|q_j\rangle + \beta_{j+1}\,|q_{j+1}\rangle \tag{62}$$

$$\text{and define} \quad |r_j\rangle := \beta_j\,|q_{j+1}\rangle = A\,|q_j\rangle - \beta_{j-1}\,|q_{j-1}\rangle - \alpha_j\,|q_j\rangle \tag{63}$$

Having refreshed the matrix elements we want to examine how to construct α_j, β_j, $|r_j\rangle$ and $|q_j\rangle$ if they have already been constructed for $i < j$. Using the Gram-Schmidt orthonormalisation method one directly gets:

$$|q_j\rangle = \frac{|r_{j-1}\rangle}{\beta_{j-1}} \quad \alpha_j = \langle q_j \,|\, A \,|\, q_j\rangle \quad \beta_j = \|r_j\| \tag{64}$$

To finish the elaboration we need to define the start values $|q_0\rangle := |0\rangle$, $|r_0\rangle = |x\rangle$ and $\beta_0 = \|x\|$.

To diagonalise a tridiagonal matrix is a relatively easy task. Until know the representation of linear map is not determined. One only needs to calculate matrix vector products and matrix elements. It is not required to do slicing or anything like this up to this moment. So a sparse representation of the matrix A or any numerical representation will suffice. For the orthogonalisation process methods like Gram-Schmidt can be used. To orthogonalise the tridiagonal matrix one either can use specialised algorithms for that.

So the Lanczos algorithm is the Ritz method with the an orthonormal Krylov sub space as initial guess. Let us know the attention the actual construction of α_i and β_i. We need this to for the iterative reconstruction of the Lanczos basis. So write some matrix elements and define the vector r_j derived from the tridiagonal shape mentioned above

To illustrate the Algorithm we want to draw attention to following pseudocode.

Algorithm 1 Lanczos algorithm

$x_0 \leftarrow$ Some initial guess
$A \leftarrow$ Symmetric Matrix
while $\sum_i^m \langle r(\theta_i, q_i), r(\theta_i, q_i)\rangle > tol$ **do** ▷ 3.2
  $Q \leftarrow \mathbb{K}_A^m(x_i)$ ▷ construction of orthonormal Krylov space
  $T \leftarrow Q^T A Q$ ▷ Triangular see 5.2
  $(\theta, x) \leftarrow eigen(T)$ ▷ Ritz process
end while

Because of the cyclic nature of Krylov subspace the biggest eigen pair is most probable to converge if the eigen vector is element of the Krylov subspace. To sum it up: Lanczos algorithm is the classical Ritz method used in a Krylov subspace to have a matrix in tridiagonal shape to be able to use specilised algorithms for that.

5.3. LOBPCG

To get an insight to the locally optimised block preconditioned continuous gradient (LOBPCG) algorithm we will follow the structure of the paper by Knyazev [15]. In a nutshell the LOBPCG is an efficient algorithm for solving sparse large symmetric generalised eigenvalue problems. It is not part of thesis discussing convergence, but in practice it delivers responsible results. This section is organised as follows. We will define a preconditioned Krylov subspaces and introduce Rayleigh quotient as preliminary theory for standard preconditioned conjugate gradient (PCG) method first, an algorithm able to calculate an eigenvector approximation if the associated eigenvalue is already known. Then we will add a method for finding the eigenvalue using local optimisation.

So let us draw the attention to the preliminary theory. Let A be a symmetric positive definite matrix. Then define $A - \lambda\mathbb{1}$ as (pencil). One will notice the similarities to the eigenproblem 59, because the kernel of the pencil is equivalent to the eigenproblem. Now introduce the object T called preconditioner. It should be chosen to be a approximation of A^{-1} such that $\exists\delta_1 \geq \delta_0 > 0$

$$\delta_0 \left\langle T^{-1}x, x \right\rangle \leq \langle x \,|\, A \,|\, x \rangle \leq \delta_1 \left\langle T^{-1}x, x \right\rangle . \tag{65}$$

Equation 65 defines if T is really an approximation to A^{-1}. Moreover one can estimate the convergence properties of the following algorithms by comparing the ratio $\frac{\delta_1}{\delta_0}$. Let us now define the **Rayleigh coefficient**. Therefore notice that all eigenpairs are in the kernel of the pencil so:

$$\langle x \,|\, A - \lambda\mathbb{1} \,|\, x \rangle = 0 \implies \langle x \,|\, A \,|\, x \rangle = \lambda \langle x \,|\, x \rangle \implies \lambda(x) = \frac{\langle x \,|\, A \,|\, x \rangle}{\langle x \,|\, x \rangle}. \tag{66}$$

As known the Rayleigh coefficient is always bigger than the smallest eigenvalue of the eigenproblem [12]. At last of preliminary theory define a preconditioned Krylov subspace the algorithm operates on.

$$\mathbb{K}_k(TA, T, x_0) = span\{P_k(TA, T)x_0\} \tag{67}$$

$$P_k(a, b) = \{a^i b^j | i + j = k\} \tag{68}$$

Now knowing the definition of a pencil, preconditioner, Rayleigh coefficient and the Krylov subspace, we can conclude the preliminary theory by stating that one could use a preconditioned Krylov space as starting point for defining a preconditioned Lanczos algorithm.

Now we want to draw the attention to the standard preconditioned conjugate gradient (PCG) to set the base to understand the local optimised PCG (LOPCG). Therefore suppose that the λ_i as smallest eigenvalue has been already found. This method is not useful in practice, because it requires the eigenvalue, we are interessted in. But we want to have a iterative method of minimising the seminorm as preliminary theory for the LOPCG.

$$\sqrt{\langle y_{j+1} \,|\, A - \lambda_1\mathbb{1} \,|\, y_{y+1} \rangle} \tag{69}$$

over the hyperplane

$$\mathbb{H}_{i+1} = y_0 + \mathbb{K}_i(T(A - \lambda_1 B), T(A - \lambda_1 \mathbb{1})y_0 \tag{70}$$

Therefore define $|x_1\rangle$ as the T^{-1} orthogonal projection of the initial guess $|y_0\rangle$ and $beta_0 = 0$ required in equation 5.3. Then one iteration step is given by

$$|y_{i+1}\rangle = |y_i\rangle + \alpha_i T(A - \lambda_1 B)\,|y_i\rangle + \beta_i(|y_i\rangle - |y_{i-1}\rangle) \tag{71}$$

α_i and β_i are scalars which are chosen to minimise the seminorm 69.

Now we introduce a method of locally optimal PCG for needing the knowledge of the eigen value anymore. Again we are looking for an iterative way of finding an extremal eigenvalue. Therefore we use the iteration step

$$|\omega_i\rangle = T(A - \lambda(|x_i\rangle)\mathbb{1})\,|x_i\rangle \tag{72}$$
$$|x_{i+1}\rangle = |\omega_i\rangle + \tau_i\,|x_i\rangle + \gamma_i\,|p_i\rangle \tag{73}$$
$$\gamma_0 = 0 \tag{74}$$
$$|p_0\rangle = |0\rangle \tag{75}$$
$$|p_{i+1}\rangle = |\omega_i\rangle + \gamma_i\,|p_i\rangle \tag{76}$$

with $\lambda(|x_i\rangle)$ being the Rayleigh coefficient 66. τ_i and γ_i are scalars who minimize the Rayleigh coeffcient $\lambda(|x_{i+1}\rangle))$. One sees easily that the algorithm determines an eigenpair considering fact, that $|\omega_i\rangle$ is the result of the preconditioner applied to pencil and the preconditioner is an approximation of A^{-1}. As one can see

$$|x_{i+1}\rangle \in Span\{|w_i\rangle\,, |x_i\rangle\,, |p_i\rangle\}, \tag{77}$$

which can be exploited in the implementation later. To sum it up:

Algorithm 2 LOPCG

$|x_0\rangle \leftarrow$ initial guess, $|p_0\rangle \leftarrow |0\rangle$
while $|\,|r_i\rangle\,| < tolerance$ until converges **do**
  $\lambda_i \leftarrow \frac{\langle x_i\,|\,A\,|\,x_i\rangle}{\langle x_i\,|\,x_i\rangle}$
  $|r_i\rangle = pencil(|x_i\rangle)$
  $|x_{i+1}\rangle \leftarrow |r_i\rangle + \tau_i\,|x_i\rangle + \gamma_i\,|p_i\rangle\,|$ that λ_{i+1} is extremal
  $|p_{i+1}\rangle \leftarrow |r_i\rangle + \gamma_i\,|p_i\rangle$
end while

We have introduced until a locally optimised preconditioned continuous gradient algorithm. To extend this we need to add a block or simultaneons iterations. Therefore are "simply" more starting vectors used subspaces are generated by Krylov subspaces of each starting vector and the Ritz values can searched simultaneously. The block version of the LOPCG will not be discussed more detailed, because it is equivalent to the LOPCG if only one vector is calculated.

6. Implementation

6.1. Used Algorithm

We want to calculate the dominant part of the spectrum of the ground state of the chain of coupled rydberg atoms. Because the memory and floating point operations on a personal computer is limited we want to reduce the complexity of the computation by constructing a useful subspace (reduced basis), to find a good starting vector for the LOBPCG. The general Algorithm has the shape:

Algorithm 3 Our Algorithm

$H1, H2, H3 \leftarrow \sum_i^N \sigma_{x,i}, \sum_i^N n_i, \sum_i^N n_i n_j$ ▷ Operators: 1, Details: 6.1.1, Part 1
$base \leftarrow$ Constructed Reduced Basis ▷ Theory Coll. 3.3, Details 6.1.2, Part 2
$H1red, H2red, H3red \leftarrow base^T(H1, H2, H3) * base$ ▷ Project into reduced basis 4
for $\frac{\Delta}{\Omega}, n_s \leftarrow$ parameter space **do** ▷ Details: 6.1.3, Part 3
    $guess = lobpcg(H1red - \frac{\Delta}{\Omega}H2red + n_s H3red)$
    $exact \leftarrow LOBPCGsweep(guess)$
end for
Analyse exact depending on research question ▷ Part 4

As is shown the algorithm can be divided into four logical steps. The first step [4] is to create the parts of the Hamiltonian and have all resources by hand for further calculation therefore we have already drawn the attention the the structure of those operators [5]. The second part will be discussed in a detailed manner in chapter 6.1.2 in three approaches. The third phase has a little optimisation that is crucial not to save every vector for further data processing due to exponential grows of the hilbert space. To avoid a filled up RAM we will extract the necessary information for further visual processing in place instead of storing each full vector.

6.1.1. Creation of the Hamiltonian

First we want to examine the structure of the hamiltonians to construct the operator quickly. This is actually not a purposeful approach to speed up the whole progress, because we only create it once as one sees in algorithm 0. But it is a good practice to develop a first intuition for the system. Like Nelles we use sparse matrices in compressed Column format. We use the insights of chapter 4.2 to fill the sparse data structure by hand. Therefore we save the entries of the matrices and transform them into three vectors like required in example 1. As can be seen in figure 1 the operators in diagonal shape are created significantly more efficiently than Nelles' approach. For the σ_x^i operator the manual creation performs bad in comparison to Nelles' approach. That is the result of a bad implementation from our side, that requires much sorting and copying of data and shows how well the $kron()$ functions performs on sparse matrices.

[4] chapter 6.1.1
[5] chapter 4.2, Structure of Hamiltonians

6.1.2. Choosing a Reduced Basis

Now thus we can construct the Hamiltonians now we will draw the attention to the construction of a reduced basis[9]. We will discuss three ways of choosing a reduced base. In a nutshell we search for some eigenvectors to the parameterized hamiltonians and construct an orthonormal system from those. How to choose these eigenvectors will be discussed later. The physical intuition says, that two Hamiltonians parameterized with adjacted parameters have some similarities. We assume that the similarities are significant enough that the eigenvectors calculated in the new base are good approximation. We will explore three ways of constructing such a basis. Since the reduced basis approach is an approximation we can define an error metric to have a concept to measure the exactness of the approximation. The used mathematical quantity is the two norm of the residue vector $|HU|v_i\rangle - \lambda_i U|v_i\rangle|$, because it the residue of the eigenvector equation $H|v_i\rangle = \lambda_i|v_i\rangle$. The two norm is induced by the sesquilinear scalar product delivered by the Hilbert space. The matrix U is the reduced basis used to embed $|v_i\rangle$ into the original vector space since it is calculated in the reduced basis.

(1) The first approach as used by Nelles[18] is to choose a parameter grid, calculate the first eigenvector for each point on the parameter grid and orthonormalize that set using singular value decomposition.

(2) We will use a two level approach. We will construct a reduced basis on very coarse grid, called guess grid. After that we will create a basis on a more fine grained grid, called train grid by approximating the eigenvectors on the basis of the guess grid. After that the approximation is corrected by LOBPCG. The orthonormal basis constructed from the train grid will then be used to get an approximation for the test grid as done by Nelles. It is performed this way, because it is more efficient than calculating the ground states for every point separately, because we already can benefit from a reduced basis at this point. On both levels we used singular value decomposition to orthonormalize the vectors.

(3) The third approach is called **greedy reduced basis** approach. The idea is to analyse the error metric to find a good parameter set to inflate the basis to get the maximal improvement for the error metric. So we choose the parameters where the error metric is maximal. That should reduce the error on that point and its sourrounding drastically. Then we use LOBPCG to calculate the eigenvector for the parameter set and add it to the basis. To avoid calculating a whole singular value decomposition we used Gram Schmidt orthonormalisization. That reduces the complexity orthonormalizing the new basis from $\mathbb{O}(n^3)$ for the n-th vector added to only $\mathbb{O}(n)$ because we could continue the Gram Schmidt algorithm. With n being the dimension of the vector space. The only thing we need to know is with which parameter set one starts and how to analyse the metric. An easy approach is to start with a point in the middle of the parameter plane and analyse the error metric on a 100×100 set. As ones sees (3) we failed to implement the greedy approach correctly. The problem is, that the error does not drop to machine epsilon[6] at one point. To explain that we want to drive the attention to the process of using a reduced basis. Therefore let $|\psi_{1,2}\rangle$ be the

[6] the accuracy of the computer

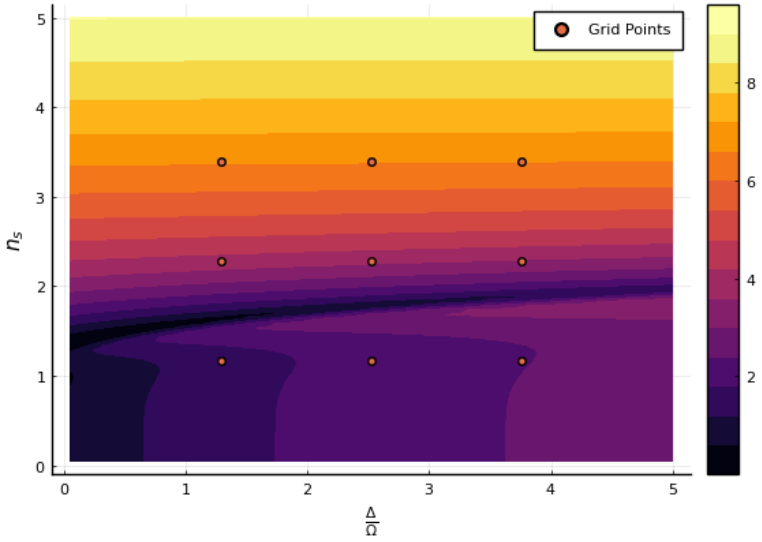

Figure 2: Error metric for regular grid for a chain of 11 Atoms and a 3 by 3 grid.

eigenvectors to H. Then in reduced basis:

$$H^{red} = \begin{pmatrix} \langle\psi_1| H |\psi_1\rangle & \langle\psi_1| H |\psi_2\rangle \\ \langle\psi_2| H |\psi_1\rangle & \langle\psi_2| H |\psi_2\rangle \end{pmatrix} =: \begin{pmatrix} \epsilon_{1,1} & \epsilon_{1,2} \\ \bar{\epsilon_{1,2}} & \epsilon_{2,2} \end{pmatrix} \tag{78}$$

$\bar{\epsilon_{1,2}} = \epsilon_{1,2}$ because our Hamiltonian has only real matrix elements. Now calculate the eigen energies:

$$0 = \chi(\lambda) = (\epsilon_{1,1} - \lambda)(\epsilon_{2,2} - \lambda) - \epsilon_{1,2}^2 = \epsilon_{1,1}\epsilon_{2,2} - \epsilon_{1,2}^2 - \lambda(\epsilon_{1,1} + \epsilon_{2,2}) + \lambda^2 \tag{79}$$

$$\implies \lambda_i = \frac{\epsilon_{1,1} + \epsilon_{2,2}}{2} \pm \sqrt{\frac{\epsilon_{1,1} + \epsilon_{2,2}}{2}^2 + \epsilon_{1,2}^2 - \epsilon_{1,1}\epsilon_{2,2}} \tag{80}$$

$$\epsilon_{1,2} \to 0 : \lambda_i = \frac{\epsilon_{1,1} + \epsilon_{2,2}}{2} \pm \frac{\epsilon_{1,1} - \epsilon_{2,2}}{2} \tag{81}$$

As one sees our approximation only is good, if of diagonal element is close to zero. Because of non diagonal shape of σ_x^i the approximation collapses. We understand $|\psi_1\rangle$ and $|\psi_2\rangle$ as the orthonormal basis vectors. So one has to choose if those fulfil the eigenvector equation or orthonormal. Both at the same time is not necessarily possible since they are calculated for different operators. Maybe we could fix it by orthonormalizing the basis so that the matrix elements which distort the approximation vanishes. Since we are using Gram-Schmidt orthogonalisation one could start with the basis vector closest to the eigenvector of the Hamiltonian considered. Unfortunately we were not able to test this due to lack of time. Instead we used the two level approach.

6.1.3. Calculation of the ground state stability

We already have constructed a vector subspace. The next step is to find a orthonormal basis. For that a singular value decomposition introduced in definition 10 is used. In

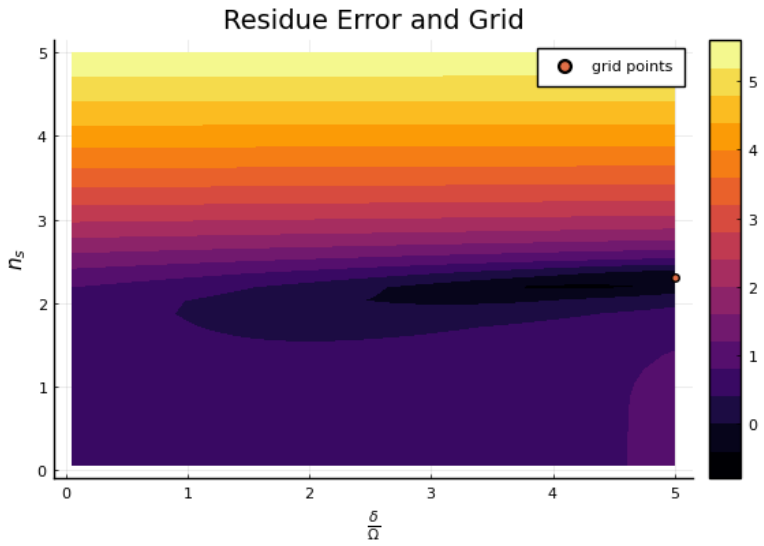

Figure 3: Error metric for our greedy approach with basis size 1 for a chain with 11 Atoms. The error is logarithmic displayed and to big.

figure 5 shows that the singular values decay rapidly. So the already reduced base can be further reduced by choosing a cut off value of the singular values. The matrix U from $H = U\Sigma V^\dagger$ as unitary matrix can be used as projector into the reduced basis (see algorithm 0). Considering the algorithm provided for definition 10 one will realise that the projector's image is the subspace defined above. Then the parts of the Hamiltonians will be projected using U on the reduced basis. After that the LOBPCG algorithm is used to get a first approximation of the ground state. The full diagonlization is used, because the reduced Hamiltonain is not in a sparse shape anymore. Then we started LOBPCG with the first approximation as start value. Because the approximation is considered as good and the nature of continuous gradient algorithms we quickly should get a nearly exact result. The correction by LOBPCG we call a LOBPCG sweep.

To analyse the result of the computation let us consider the question asked: We want to know how which is the dominant part of the spectrum of the ground state of the chain on atoms is. Some of these atoms are in their ground states g and some are in a highly excited rydberg state r. We used Nelles' approach to use the largest absolute value in the vector. Since our vector space is constructed by the tensor product over the single particle spaces which are generated by the basis $g = (10)^T$ and $r = (01)^T$. The ground state $|s\rangle$ can be written with respect to the basis definied above. Then $|\langle rgrgr... \,|\, s\rangle|$ is the proportion of $|rgrgr...\rangle$ of the spectrum of $|s\rangle$. One can replace the r, g in the sequence $|rgrgr...\rangle$ with $0, 1$. The one obtains the 10101... is the binary representation of the number i of the searched element. $\langle rgrgr... \,|\, s\rangle = |s\rangle_i^2$. In Appendix A several spectra are presented for different chain lengths.

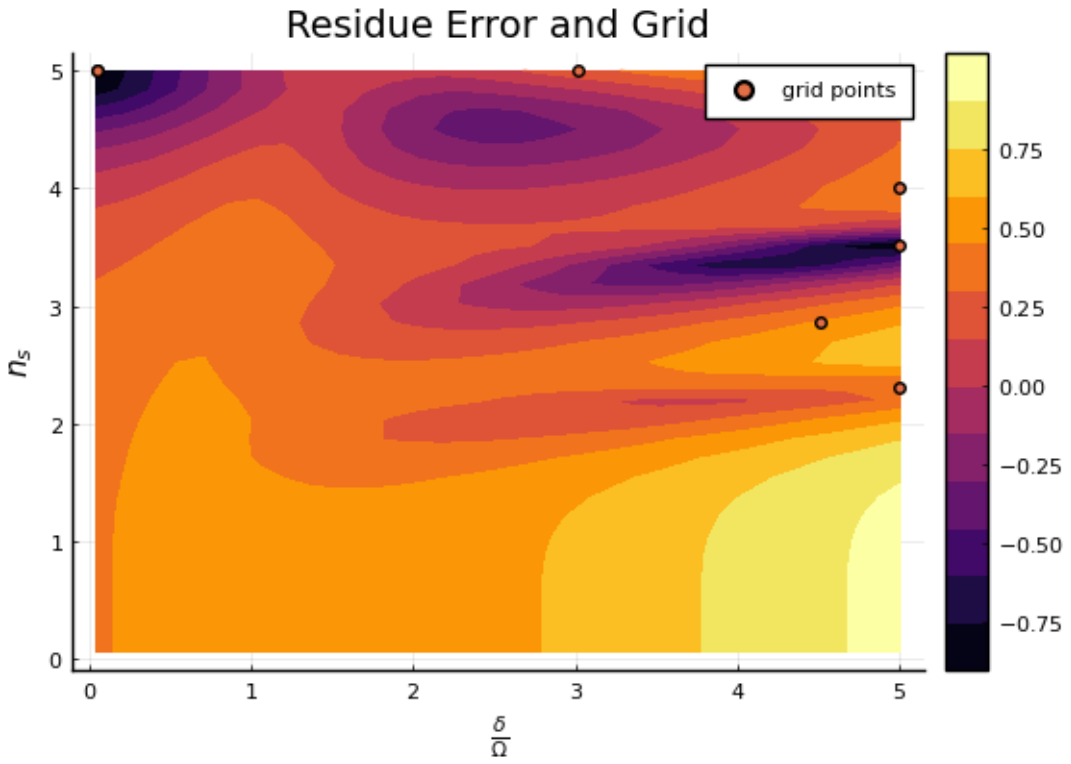

Figure 4: Error metric for our greedy approach with basis size 5 for a chain with 11 Atoms. The error is logarithmic displayed and to big.

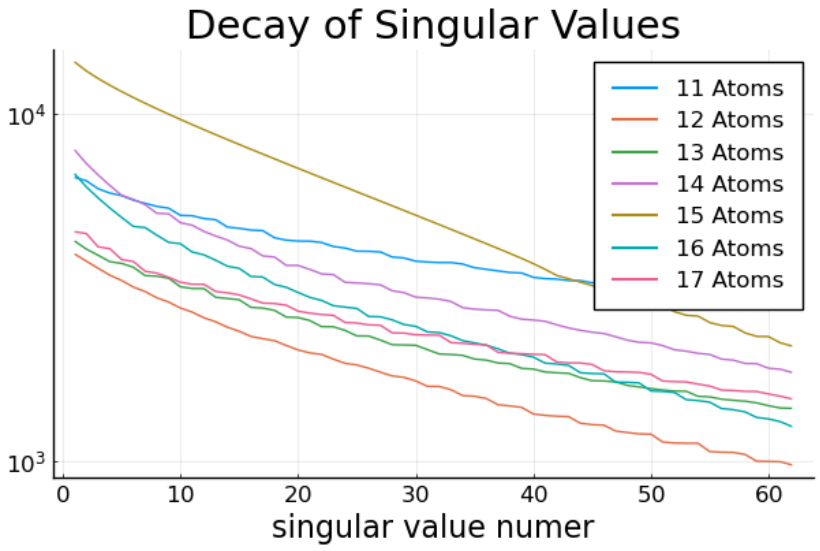

Figure 5: Decay of singular values

6.2. Structure of Code

For **Verification** of our Code we compared the Hamiltonians constructed by Nelles with ours using PyCall and the standard unit testing framework. The Hamiltonians are excatly the same. To ensure **correctness of the computation** we rely on the well engineered standard libraries of the Julia Community. This is a reliable approach since they are well reviewed by the scientific community and tested in various fields of research. But we definitely have to check for logical errors, so we will later 7 compare the results to Nelles' results. As can be seen in Appendix A the calculation only has a few divergent points for $n_s \approx 0$ and big $\frac{\Delta}{\Omega}$s. The **parallelisation** is done by mapping big map call in equally big chunks on the CPU cores. This is used for the approximation and LOBPCG sweep in step 3 and provided by Threaded Iterables. Before that parallisation will not result in a shorter runtime because of threading overhead. Unfortunately we did not parallelise singular value decomposition because of lack of time or the absence of standard libraries providing relatively new algorithms for parallel singular value decomposition.

To reduce the memory foot print we aggregated the steps of finding the spectrum of the spectrum with the calculation of the approximation and correction in the original vector space instead of parallelising them separate because the vector sizes increase exponentially and one expects the RAM size as the bottleneck. An observation during the execution of the algorithm was a big fluctuation of the allocated RAM. There is space for performance improvements due to avoid memory management system calls[7].

6.3. Finetuning the size of the reduced base

We discovered that the performance of the algorithm described above is highly dependent on the size of the reduced basis. Since we are interested in outperforming the method used by Nelles some performance considerations are appropriate. Since proper calculation time analysis is scientifically very tricky and time consuming we only will elaborate some tendencies. The difficulties of doing so are due some hard to control technical aspects. To explain the difficulties we need to understand the concept of context switches, garbage collection (GC), operating system interrupts and bare metal memory management C.

The data shown displays the mean of the runtimes of our algorithm repeated (10 times) with the standard deviation as uncertainty plotted against the basis sizes. As one can see the results are not statistically significant but we can guess the trend that the computation is faster for smaller reduced basises. The interesting property of our algorithm is, that we do not significantly loose precision by choosing a smaller reduced basis because of LOBPCG sweepes. The size of the basis influences the computation time as a tradeof between creation time of the bases, singular value decomposition, "guessing" an approximation and the LOBPCG sweep. Since we now know that small reduced basises result in a shorter runtime, we will try to solve bigger systems.

[7] alloc, free

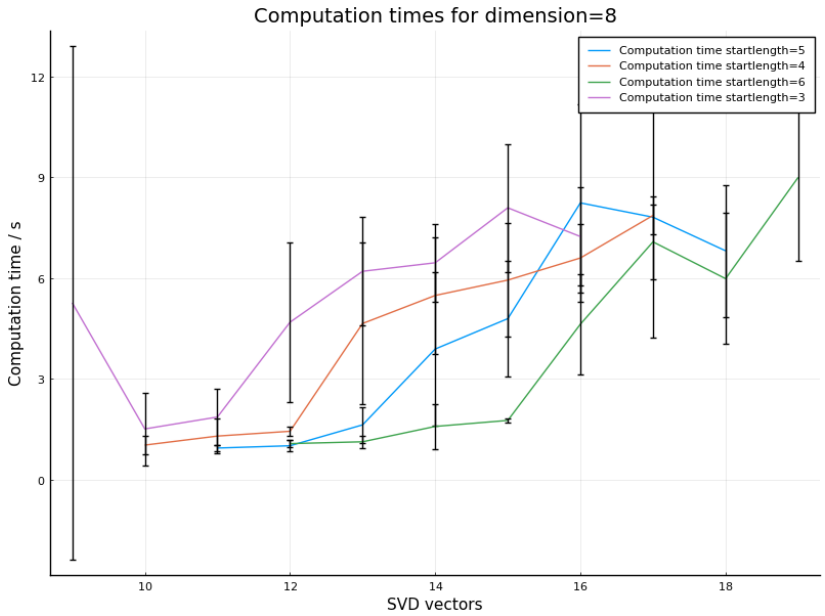

Figure 6: Runtimes for different configurations. A system of 8 Atoms was used. The startlength are the gridpoints of the guess grid and the inflate length is the size of the train grid.

7. Comparison with existing implementation

7.1. Why choose LOBPCG over Arnoldi-Lanczos

Let us reconsider the steps of the algorithm first: First we build the guess in the full basis and the train grid with the two level approach. Then we calculate Ritz vectors on the test grid in the basis induced by the guess and train grid to have a good first guess for the LOBPCG in the full basis. The LOBPCG has a good efficiency when the initial guess and the preconditioner are well chosen. The Hamiltonian consist of self inverse and diagonal parts as can be seen in figure 1. So a digonal preconditioner can be used for this Hamiltonian. Since the guess grid and the train grid require computations in the full basis we can make use the shape of the Hamiltonian which is not possible with Lanczos algorithm. When we calculating the Ritz vectors in the basis induced by the train grid to prepare the LOBPCG sweep the structure of the Hamiltonian is lost. We would prefer Lanczos algorithm, because it can construct a diagonal matrix efficiently and then a Ritz vector is computed relative quickly.

7.2. Creation of Hamiltonians

As one can see the code by directly inserting the values is incredibly slow for the $\sigma_{x,i}$ operator. CSC-Matrices are very good in multiplication and very bad by for index manipulation. In this implementation an approach is used to directly build the CSC structure by hand. The insertion in the structure is even slower than a

manual construction which under performs the old approach to compute the Kornecker products. For the construction of the H_2 and H_3 operators one can first store the diagonal values in an array which is later turned into a CSC structure which performs quite nice. The key difference between the cases of H_1 and H_2, H_3 are the number of indexwise insertions. As one can see in explanation of figure 1 this matrix format requires a special arrangement of the row and column pointer arrays. The manual construction of $H1$ performs bad, because our implementation is not very efficient. The structure of $H1$ is hard to exploit because row and column pointer increment in the same steps. We stored the entries by following the diagonal like pattern. This causes an computational expensive reordering of the data because the diagonal pattern repeats.

7.3. Accuracy of Eigenvectors

Consider the case of a chain of 11 Atoms and calculate the ground state for $(\frac{\Delta}{\Omega}, n_s) = (4.5, 1.5)$. The dominant arrangement of the excited and unexcited rydberg states is $|rgrgrgrgrgr\rangle$. But using a only a reduced base approximation this arrangement has the probality of 0.901 in contrast to the exact result of a computational cheap lobpcg correction which gives us a probailty of 0.9164. So the relative numerical error is about 1.7 %. So the method used by Nelles is relatively good but the accuracy of algorithm proposed by us is limited by the square root of the machine epsilon describing the relative deviation describable by floating point numbers. Of course calculating this exact would inflate the runtime.

7.4. Speed

Nelles algorithm needed 103.45 s to compute the spectrum of the parameterized Hamiltonian for 11. Our method finished after just about 87 s if the size of the basis is chosen small enough and the accuracy of LOBPCG is not checked. Out final run took 258 s, which is much slower but forces LOBPCG to finish with a residue norm smaller than 10^{-3}. Considering that most of the computation had been done in parallel it is definitely no performance improvement, but we used a much older CPU and there are no information about the usage of vectorized operations in Nelles' thesis.

8. Conclusion

First we set the mathematical foundation and got insights into the Lanczos Algorithm and LOBPCG. We used a Hamiltionian with two parameters. One depicting the frequency of the Laser and the other one is the coupling strength between the atoms. Since the dimensionality of our Hilbert space grows exponentially we had to build a reduced basis. This was done by using a two level approach. We constructed the basis by using a guess grid to have a good start value for the computation of the train grid. We used singular value decomposition to orthonormalize the basis and projected the

Hamiltonian on the reduced basis. For the calculation on the test grid we diagonalised the reduced Hamiltonian with LOBPCG and did an LOBPCG sweep on the approximation. We used LOBPCG to correct our Ritz vectors, because considering the shape of the Hamiltonian a diagonal preconditioner is a good approximation of the Hamiltonian. Because of the LOBPCG sweep only a small reduced basis is required to produce accurate results. We reproduced Nelles' results but found a deviation from Nelles c^2_{max} by 1.7 % for the parameters $\frac{\Delta}{\Omega} = 4.5$ $n_s = 1.5$. We failed to implement greedy basis choosing, because the error metric does not get small enough on the points added to the basis.

9. Outlook

One could compare the efficiency of LOBPCG to density renormalization groups (DMRG) to have a comparision to the algorithm typically used for those problems. DMRG could allow us to construct a reduced basis or to do the correction of the approximation more quickly. Moreover a working greedy reduced basis approach would be advantageous, because it promises a better accuracy without inflating the size of the train grid. Since we now know how exact the calculation of Nelles is, we can continue to research time dependent factors of the Hamiltonian and can analyse higher states to do thermodynamical considerations. It would also be interesting to do time evolution to solve the time dependent Schrödinger equation. Moreover the memory saving algorithm slows down the calculation because unnecessary reallocations of memory and should be implemented in with a programming language allowing manual memory management. Another possible research goal would be to implement greedy base choosing with LOBPCG sweeps during base expansion. This might be a promissing approach, since LOBPCG has shown itself extremely efficient correcting small deviations from a vector.

A. Ground State Figures and Spectra

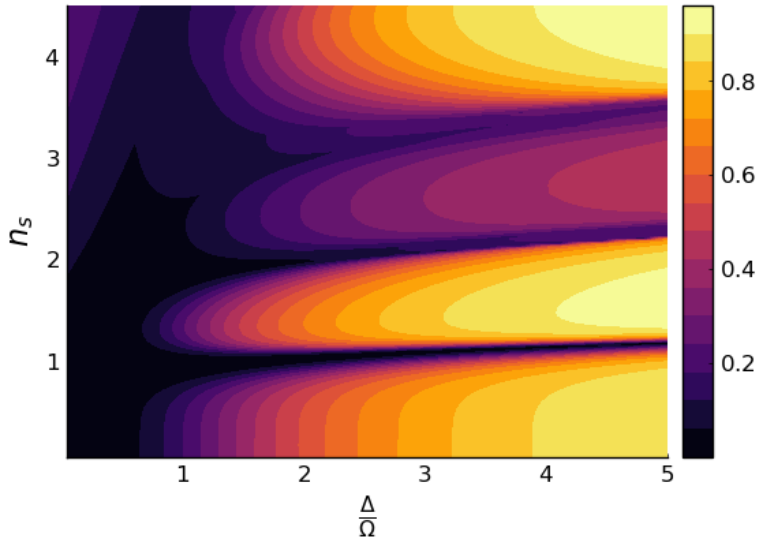

Figure 7: Diagram Depicting $|c_{max}|^2$ for a system of size $N = 11$

Table 1: Spectrum for chain of 11 Atoms

n_s	$\frac{\Delta}{\Omega}$					
	0.5		2.5		4.5	
0.5	0.0251	$\|gggggggggg\rangle$	0.6643	$\|gggggggggg\rangle$	0.8751	$\|gggggggggg\rangle$
	0.0100	$\|gggggggrggg\rangle$	0.0252	$\|ggggrgggggg\rangle$	0.0106	$\|ggggrgggggg\rangle$
	0.0100	$\|gggrgggggg\rangle$	0.0252	$\|gggggggrgg\rangle$	0.0106	$\|gggggrgggg\rangle$
1.5	0.0144	$\|grrrrrrrrg\rangle$	0.7513	$\|grgrgrgrgrg\rangle$	0.9164	$\|grgrgrgrgrg\rangle$
	0.0135	$\|grgrrrrrrg\rangle$	0.0379	$\|grgrrrgrgrg\rangle$	0.0131	$\|grgrrrrgrg\rangle$
	0.0135	$\|grrrrrrrgrg\rangle$	0.0379	$\|grgrgrrrgrg\rangle$	0.0131	$\|grgrrrgrgrg\rangle$
2.7	0.0517	$\|rrrrrrrrrr\rangle$	0.3020	$\|grrgrrrgrrg\rangle$	0.4371	$\|grrgrrrgrrg\rangle$
	0.0498	$\|grrrrrrrrr\rangle$	0.1936	$\|grrgrrgrrrg\rangle$	0.2334	$\|grrgrrgrrrg\rangle$
	0.0498	$\|rrrrrrrrrg\rangle$	0.1936	$\|grrrgrrgrrg\rangle$	0.2333	$\|grrrgrrgrrg\rangle$
4.0	0.1190	$\|rrrrrrrrrr\rangle$	0.6414	$\|grrrrgrrrrg\rangle$	0.8986	$\|grrrrgrrrrg\rangle$
	0.0802	$\|grrrrrrrrr\rangle$	0.0913	$\|grrrrrrrrrg\rangle$	0.0231	$\|grrrrrrrrrg\rangle$
	0.0801	$\|rrrrrrrrrg\rangle$	0.0485	$\|grrrrrgrrrg\rangle$	0.0163	$\|grrrrrgrrrg\rangle$

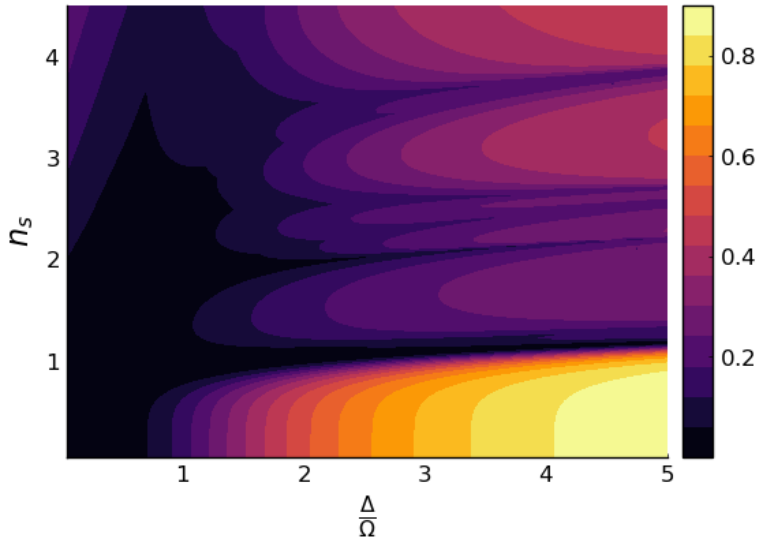

Figure 8: Diagram Depicting $|c_{max}|^2$ for a system of size $N = 12$

Table 2: Spectrum for chain of 12 Atoms

n_s	$\frac{\Delta}{\Omega}$					
	0.5		2.5		4.5	
0.5	0.0179	$\|gggggggggggg\rangle$	0.6401	$\|gggggggggggg\rangle$	0.8645	$\|gggggggggggg\rangle$
	0.0071	$\|ggggggggrggg\rangle$	0.0242	$\|ggggggggrgg\rangle$	0.0105	$\|ggggrgggggg\rangle$
	0.0071	$\|gggggrgggggg\rangle$	0.0242	$\|gggggrgggggg\rangle$	0.0105	$\|ggrgggggggg\rangle$
1.6	0.0119	$\|grrrrrrrrrrg\rangle$	0.2044	$\|grgrgrrgrgrg\rangle$	0.2790	$\|grgrgrrgrgrg\rangle$
	0.0093	$\|grgrrrrrrrrg\rangle$	0.1641	$\|grgrrgrgrgrg\rangle$	0.2172	$\|grgrrgrgrgrg\rangle$
	0.0093	$\|grrrrrrrrgrg\rangle$	0.1633	$\|grgrgrgrrgrg\rangle$	0.2164	$\|grgrgrgrrgrg\rangle$
2.3	0.0315	$\|grrrrrrrrrrg\rangle$	0.1610	$\|grrgrrrrgrrg\rangle$	0.2837	$\|grrgrgrrgrrg\rangle$
	0.0287	$\|rrrrrrrrrrrg\rangle$	0.1112	$\|grrgrrgrgrrg\rangle$	0.2835	$\|grrgrrgrgrrg\rangle$
	0.0287	$\|grrrrrrrrrrr\rangle$	0.1111	$\|grrgrgrrgrrg\rangle$	0.1177	$\|grrgrrgrrgrg\rangle$
2.55	0.0364	$\|grrrrrrrrrrg\rangle$	0.1315	$\|grrgrrrrgrrg\rangle$	0.2266	$\|grrgrrrrgrrg\rangle$
	0.0359	$\|rrrrrrrrrrrg\rangle$	0.1248	$\|grrgrrrgrrrg\rangle$	0.1415	$\|grrgrgrrgrrg\rangle$
	0.0359	$\|grrrrrrrrrrr\rangle$	0.1247	$\|grrrgrrrgrrg\rangle$	0.1415	$\|grrgrrgrgrrg\rangle$
3.2	0.0649	$\|rrrrrrrrrrr\rangle$	0.2328	$\|grrrgrrgrrrg\rangle$	0.4103	$\|grrrgrrgrrrg\rangle$
	0.0536	$\|rrrrrrrrrrrg\rangle$	0.1354	$\|grrrgrrrgrrg\rangle$	0.2302	$\|grrrgrrrgrrg\rangle$
	0.0536	$\|grrrrrrrrrrr\rangle$	0.1353	$\|grrgrrrgrrrg\rangle$	0.2292	$\|grrgrrrgrrrg\rangle$
4.5	0.1279	$\|rrrrrrrrrrr\rangle$	0.3309	$\|grrrrgrrrrrg\rangle$	0.4490	$\|grrrrgrrrrrg\rangle$
	0.0766	$\|rrrrrrrrrrrg\rangle$	0.3309	$\|grrrrrgrrrrg\rangle$	0.4489	$\|grrrrrgrrrrg\rangle$
	0.0766	$\|grrrrrrrrrrr\rangle$	0.1122	$\|grrrrrrrrrrg\rangle$	0.0356	$\|grrrrrrrrrrg\rangle$

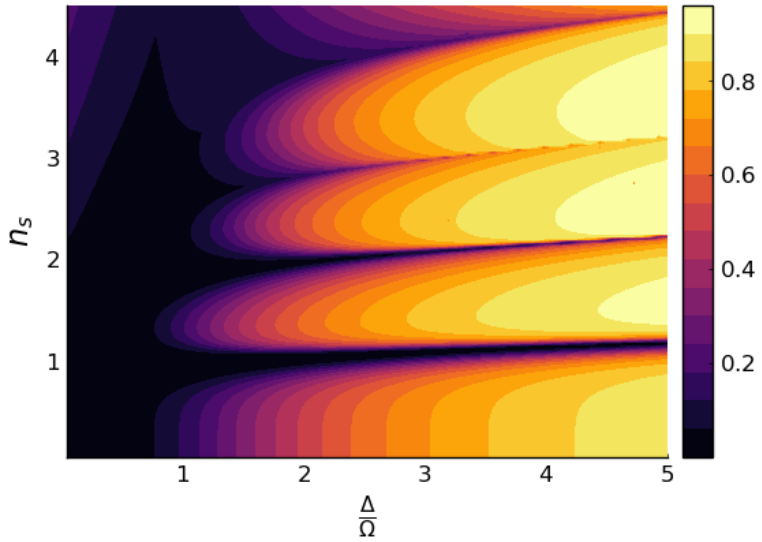

Figure 9: Diagram Depicting $|c_{max}|^2$ for a system of size $N = 13$

Table 3: Spectrum for chain of 13 Atoms

n_s	$\frac{\Delta}{\Omega}$					
	0.5		2.5		4.5	
0.5	0.0127	$\|gggggggggggg\rangle$	0.6167	$\|gggggggggggg\rangle$	0.8541	$\|gggggggggggg\rangle$
	0.0051	$\|ggggggggggrg\rangle$	0.0234	$\|gggrgggggggg\rangle$	0.0104	$\|gggggggggrgg\rangle$
	0.0051	$\|gggrgggggggg\rangle$	0.0234	$\|ggggggggggrg\rangle$	0.0104	$\|gggggggggrgg\rangle$
1.6	0.0082	$\|grrrrrrrrrrg\rangle$	0.6879	$\|grgrgrgrgrgg\rangle$	0.8994	$\|grgrgrgrgrgg\rangle$
	0.0064	$\|grrrrrrrrrgg\rangle$	0.0399	$\|grgrgrrrgrgg\rangle$	0.0139	$\|grgrgrrrgrgg\rangle$
	0.0064	$\|grgrrrrrrrrg\rangle$	0.0399	$\|grgrrrgrgrgg\rangle$	0.0139	$\|grgrrrgrgrgg\rangle$
2.7	0.0338	$\|rrrrrrrrrrrr\rangle$	0.6354	$\|grrgrrgrrgrg\rangle$	0.9077	$\|grrgrrgrrgrg\rangle$
	0.0326	$\|grrrrrrrrrrr\rangle$	0.0715	$\|grrgrrrrrgrg\rangle$	0.0196	$\|grrgrrrrrgrg\rangle$
	0.0326	$\|rrrrrrrrrrrg\rangle$	0.0704	$\|grrrrrgrrgrg\rangle$	0.0195	$\|grrrrrgrrgrg\rangle$
3.7	0.0763	$\|rrrrrrrrrrrr\rangle$	0.5915	$\|grrrgrrrgrrg\rangle$	0.9138	$\|grrrgrrrgrrg\rangle$
	0.0551	$\|grrrrrrrrrrr\rangle$	0.1007	$\|grrrrrrrgrrg\rangle$	0.0235	$\|grrrrrrrgrrg\rangle$
	0.0551	$\|rrrrrrrrrrrg\rangle$	0.1007	$\|grrrgrrrrrrg\rangle$	0.0235	$\|grrrgrrrrrrg\rangle$
4.7	0.1211	$\|rrrrrrrrrrrr\rangle$	0.4458	$\|grrrrrgrrrrg\rangle$	0.7951	$\|grrrrrgrrrrg\rangle$
	0.0699	$\|rrrrrrrrrrrg\rangle$	0.1028	$\|grrrrrrrrrrg\rangle$	0.0506	$\|grrrrrrgrrrg\rangle$
	0.0699	$\|grrrrrrrrrrr\rangle$	0.0965	$\|grrrrrrgrrrg\rangle$	0.0506	$\|grrrrgrrrrrg\rangle$

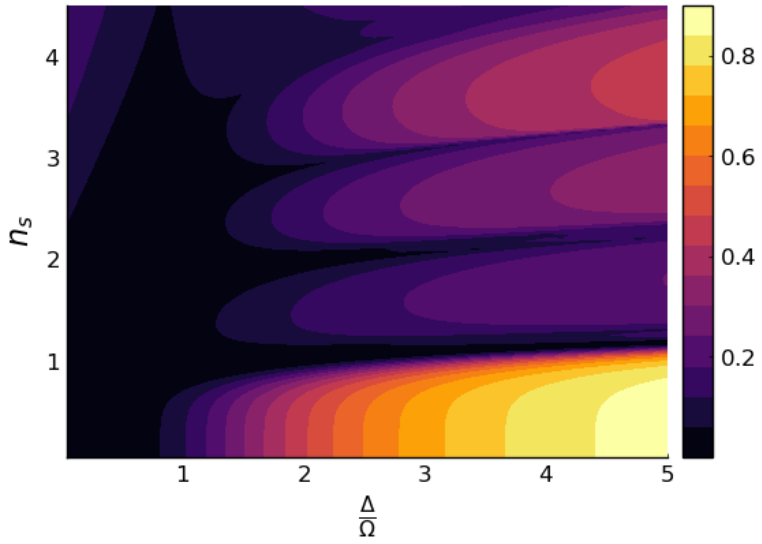

Figure 10: Diagram Depicting $|c_{max}|^2$ for a system of size $N = 14$

Table 4: Spectrum for chain of 14 Atoms

n_s	$\frac{\Delta}{\Omega}$					
	0.5		2.5		4.5	
0.5	0.0091	$\|ggggggggggggg\rangle$	0.5941	$\|ggggggggggggg\rangle$	0.8438	$\|ggggggggggggg\rangle$
	0.0036	$\|gggrgggggggggg\rangle$	0.0225	$\|gggggggggggrgg\rangle$	0.0103	$\|gggggggggggrgg\rangle$
	0.0036	$\|gggggggggggrggg\rangle$	0.0225	$\|gggrgggggggggg\rangle$	0.0103	$\|ggggrgggggggggg\rangle$
1.6	0.0056	$\|grrrrrrrrrrrg\rangle$	0.1606	$\|grgrgrrgrgrgrg\rangle$	0.2265	$\|grgrgrgrrgrgrg\rangle$
	0.0044	$\|grrrrrrrrrrgrg\rangle$	0.1602	$\|grgrgrgrrgrgrg\rangle$	0.2264	$\|grgrgrrgrgrgrg\rangle$
	0.0044	$\|grgrrrrrrrrrrg\rangle$	0.1137	$\|grgrgrgrgrrgrg\rangle$	0.1536	$\|grgrgrgrgrrgrg\rangle$
2.2	0.0173	$\|grrrrrrrrrrrrg\rangle$	0.1523	$\|grrgrrgrrrgrrg\rangle$	0.1390	$\|grrgrgrrgrgrrg\rangle$
	0.0152	$\|rrrrrrrrrrrrrg\rangle$	0.1523	$\|grrgrrrgrrgrrg\rangle$	0.0864	$\|grgrrgrrgrgrrg\rangle$
	0.0152	$\|grrrrrrrrrrrrr\rangle$	0.0756	$\|grrrgrrgrrgrrg\rangle$	0.0863	$\|grrgrgrrgrrgrg\rangle$
2.6	0.0243	$\|rrrrrrrrrrrrrr\rangle$	0.2112	$\|grrgrrgrrrgrrg\rangle$	0.3107	$\|grrgrrgrrrgrrg\rangle$
	0.0242	$\|rrrrrrrrrrrrrg\rangle$	0.2110	$\|grrgrrrgrrgrrg\rangle$	0.3106	$\|grrgrrrgrrgrrg\rangle$
	0.0242	$\|grrrrrrrrrrrrr\rangle$	0.1098	$\|grrrgrrgrrgrrg\rangle$	0.1308	$\|grrgrrgrrgrrrg\rangle$
3.7	0.0659	$\|rrrrrrrrrrrrrr\rangle$	0.2506	$\|grrrgrrrrgrrrg\rangle$	0.4250	$\|grrrgrrrrgrrrg\rangle$
	0.0475	$\|rrrrrrrrrrrrrg\rangle$	0.1768	$\|grrrgrrrgrrrrg\rangle$	0.2336	$\|grrrgrrrgrrrrg\rangle$
	0.0475	$\|grrrrrrrrrrrrr\rangle$	0.1768	$\|grrrrgrrrgrrrg\rangle$	0.2336	$\|grrrgrrrgrrrrg\rangle$
4.7	0.1068	$\|rrrrrrrrrrrrrr\rangle$	0.2068	$\|grrrrrrgrrrrrg\rangle$	0.2998	$\|grrrrrrgrrrrrg\rangle$
	0.0619	$\|rrrrrrrrrrrrrg\rangle$	0.2068	$\|grrrrrgrrrrrrg\rangle$	0.2997	$\|grrrrrgrrrrrrg\rangle$
	0.0619	$\|grrrrrrrrrrrrr\rangle$	0.0961	$\|grrrrrrrrrrrrg\rangle$	0.0902	$\|grrrrgrrrrrrrg\rangle$

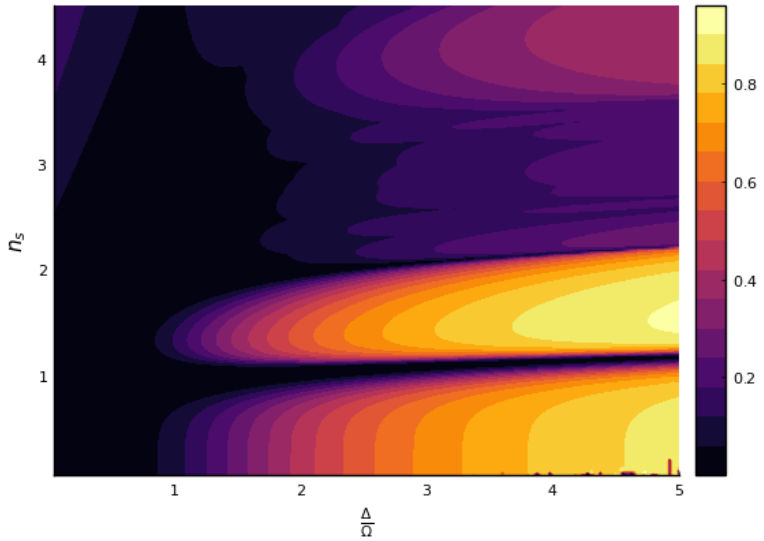

Figure 11: Diagram Depicting $|c_{max}|^2$ for a system of size $N = 15$

Table 5: Spectrum for chain of 15 Atoms

n_s	$\frac{\Delta}{\Omega}$					
	0.5		2.5		4.5	
0.5	0.0065	$\|ggggggggggggggg\rangle$	0.5724	$\|ggggggggggggggg\rangle$	0.8336	$\|ggggggggggggggg\rangle$
	0.0026	$\|ggggggggggrgggg\rangle$	0.0217	$\|gggrgggggggggg\rangle$	0.0101	$\|ggggggggrggggg\rangle$
	0.0026	$\|ggggggggrgggggg\rangle$	0.0217	$\|ggggggggrgggggg\rangle$	0.0101	$\|gggrgggggggggg\rangle$
1.6	0.0038	$\|grrrrrrrrrrrrg\rangle$	0.6498	$\|grgrgrgrgrgrgrg\rangle$	0.8854	$\|grgrgrgrgrgrgrg\rangle$
	0.0030	$\|grgrrrrrrrrrrg\rangle$	0.0377	$\|grgrgrrrgrgrgrg\rangle$	0.0137	$\|grgrgrgrrrgrgrg\rangle$
	0.0030	$\|grrrrrrrrrrrgrg\rangle$	0.0377	$\|grgrgrgrrrgrgrg\rangle$	0.0137	$\|grgrgrrrgrgrgrg\rangle$
2.25	0.0141	$\|grrrrrrrrrrrrg\rangle$	0.1099	$\|grrgrrgrgrrgrrg\rangle$	0.2548	$\|grrgrrgrgrrgrrg\rangle$
	0.0126	$\|rrrrrrrrrrrrrg\rangle$	0.0986	$\|grrgrrrrgrrgrrg\rangle$	0.1970	$\|grrgrrgrrgrrgrg\rangle$
	0.0126	$\|grrrrrrrrrrrrr\rangle$	0.0985	$\|grrgrrgrrrrgrrg\rangle$	0.1970	$\|grrgrrgrrgrgrrg\rangle$
2.5	0.0179	$\|grrrrrrrrrrrrg\rangle$	0.0980	$\|grrgrrgrrrrgrrg\rangle$	0.1711	$\|grrgrrgrgrrgrrg\rangle$
	0.0174	$\|rrrrrrrrrrrrrg\rangle$	0.0979	$\|grrgrrrrgrrgrrg\rangle$	0.1306	$\|grrgrrgrrrrgrrg\rangle$
	0.0174	$\|grrrrrrrrrrrrr\rangle$	0.0804	$\|grrgrrrgrrrgrrg\rangle$	0.1305	$\|grrgrrrrgrrgrrg\rangle$
3.0	0.0314	$\|rrrrrrrrrrrrrr\rangle$	0.1013	$\|grrrgrrgrrgrrrg\rangle$	0.1997	$\|grrrgrrgrrrgrrg\rangle$
	0.0276	$\|rrrrrrrrrrrrrg\rangle$	0.0903	$\|grrrgrrgrrrgrrg\rangle$	0.1996	$\|grrgrrrgrrgrrrg\rangle$
	0.0276	$\|grrrrrrrrrrrrr\rangle$	0.0903	$\|grrgrrrgrrgrrrg\rangle$	0.1767	$\|grrgrrrgrrrgrrg\rangle$
4.0	0.0683	$\|rrrrrrrrrrrrrr\rangle$	0.2158	$\|grrrrgrrrgrrrrg\rangle$	0.3865	$\|grrrrgrrrgrrrrg\rangle$
	0.0457	$\|grrrrrrrrrrrrr\rangle$	0.1464	$\|grrrrgrrrrgrrrg\rangle$	0.2371	$\|grrrrgrrrrgrrrg\rangle$
	0.0457	$\|rrrrrrrrrrrrrg\rangle$	0.1464	$\|grrrgrrrrgrrrrg\rangle$	0.2371	$\|grrrgrrrrgrrrrg\rangle$

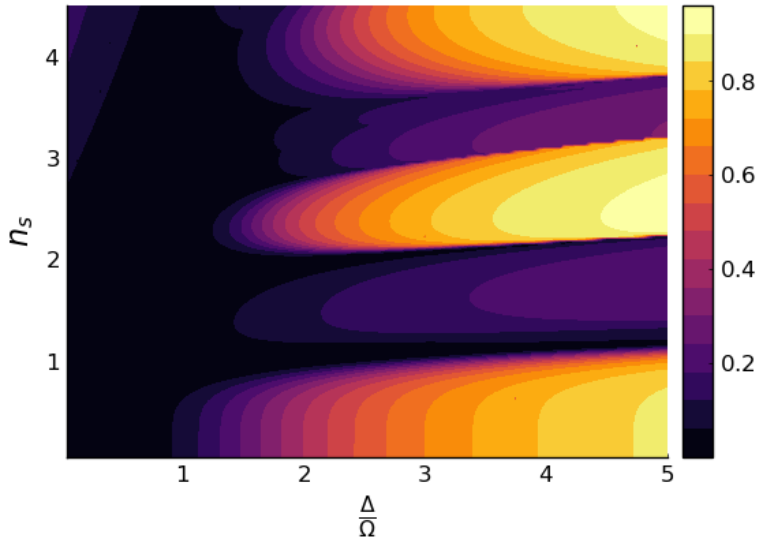

Figure 12: Diagram Depicting $|c_{max}|^2$ for a system of size $N = 16$

Table 6: Spectrum for chain of 16 Atoms

n_s	$\frac{\Delta}{\Omega}$					
	0.5		2.5		4.5	
0.5	0.0046	$\|gggggggggggggggg\rangle$	0.5516	$\|gggggggggggggggg\rangle$	0.8235	$\|gggggggggggggggg\rangle$
	0.0018	$\|ggggggggggrgggg\rangle$	0.0209	$\|ggggggggggrgggg\rangle$	0.0100	$\|ggggggggggrgggg\rangle$
	0.0018	$\|gggggggggrggggg\rangle$	0.0209	$\|gggggggggggggrgg\rangle$	0.0100	$\|gggggggggggggrgg\rangle$
1.6	0.0026	$\|grrrrrrrrrrrrrg\rangle$	0.1399	$\|grgrgrgrrgrgrgrg\rangle$	0.2066	$\|grgrgrgrrgrgrgrg\rangle$
	0.0020	$\|grgrrrrrrrrrrrrg\rangle$	0.1231	$\|grgrgrgrgrrgrgrg\rangle$	0.1790	$\|grgrgrgrgrrgrgrg\rangle$
	0.0020	$\|grrrrrrrrrrrrgrg\rangle$	0.1220	$\|grgrgrrgrgrgrgrg\rangle$	0.1779	$\|grgrgrrgrgrgrgrg\rangle$
2.25	0.0109	$\|grrrrrrrrrrrrrrg\rangle$	0.6531	$\|grrgrrgrrgrrgrrg\rangle$	0.8828	$\|grrgrrgrrgrrgrrg\rangle$
	0.0098	$\|rrrrrrrrrrrrrrrg\rangle$	0.0435	$\|grrgrrgrrrrrgrrg\rangle$	0.0151	$\|grrgrrrrrgrrgrrg\rangle$
	0.0098	$\|grrrrrrrrrrrrrrr\rangle$	0.0435	$\|grrgrrrrrgrrgrrg\rangle$	0.0151	$\|grrgrrgrrrrrgrrg\rangle$
3.25	0.0336	$\|rrrrrrrrrrrrrrrr\rangle$	0.1301	$\|grrrgrrgrrrgrrrg\rangle$	0.2895	$\|grrrgrrrgrrgrrrg\rangle$
	0.0274	$\|grrrrrrrrrrrrrrr\rangle$	0.1301	$\|grrrgrrrgrrgrrrg\rangle$	0.2894	$\|grrrgrrrgrrrgrrrg\rangle$
	0.0274	$\|rrrrrrrrrrrrrrrg\rangle$	0.1050	$\|grrrgrrrrrrgrrrg\rangle$	0.1267	$\|grrgrrrgrrrgrrrg\rangle$
4.25	0.0686	$\|rrrrrrrrrrrrrrrr\rangle$	0.5138	$\|grrrrgrrrrgrrrrg\rangle$	0.8832	$\|grrrrgrrrrgrrrrg\rangle$
	0.0433	$\|grrrrrrrrrrrrrrr\rangle$	0.0790	$\|grrrrgrrrrrrrrrg\rangle$	0.0221	$\|grrrrgrrrrrrrrrg\rangle$
	0.0433	$\|rrrrrrrrrrrrrrrg\rangle$	0.0790	$\|grrrrrrrrrgrrrrg\rangle$	0.0221	$\|grrrrrrrrrgrrrrg\rangle$

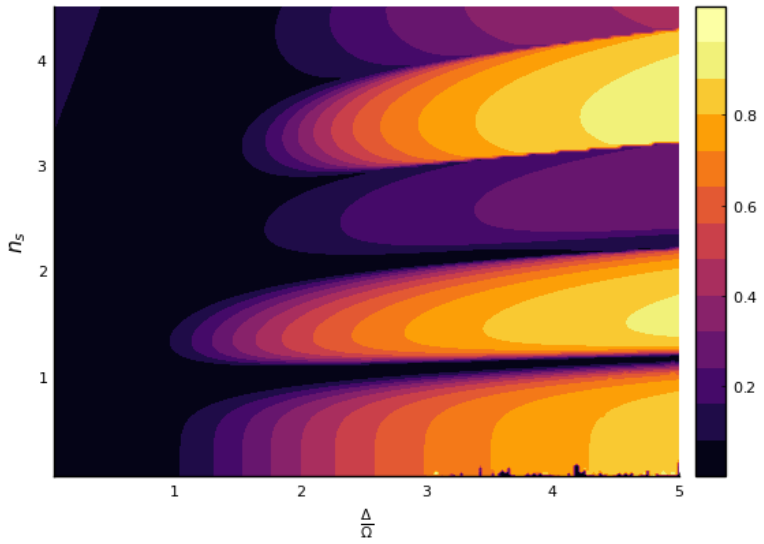

Figure 13: Diagram Depicting $|c_{max}|^2$ for a system of size $N = 17$. You see some numerical artefacts due to disabled tolerance checking. That was done because we had no time left to do a precision run on this system.

Table 7: Spectrum for chain of 17 Atoms

n_s	$\frac{\Delta}{\Omega}$					
	0.5		2.5		4.5	
0.5	0.0033	$\|gggggggggggggggg\rangle$	0.5313	$\|gggggggggggggggg\rangle$	0.8136	$\|gggggggggggggggg\rangle$
	0.0013	$\|gggggrgggggggggg\rangle$	0.0201	$\|gggggggggggrgggg\rangle$	0.0099	$\|gggggggggggggrgg\rangle$
	0.0013	$\|grgggggggggggggg\rangle$	0.0201	$\|ggggrgggggggggg\rangle$	0.0099	$\|ggggrgggggggggg\rangle$
1.5	0.0013	$\|grrrrrrrrrrrrrrg\rangle$	0.6460	$\|grgrgrgrgrgrgrg\rangle$	0.8758	$\|grgrgrgrgrgrgrg\rangle$
	0.0012	$\|grrrrrrrrrrrrrgrg\rangle$	0.0326	$\|grgrgrgrrrgrgrg\rangle$	0.0125	$\|grgrgrgrrrgrgrg\rangle$
	0.0012	$\|grgrrrrrrrrrrrrg\rangle$	0.0326	$\|grgrgrgrrrgrgrg\rangle$	0.0125	$\|grgrgrgrgrrrgrg\rangle$
2.25	0.0085	$\|grrrrrrrrrrrrrrg\rangle$	0.1488	$\|grrgrrgrrrgrrgrg\rangle$	0.0915	$\|grrgrrgrgrgrrgrg\rangle$
	0.0076	$\|grrrrrrrrrrrrrrr\rangle$	0.1202	$\|grrgrrgrrgrrrgrg\rangle$	0.0836	$\|grrgrrgrrgrrrgrg\rangle$
	0.0076	$\|rrrrrrrrrrrrrrrg\rangle$	0.1201	$\|grrgrrrgrrgrrgrg\rangle$	0.0835	$\|grrgrrgrgrrgrgrg\rangle$
3.4	0.0326	$\|rrrrrrrrrrrrrrrr\rangle$	0.5908	$\|grrrgrrrgrrrgrrg\rangle$	0.8959	$\|grrrgrrrgrrrgrrg\rangle$
	0.0255	$\|rrrrrrrrrrrrrrrg\rangle$	0.0639	$\|grrrgrrrrrrrgrrg\rangle$	0.0187	$\|grrrgrrrrrrrgrrg\rangle$
	0.0255	$\|grrrrrrrrrrrrrrr\rangle$	0.0623	$\|grrrrrrrgrrrgrrg\rangle$	0.0186	$\|grrrgrrrgrrrrrrg\rangle$
4.4	0.0654	$\|rrrrrrrrrrrrrrrr\rangle$	0.2089	$\|grrrrgrrrrrgrrrg\rangle$	0.3958	$\|grrrrgrrrrrgrrrg\rangle$
	0.0400	$\|rrrrrrrrrrrrrrrg\rangle$	0.1546	$\|grrrrgrrrrgrrrrg\rangle$	0.2320	$\|grrrrgrrrrgrrrrg\rangle$
	0.0400	$\|grrrrrrrrrrrrrrr\rangle$	0.1546	$\|grrrrrgrrrrgrrrg\rangle$	0.2320	$\|grrrrrgrrrrgrrrg\rangle$

B. Using the Framework

Now we want to elaborate the implementation of the algorithm. As a toolchain the Julia programming language is used with the standard packages:

- SparseArrays [6] to get a good implementation of sparse matrices
- IterativeSolvers [1] to get a LOBPCG implementation and a singular value decomposition in an iterative way suitable for sparse matrices.
- Preconditioniers to get a Diagonal preconditioner.
- Plots [2] to used Python's matplotlib in Julia.
- ThreadedIterables [7] for an easy way of mapping the computation on the CPU cores
- ProgressMeter [4] to display an estimated time for finishing the computation to calm down the programmer that one sees the computation proceed
- Additonally we used PyCall [5] to embed Nelles' code for testing.

Having discussed the basic dependencies we also used and . We used the standard approach to structure the project: The actual computations are stored in an examples folder, the framework one finds in a separate source directory and the comparision to the results found by Nelles are located in the test direcrtory. Do not forget to create a res, plots and plots/surfaces directory used to store the results of our computation when cloning the project from git [3]. Make sure that all dependencies are installed. One the example "createcolorplots.jl" creates the colorplots. The file examples "eagertest.jl" contains our experiment with greedy algorithm and plot_operators_and_benchmark.jl creates figure 1. benchmarkcreateresults[8] runs the run time estimations and the plot is created with caltimeplots.jl. analyse_spectrum.jl is used to create the tables in appendix A.

Two files are used to do the experiments for this thesis. The first one is fast_hamil.jl that creates the parts of Hamiltonians by building the CSC-structure manually. The lobpcg.jl file contains the function used for all plots. Note the underlying structure is vector with a 4-tuple. The tuples contain the $\frac{\Delta}{\Omega}$ and n_S values to keep track where the vector belongs two. The third element is the eigenvalue that is shifted by an integral number and the last is the vector in the full base. There are 5 elementary functions and 2 special functions in that file. The elementary functions are to calculate the eigenvector in the full space (exaktbase). approxbase takes care of calculating approx imitation with respect to the basis constructed by reducebase!. exactfromapprox calculates an lobpcg sweep on the approximation. The two special are plotsurface which draws a countour plot with an error metric. One can either use the standard optimiser being the residue norm 3.2 or some own implementations. The return value of this

[8] One does not want to execute this. Takes like forever

function are the norm values to each points. lowmemcolorplot creates a colorplot with an approximation and LOBPCG sweep already included. The purpose of that function is to save memory by not allocating $2^N * sizeof(Float64) * 200^2$ bytes but only $2^N * sizeof(Float64) * 16$. That slows down the calculation due to memory management but avoids a fully requested RAM. In the test directory one find the comparison of Nelles [18] and our Hamiltonians. One does not forget to git submodule properly.

C. Excourse to operating system designs

On modern personal computers a process consists of register states, stacks, code and associated resources like open files, threads or memory. [21] A Thread basically is an executable part of a process that can be run on one CPU core. So it needs a stack and a register states of its own. Now a **Context Switch** is when a part of the operating system called Schedular decides to load the resources of a thread into the cores hardware and jump into processes code. Every keystroke on the keyboard or movement of the mouse triggers or signal on the ethernet socket triggers a context switch via **interrupt**. Then the operating system saves the state of a threads and jumps in to operating system routine to process such input signals. Also opening a file or allocating memory induces a context switch, since those resources are assinged by the operating systems.

Memory management is a way to assign the resource memory. The main problem is to deal with defragmented memory on the side of the operating system. And to request and free memory on the processes side. To deal with defragmentation modern x64 architectures use a technology called **paging** wich allows to virtually concat not adjacting parts of memory to a virtual coherent array of memory. Performance wise high level instructions on the CPU do avoid performance loss but if the memory can defragmented it required some time to find enough not claimed pages to concat. On the process side it is very to see when one need to request memory and very to check when to free the memory. So there are 3 approaches to manage that. (1) In high performance computing it would be best to manage the memory completely manulaly. So Requesting at the beginning as much memory one needs and adapt the algorithm to get along with no defragmentation on the developer side. This is relatively hard and error prone. (2) The secound way is to use the old C++ desing pattern **Resource Aquisition is Initialization (RAII)** where one requests the resources needed in the contructor and free it when via destructor when ones object goes out of scope. This is less error prone but requires much discipline to avoid unaccessible resources or double freeing of memory. The third (3) way is to use a **Garbage Collection (GC)** algorithm which is specialized on finding unused resources and freeing them. There are many good algorithms like the effecient Javas generational approach or the realtime ensuring Tricolor Algorithm by Google. The easierst way is just reference counting. This is done by count the reference on an object by increasing a number for every object that assigned to reference and decreasing if it goes out of scope. This way is relative reliable when one avoids cycle reference which are almost always created on purpose.

It is practically impossible to avoid context switches by the operating triggered by

input output operations or some other interrupts caused by the hardware and other necessities of the operating system. This is problem because during the context switch takes time and after it the calculation is stopped until the calculation thread is switched back on the CPU core. Another problem are GC cycles and memory management. Since Julia is not fully RAII sometimes the interpreter has to delete or at least search for unused objects. That is a very optimised but still time consuming process. Another problem is memory management at all since the malloc calls use paging to create virtual coherent array of RAM. So the performance of memory management highly depends on the state of the computer when starting the calculation an other processes fragmenting the memory. An approach to solve that problem would be to manual memory management avoiding unneeded freeing and allocating memory. That is technically very hard to implement and would probably not result in huge performance increase.

References

[1] URL https://iterativesolvers.julialinearalgebra.org/dev/.

[2] URL http://docs.juliaplots.org/latest/.

[3] The code. URL https://git.rwth-aachen.de/philipp.rosendahl/rydberg-chain-with-dmrg.

[4] URL https://www.juliapackages.com/p/progressmeter.

[5] URL https://github.com/JuliaPy/PyCall.jl.

[6] URL https://docs.julialang.org/en/v1/stdlib/SparseArrays/.

[7] URL https://github.com/marekdedic/ThreadedIterables.jl.

[8] Compressed sparse column format (csc) lecuture notes. URL https://scipy-lectures.org/advanced/scipy_sparse/csc_matrix.html. Online accessed 30.10.2021.

[9] K Veroy A-L Gerner. Certified reduced basis methods for parametrized saddle point problems. *SIAM J. Sci. Comput.*

[10] Sandvik Anders W. Stochastic series expansion method with operator-loop update. *Phys.Rev.B59*, 1999. URL arXiv:cond-mat/9902226.

[11] Wolfgang Demtröder. *Experimental Physik 3*, chapter 6.6. Springer-Verlag, 2010.

[12] Prof. Fabian Hassler. Quantenmechanik. *lecture notes*, 2019.

[13] Kh. Dr. Ikramov. Numerische mathemkt. *Journal of Soviet Mathematics*, 1986. URL https://doi.org/10.1007/BF01262409.

[14] Barth Knaber. *Lineare Algebra*, chapter 4.6. Springer-Verlag, 2018.

[15] Andrew Knyazev. *SIAM Journal on Scientific Computing*, 2020. Online, accessed 19.08.2021.

[16] Samuel Mahler, 2021.

[17] Gabriele Nebe. *Skript zur Vorlesung Lineare Algebra I*, chapter 3.3. 2017.

[18] Nigel Nelles. Modelling and numerical simulation of a chain of neutral atoms coherently coupled to highly excited rydberg states, 2020.

[19] Wolfgang Nolting. *Grundkurs Theoretische Physik 5/2*, chapter 7.3. Springer-Verlag, 2014.

[20] Richard A. Snay. Reducing the profile of sparse symmetric matrices. *Bulletin Géodésique, Volume 50*, 1976. URL `https://ui.adsabs.harvard.edu/abs/1976BGeod..50..341S/abstract`.

[21] Andrew Tanenbaum. *OPERATING SYSTEMS DESIGN AND IMPLEMENTATION*, chapter 1.3, 2.1, 2.4. Pearson Education, Inc., 2006.