In this section we present a practical introduction to the manipulation of Fermionic Gaussian systems. In particular we will focus on the encoding of these systems in a computer using a Julia package called `F_utilities`. Starting from the basis, we will move to relevant modern results and techiques, and finish introducing novel algorithms. By the end of the chapter the reader should be able to reproduce all the results of this thesis using the package `F_utilities` and make good use of `F_utilities` for other relevant numerical simulations. It should be possible to read this chapter independently from the rest of the thesis and use it as the documentation of the package `F_utilities` and as a useful resource in the broad and fertile field of Fermionic Gaussian states.

0.1 The canonical anticommutation relations

0.1.1 The Hilbert space characterised by the canonical anticommutation relations

Consider a set of operators $\{a_i\}_{i=1}^N$ acting on a Hilbert space $\mathcal{H}$. We say that these operators satisfy the *canonical anticommutation relation* (CAR) when they satisfy

$$\{a_i, a_j^\dagger\} = \mathbb{I}\delta_{i,j}; \qquad \{a_i, a_j\} = 0, \tag{1}$$

with $\{a, b\} \coloneqq ab + ba$ the notation for the anticommutator.
As shown in [**?**] a number of properties of the set of operators $\{a_i\}_{i=1}^N$ and of the Hilbert space $\mathcal{H}$ can be inferred just by the fact that such operators exist and obey the CAR.
The $a_i^\dagger a_i$ form a set of *commuting, Hermitian, positive operators* with eigenvalues $\{0, 1\}$. We denote with $\vec{x} \in \{0, 1\}^N$ a binary string of length N with the i-th elements x_i. With $|\vec{x}\rangle$ we identify one of the 2^N states that is the simultaneous eigenstate of $a_i^\dagger a_i$ for all $i = 1, \dots, N$ with eigenvalues respectively x_i. The operator a_i acts as a *lowering operator* for $a_i^\dagger a_i$ and $a_i^\dagger$ acts as a *raising operator* for $a_i^\dagger a_i$ in the sense that

1. If $a_i^\dagger a_i|\vec{x}\rangle = |\vec{x}\rangle$, that is, $|\vec{x}\rangle$ is an eigenvector of $a_i^\dagger a_i$ with eigenvalue equal to 1, then the action of a_i on $|\vec{x}\rangle$ lowers the corresponding eigenvalue, meaning that $a_i^\dagger a_i(a_i|\vec{x}\rangle) = 0(a_i|\vec{x}\rangle)$.

2. If $a_i^\dagger a_i|\vec{x}\rangle = 0|\vec{x}\rangle$, that is, $|\vec{x}\rangle$ is an eigenvector of $a_i^\dagger a_i$ with eigenvalue equal to 0, then the action of $a_i^\dagger$ on $|\vec{x}\rangle$ raises the corresponding eigenvalue, meaning that $a_i^\dagger a_i(a_i^\dagger|\vec{x}\rangle) = 1(a_i^\dagger|\vec{x}\rangle)$.

We define an *ordering* by explicitly defining $|\vec{x}\rangle \coloneqq (a_1^\dagger)^{x_1}(a_2^\dagger)^{x_2} \dots (a_N^\dagger)^{x_N}|\vec{0}\rangle$, where $\vec{0}$ is the string of N zeros. The set $\{|\vec{x}\rangle\}_{\vec{x}\in\{0,1\}^N}$ forms an orthonormal basis. Since the dimension of the Hilbert space $\mathcal{H}$ is 2^N, then $\{|\vec{x}\rangle\}_{\vec{x}\in\{0,1\}^N}$ is an orthonormal basis of $\mathcal{H}$.
The orthonormal basis $\{|\vec{x}\rangle\}_{\vec{x}\in\{0,1\}^N}$ is called *Fock basis*. The action of the

raising and lowering operators on $|\vec{x}\rangle$ is then

$$a_i|\vec{x}\rangle = \begin{cases} -(-1)^{S^i_{\vec{x}}}|\vec{x'}\rangle \text{ with } x'_i = 0 \text{ and } x'_{j\neq i} = x_{j\neq i}, & \text{if } x_i = 1 \\ 0 & \text{if } x_i = 0 \end{cases}, \tag{2}$$

$$a_i^\dagger|\vec{x}\rangle = \begin{cases} 0 & \text{if } x_i = 1 \\ -(-1)^{S^i_{\vec{x}}}|\vec{x'}\rangle \text{ with } x'_i = 1 \text{ and } x'_{j\neq i} = x_{j\neq i}, & \text{if } x_i = 0 \end{cases}, \tag{3}$$

with $S^i_{\vec{x}} = \sum_{k=1}^{i-1} x_k$.
In appendix **??** we report some useful equalities valid for operators satisfying the CAR.

0.1.2 Dirac and Majorana representations

The raising and lowering operators $a_i^\dagger$,a_i are called *Dirac operators* and they represent the action of adding and removing the i-th Fermionic mode.
Both a_i and its adjoint $a_i^\dagger$ are not Hermitian. The Hermitian combinations of the raising and lowering operators

$$x_i = \tfrac{a_i + a_i^\dagger}{\sqrt{2}}, \qquad p_i = \tfrac{a_i - a_i^\dagger}{i\sqrt{2}}, \tag{4}$$

are called *Majorana operators.*
The inverse transformations are:

$$a_i = \tfrac{x_i + ip_i}{\sqrt{2}}, \qquad a_i^\dagger = \tfrac{x_i - ip_i}{\sqrt{2}}. \tag{5}$$

In terms of Majorana operators the CARs read as

$$\{x_i, x_j\} = \{p_i, p_j\} = \delta_{i,j}, \qquad \{x_i, p_j\} = 0. \tag{6}$$

We remark that to Majorana operators labelled by i correspond Dirac operators labelled by i. Moving between Majorana and Dirac operators does not mix modes.

Vector notation We can collect the Dirac operators of a system with N modes in the vector $\vec{\alpha}$ of length $2N$ defined as

$$\vec{\alpha} = \begin{pmatrix} a_0^\dagger \\ \vdots \\ a_{N-1}^\dagger \\ a_0 \\ \vdots \\ a_{N-1} \end{pmatrix} \quad , \quad \vec{\alpha}^\dagger = \begin{pmatrix} a_0 & \dots & a_{N-1} & a_0^\dagger & \dots & a_{N-1}^\dagger \end{pmatrix}. \tag{7}$$

Analogously we can collect the Majorana operators in the vector $\vec{r}$ defined as

$$\vec{r} = \left(\begin{array}{c} x_0 \\ \vdots \\ x_{N-1} \\ p_0 \\ \vdots \\ p_{N-1} \end{array} \right) , \tag{8}$$

in terms of $\vec{r}$ the CAR are conveniently written as

$$\{r_i, r_j\} = \delta_{i,j}. \tag{9}$$

We define the unitary matrix Ω as

$$\Omega = \tfrac{1}{\sqrt{2}} \left(\begin{array}{cc} \mathbb{I} & \mathbb{I} \\ i\mathbb{I} & -i\mathbb{I} \end{array} \right) \quad , \quad \Omega^\dagger = \Omega^{-1} = \tfrac{1}{\sqrt{2}} \left(\begin{array}{cc} \mathbb{I} & -i\mathbb{I} \\ \mathbb{I} & i\mathbb{I} \end{array} \right) . \tag{10}$$

Such a matrix, applied to the vector of the Dirac operators $\vec{\alpha}$, returns the vector of Majorana operators $\vec{r} = \Omega\vec{\alpha}$.

Fermionic transformation A transformation $\vec{r} \to \vec{s} = O\vec{r}$ is said to preserve the CAR in the Majorana representation if it maps a vector of Majorana operators $\vec{r}$ to a new one $\vec{s} = O\vec{r}$. If we explicitly impose $\vec{s} = O\vec{r}$ to preserve the CAR we obtain

$$\delta_{i,j} = \{s_i, s_j\} = \sum_{k,l} O_{i,k} O_{j,l} \{r_k, r_l\} = (OO^T)_{i,j}, \tag{11}$$

thus matrix O must be an orthogonal matrix.
We call *Fermionic transformation* any transformation $\vec{\alpha} \to \vec{\beta} = U\vec{\alpha}$ that preserves the CAR of the Dirac operators vectors. Matrix U has the form of $U = \Omega^\dagger O \Omega$ with O an orthogonal matrix. It has been shown in [**?**] that Fermionic transformations are generated by Fermionic quadratic Hamiltonian (to be defined later), thus have the general form $U = e^{-i\hat{H}}$, with $\hat{H}$ a generic Fermionic quadratic Hamiltonian.

Clifford Algebra The Majorana operators $\{r_i\}_{i=1,\dots,2N}$ are Hermitian, traceless and generate the Clifford algebra denoted by $\mathcal{C}_{2N}$.
Any arbitrary operator $X \in \mathcal{C}_{2N}$ can be represented as a polynomial of the Majorana operators as [**?**]

$$X = \alpha_0 \mathbb{I} + \sum_{p=1}^{2N} \sum_{1 \le q_1 < \dots < q_p \le 2N} \alpha_{q_1,\dots,q_p} r_{q_1} \dots r_{q_p}, \tag{12}$$

where $\mathbb{I}$ is the identity and the coefficients α_0 and $\alpha_{q_1,\dots,q_p}$ are real. When the representation of $X \in \mathcal{C}_{2N}$ involves only even powers of Majorana operators, we call it an *even operator*. If the representation of X involves only odd powers of Majorana operators, then X is called *odd* operator.

We define the *parity operator* as

$$P = (i2)^N r_1 r_2 \dots r_{2N} = \prod_{i=1}^{N} (\mathbb{I} - 2a_i^\dagger a_i) \tag{13}$$

Every even operator X commutes with the parity operator P. The parity p_X of an operator X is defined as $PX = p_X X$ and it can only assume the two values $p_X \in \{-1, 1\}$.

Fermionic quadratic Hamiltonians (to be defined later) are even operators. For an N-mode Fermionic system with orthonormal basis $\{|\vec{x}\rangle\}$, the matrices $|\vec{x}\rangle\langle\vec{x}|$ defined for every $\vec{x}$ have the polynomial representation

$$|\vec{x}\rangle\langle\vec{x}| = \left(\frac{\mathbb{I}}{2} - i(-1)^{x_1} r_1 r_2\right)\left(\frac{\mathbb{I}}{2} - i(-1)^{x_2} r_3 r_4\right)\dots\left(\frac{\mathbb{I}}{2} - i(-1)^{x_N} r_{2N-1} r_{2N}\right), \tag{14}$$

thus $\{|\vec{x}\rangle\langle\vec{x}|\}$ are all even operators with parity $p_{|\vec{x}\rangle\langle\vec{x}|} = -(-1)^{\sum_{i=1}^N x_i}$.

Mixed matrices $|\vec{x}\rangle\langle\vec{x'}|$ with $\vec{x} \neq \vec{x'}$ are odd operators if $\text{mod}(d(\vec{x}, \vec{x'}), 2) = 1$, where $d(\vec{x}, \vec{y})$ is the Hamming distance of $\vec{x}$ and $\vec{y}$, and they are even operators if $\text{mod}(d(\vec{x}, \vec{x'}), 2) = 0$.

0.2 Fermionic Quadratic Hamiltonians

0.2.1 Dirac Representation

The general *Fermionic quadratic Hamiltonians* (f.q.h.) on a finite lattice of N sites in the Dirac operators representation can be written as

$$\hat{H} = \sum_{i,j=1}^{N} \left(A_{i,j} a_i^\dagger a_j - \bar{A}_{i,j} a_i a_j^\dagger + B_{i,j} a_i a_j - \bar{B}_{i,j} a_i^\dagger a_j^\dagger\right), \tag{15}$$

where A is a *Hermitian* complex matrix, $A^\dagger = A$, and B is a *skew-symmetric* complex matrix, $B^T = -B$.

Defining the matrix

$$H = \begin{pmatrix} -\bar{A} & B \\ -\bar{B} & A \end{pmatrix}, \tag{16}$$

the *compact form* of equation (15) reads

$$\hat{H} = \vec{\alpha}^\dagger H \vec{\alpha}. \tag{17}$$

We will call *Hamiltonians* both $\hat{H}$ and H as for a fixed choice of Dirac operators one completely identifies the other.

0.2.2 Majorana Representation

The Majorana representation of the generic f.q.h. reads as

$$\hat{H} = i \sum_{i,j=1}^{N} \left(h^{xx}_{i,j} x_i x_j + h^{pp}_{i,j} p_i p_j + h^{xp}_{i,j} x_i p_j + h^{px}_{i,j} p_i x_j \right) = i \vec{r}^\dagger h \vec{r}, \tag{18}$$

where

$$ih = \Omega H \Omega^\dagger = i \begin{pmatrix} \Im\{A+B\} & \Re\{A+B\} \\ \Re\{B-A\} & \Im\{A-B\} \end{pmatrix} = i \begin{pmatrix} h^{xx} & h^{xp} \\ h^{px} & h^{pp} \end{pmatrix}. \tag{19}$$

Where $\Im\{\cdot\}$ and $\Re\{\cdot\}$ are respectively the imaginary and the real part of their argument.
Using the properties of matrices A and B, it is easy to see that matrix h is real and skew-symmetric.

0.2.3 Diagonalisation

Diagonal form of the Hamiltonian with Dirac operators Given a particular f.q.h. $\hat{H}$ in the general form (15) it is always possible to find a new set of Dirac operators $\{b_k\}_{k=1}^N$ such that $\hat{H}$ in terms of $\{b_k\}_{k=1}^N$ reads as

$$\hat{H} = \sum_{k=1}^{N} \epsilon_k (b_k^\dagger b_k - b_k b_k^\dagger), \tag{20}$$

with $\epsilon_k \in \mathbb{R}$ for all $k = 1, 2, \dots, N$ [**?**].
We call Hamiltonians in this form free-free fermion Hamiltonians.
In compact form

$$\hat{H} = \vec{\beta}^\dagger H_D \vec{\beta} \tag{21}$$

with

$$H_D = U^\dagger H U = \begin{pmatrix} -\epsilon_1 & 0 & \dots & & \dots & 0 \\ 0 & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & -\epsilon_N & & & \\ & & & \epsilon_1 & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & \dots & & & \dots 0 & \epsilon_N \end{pmatrix}, \tag{22}$$

where $\vec{\beta}$ is the collection of the Dirac operators $b_k, b_m^\dagger$ ordered as in $\vec{\alpha}$, and U is the Fermionic tranformation that diagonalises the Hamiltonian.
We will always order the eigenvalues in descending order ($\epsilon_1 \geq \epsilon_2 \geq \dots \geq \epsilon_N \geq 0$).

Diagonal form of the Hamiltonian with Majorana operators In terms of Majorana operators the diagonal form of a generic f.q.h. reads as

$$\hat{H} = i \sum_{i=1}^{N} \lambda_i (\tilde{x}_i \tilde{p}_i - \tilde{p}_i \tilde{x}_i). \tag{23}$$

for a set of Majorana operators $\{\tilde{x}_i\}_i$, $\{\tilde{p}_i\}_i$. In compact form

$$\hat{H} = i \vec{s}^{\dagger} h_D \vec{s}, \tag{24}$$

where $\vec{s}$ is the collection of the Majorana operators $\tilde{x}_i, \tilde{p}_j$ ordered as (31) and where

$$h_D = O^T h O = \bigoplus_{i=1}^{N} \begin{pmatrix} 0 & \lambda_i \\ -\lambda_i & 0 \end{pmatrix} \tag{25}$$

is a block diagonal matrix and O the orthogonal transformation that diagonalises the Hamiltonian in the Majorana operators representation. Substituting the definition of Majorana operators (4) into equation (23) and confronting with equation (20) we note that $\epsilon_k = \lambda_k$.

0.2.4 Numerical diagonalisation

As seen in subsection 0.2.3, diagonalising a general f.q.h. $\hat{H}$ reduces to diagonalising (or to block diagonalise in the case of Majorana representation) the matrix H of its compact form.

We are thus interested in finding the Fermionic transformation U that maps H and the vector of Dirac operators $\vec{\alpha}$ respectively to the diagonal matrix $H_D = U^{\dagger} H U$ and to the vector of Dirac operators $\vec{\beta} = U \vec{\alpha}$ such that, in term of $\vec{\beta}$, the Hamiltonian is in the diagonal form (20).

Here we focus on the numerical approach, we diagonalise the Hamiltonian using standard matrix decomposition techniques. For a more physical approach we refer to [**?**].

First step in the diagonalisation procedure is moving to the Majorana representation of H

$$\begin{aligned} \hat{H} &= \vec{\alpha}^{\dagger} H \vec{\alpha} = \vec{\alpha}^{\dagger} \Omega^{\dagger} \Omega H \Omega^{\dagger} \Omega \vec{\alpha} = \\ &= i \vec{r}^{\dagger} h \vec{r}. \end{aligned} \tag{26}$$

The following theorem is a standard result in matrix theory [**?**, **?**]

Theorem 1 (Block diagonal form of skew-symmetric matrices)
Let h be $2N \times 2N$ a real, skew-symmetric matrix. There exists a real special orthogonal matrix O such that

$$h = O h_D O^T, \tag{27}$$

with h_D a block diagonal matrix of the form

$$h_D = \bigoplus_{i=1}^{N} \begin{pmatrix} 0 & \lambda_i \\ -\lambda_i & 0 \end{pmatrix} \tag{28}$$

for real, positive-definite $\{\lambda_i\}_{i=1,\dots,N}$. The non-zero eigenvalues of matrix h are the imaginary numbers $\{\pm i\lambda_i\}_{i=1,\dots,N}$.
For a more general form of this theorem see appendix **??**.

Matrix h in (18) is real, skew-symmetric, thus, using theorem (27) we know there exists an orthogonal transformation O that diagonalises the matrix

$$\hat{H} = i\vec{r}^{\dagger} h\vec{r} = i\vec{r}^{\dagger} OO^{\dagger} hOO^{\dagger}\vec{r} = \tag{29}$$

$$= i\vec{s}^{\dagger} \left(\bigoplus_{i=1}^{N} \begin{pmatrix} 0 & \lambda_i \\ -\lambda_i & 0 \end{pmatrix} \right) \vec{s} = i\vec{s}^{\dagger} h_D \vec{s} \tag{30}$$

That is $\hat{H} = i\sum_{i=0}^{N-1} \lambda_i(\tilde{x}_i\tilde{p}_i - \tilde{p}_i\tilde{x}_i)$ once defined the new collection of Majorana operators $\vec{s} = O\vec{r}$ as

$$\vec{s} = \begin{pmatrix} \tilde{x}_0 \\ \tilde{p}_0 \\ \tilde{x}_1 \\ \tilde{p}_1 \\ \vdots \\ \tilde{x}_{N-1} \\ \tilde{p}_{N-1} \end{pmatrix}. \tag{31}$$

The vector of Majorana operators $\vec{s}$ has a different ordering with respect to the vector $\vec{r}$. We call the order of the operators in $\vec{s}$ an xp ordering and the ordering of the operators in $\vec{r}$ and xx ordering. The transformation matrix $\Omega^{\dagger}$ maps a vector of Majorana operators in xx ordering to a vector of Dirac operators, thus, before being able to move to the Dirac representation we have to reorder the element of vector $\vec{s}$. To do so we use the matrix

$$F_{xp\to xx} = \begin{array}{c} i=0 \\ i=1 \\ \vdots \\ \vdots \\ i=N \\ i=N+1 \\ \vdots \\ i=2N+1 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & \dots & 0 & 0 \\ 0 & \vdots & 1 & \vdots & & & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & & 0 & \vdots \\ \vdots & 0 & 0 & 0 & & & 1 & \vdots \\ \vdots & 1 & 0 & 0 & & & 0 & \vdots \\ \vdots & 0 & \vdots & 1 & & & \vdots & \vdots \\ \vdots & \vdots & \vdots & 0 & & & \vdots & 0 \\ 0 & 0 & 0 & \vdots & \dots & \dots & 0 & 1 \end{pmatrix} \tag{32}$$

that applied to a vector $\vec{s}$ with the xp ordering returns a vector $\vec{r} = F_{xp\to xx}\vec{s}$ with xx ordering. Mapping back to the Dirac representation we obtain the diagonal form of the Hamiltonian in the Dirac operators representation as

$$\hat{H} = i\left(\vec{s}F^{T}_{xp\to xx}\Omega\right)\left(\Omega^{\dagger}F_{xp\to xx}h_D F^{T}_{xp\to xx}\Omega\right)\left(\Omega^{\dagger}F_{xp\to xx}\vec{s}\right) = \tag{33}$$

$$= \sum_{k=1}^{N}\epsilon_k(b^{\dagger}_k b_k - b_k b^{\dagger}_k) = \vec{\beta}^{\dagger}H_D\vec{\beta}. \tag{34}$$

The Fermionic transformation U that diagonalises the Hamiltonian H in the form (22) is

$$U = \Omega^{\dagger}\cdot O\cdot F^{\dagger}_{xp\to xx}\cdot\Omega. \tag{35}$$

F_utilities 0.2.1: Diag_h(H**)**$\to H_D, U$

This function diagonalises H. H_D is the diagonal form with the first half diagonal negative and the second one positive ordered as (22). U is the Fermionic transformation such that: $H = UH_DU^{\dagger}$.

Block-diagonal form of real skew-symmetric matrices The matrix decomposition (27) of theorem 1 is numerically obtained in 3 steps

1. Compute numerically a Schur decomposition (or Schur triangularisation as in [**?**]) of the skew-symmetric matrix h such that: $h = \tilde{O}\tilde{h}_D\tilde{O}^T$. The matrix $\tilde{h}_D$ should be a block-diagonal matrix with each block in the anti-diagonal form

$$\begin{pmatrix} 0 & \tilde{\lambda}_i \\ -\tilde{\lambda}_i & 0 \end{pmatrix}. \tag{36}$$

   It is not guaranteed that the $\tilde{\lambda}_i$ are positive for each i. It is necessary to reorder them as in step 2.

2. Build the orthogonal matrix $S = \bigoplus_{i=1}^{\lfloor N/2\rfloor} s_i$ with

$$s_i = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{37}$$

   if $\tilde{\lambda}_i < 0$ or

$$s_i = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{38}$$

   if $\tilde{\lambda}_i > 0$.

3. The final orthogonal transformation is $O = \tilde{O}S$ such that $h = Oh_DO^T$.

F_utilities 0.2.2: Diag_real_skew(h) $\to h_D, O$

This function implements the algorithm for the block diagonalisation of h a generic skew-symmetric real matrix. h_D is the block-diagonal matrix of (27) and has the following property: it is in the block diagonal form, each 2×2 block is skew-symmetric with the upper-right element positive and real and h_D is in ascending order for the upper diagonal. O is an orthogonal matrix such that: $h = O h_D O^T$.

0.3 Fermionic Gaussian States

0.3.1 Fermionic Gaussian states

Definition 1 (Fermionic Gaussian state)
A state ρ is a Fermionic Gaussian state (f.g.s.) if it can be represented as

$$\rho = \frac{e^{-\hat{H}}}{Z} = \frac{e^{-\vec{\alpha}^\dagger H \vec{\alpha}}}{Z} \tag{39}$$

with $Z := Tr\left[e^{-\hat{H}}\right]$ a normalisation constant and $\hat{H}$ a Fermionic Gaussian Hamiltonian called parent Hamiltonian of ρ.
Every possible value of the norm of the Hamiltonian is admitted, $\left\|\hat{H}\right\|_1 \in [0, +\infty]$. Both extremum values are reached with a single sided limit procedure in the definition of ρ.
All the information about the state is encoded in the $2N \times 2N$ matrix H at the exponential.

Fermionic Gaussian states have an immediate interpretation as thermal Gibbs states of f.q.h.. One can even rescale the parent Hamiltonian as $\hat{\tilde{H}} = \frac{1}{\beta}\hat{H}$ such that $\left\|\hat{\tilde{H}}\right\| = 1$ and $\beta = \frac{1}{\|\hat{H}\|}$. In this way the state reads as $\rho = \frac{e^{-\beta\hat{\tilde{H}}}}{Z}$ with $\beta \in [0, +\infty]$. Since f.g.s are exponential of f.g.h. and f.g.h. are even operator, it follows that f.g.s are even operator.

Single mode Gaussian states Consider the single mode parent Hamiltonian $\hat{H}_1 = \epsilon(b^\dagger b - b b^\dagger)$ of the f.g.s. $\rho = \frac{1}{Z}e^{-\hat{H}_1}$. The explicit representation of ρ on the basis $\{b^\dagger|0\rangle, |0\rangle\}$ is

$$\rho = \begin{pmatrix} 1-f & 0 \\ 0 & f \end{pmatrix} \tag{40}$$

where $f = \langle 0|\rho|0\rangle$ and the two coherences are 0 because we cannot have the odd terms $|0\rangle\langle 1|$ and $|1\rangle\langle 0|$ in the expansion of the even operator ρ (see [**?**, **?**] for a detailed and beautiful analysis of the admitted coherences). Using the polynomial expansion (14) we can see that $f = \mathrm{Tr}\left[\rho b^\dagger b\right] := \langle b^\dagger b\rangle$, that is the occupation of the Fermionic mode, thus a single mode Gaussian state is completely characterised by the occupation $\langle b^\dagger b\rangle$.

0.3.2 Correlation Matrix

We have seen that for any f.q.h. H it is always possible to find a Fermionic transformation U that diagonalises H transforming the Dirac operators vector as $\vec{\beta} = U\vec{\alpha}$. Diagonalising the parent Hamiltonian of a f.g.s. ρ we obtain its decomposition in terms of single-mode thermal states

$$\rho = \frac{e^{-\vec{\beta}^\dagger H_D \vec{\beta}}}{Z} = \frac{1}{Z}\bigotimes_{k=1}^{N} e^{-\epsilon_k(b_k^\dagger b_k - b_k b_k^\dagger)} = \bigotimes_{k=1}^{N} \frac{e^{-\epsilon_k(b_k^\dagger b_k - b_k b_k^\dagger)}}{Z_k}, \tag{41}$$

where $Z_k = \mathrm{Tr}\left[e^{-\epsilon_k(b_k^\dagger b_k - b_k b_k^\dagger)}\right]$.
Each single-mode thermal state is completely characterised by its occupation number, thus ρ is completely characterised by the set of occupations $\{\langle b_i^\dagger b_i\rangle\}_{i=1}^{N}$.
Expressing the occupations in term of the operators $\vec{\alpha} = U^\dagger\vec{\beta}$, we find that every f.g.s. is *completely characterised* by the collection of all the correlators $\Gamma^{a^\dagger a}_{i,j} := \langle a_i^\dagger a_j\rangle$ and $\Gamma^{aa}_{i,j} := \langle a_i a_j\rangle$. We collect these correlators in the so called *correlation matrix*

$$\Gamma := \langle\vec{\alpha}\vec{\alpha}^\dagger\rangle = \begin{pmatrix} \Gamma^{a^\dagger a} & \Gamma^{a^\dagger a^\dagger} \\ \Gamma^{aa} & \Gamma^{aa^\dagger} \end{pmatrix} \tag{42}$$

with $\Gamma^{aa}_{i,j} = -\overline{\Gamma^{a^\dagger a^\dagger}_{i,j}}$ and $\Gamma^{aa^\dagger}_{i,j} = (\mathbb{I} - \Gamma^{a^\dagger a})^\dagger_{i,j}$, where $\overline{A}$ is the conjugate of A. The correlation matrix Γ is Hermitian, Γ^{aa} and $\Gamma^{a^\dagger a^\dagger}$ are skew-symmetric, and $\Gamma^{a^\dagger a}$ and $\Gamma^{aa^\dagger}$ are Hermitian.
Expressed in term of Majorana operator the correlation matrix is defined as

$$\Gamma^{maj} := \langle\vec{r}\vec{r}^\dagger\rangle = \Omega\Gamma\Omega^\dagger. \tag{43}$$

It is interesting observing that, since a f.g.s. is completely described by its correlation matrix, with the spirit of the maximum entropy principle (see [**?**, **?**]), it is possible to equivalently define Fermionic Gaussian states as the states that maximise the von Neumann entropy given the expectation values collected in the correlation matrix.

0.3.3 Covariance matrix

The *covariance matrix* of a f.g.s. is the real, skew-symmetric matrix defined as

$$\gamma := i\mathrm{Tr}\left[\rho[\vec{r}_i, \vec{r}_j]\right], \tag{44}$$

with $[\vec{r}_i, \vec{r}_j]$ the commutator of the two Majorana operators $\vec{r}_i$ and $\vec{r}_j$.
As for the correlation matrix, the covariance matrix of a f.g.s ρ completely describes the states. In fact γ and Γ are related by the equality

$$\gamma = -i\Omega\left(2\Gamma - \mathbb{I}\right)\Omega^\dagger = -i\left(2\Gamma^{maj} - \mathbb{I}\right). \tag{45}$$

In this thesis we will use both the covariance matrix and the correlation matrix approach.

0.3.4 Wick's theorem

As mentioned, f.g.s. are fully characterised by their covariance matrix. This means that it must be possible to obtain the expectation value of every operator X from γ solely. To do so we just need to take the polynomial expansion (12) of X and apply the celebrated Wick's theorem [?] to each monomial term. The Wick's theorem states that for a f.g.s. ρ and a monomial of Majorana operators $r_{q_1} r_{q_2} \dots r_{q_p}$ one has

$$\mathrm{Tr}\left[\rho r_{q_1} r_{q_2} \dots r_{q_p}\right] = \mathrm{Pf}(\gamma_{|_{q_1,q_2,\dots,q_p}}) \tag{46}$$

where $1 \leq q_1 < q_2 < \cdots < q_p \leq 2N$ and $\gamma_{|_{q_1,q_2,\dots,q_p}}$ is the restriction of the covariance matrix to all the two points correlators involving just the Majorana operators $\{r_{q_1}, r_{q_2}, \dots, r_{q_p}\}$ and Pf() is called the *Pfaffian*. Since the Pfaffian is nonvanishing only for a $2N \times 2N$ skew-symmetric matrix [?], it is clear that the expectation value of any odd operators is always zero.

Example

Consider a system composed by 2 Fermionic modes corresponding to the Dirac operators a_1 and a_2. The Majorana operators vector is $\vec{r} = (r_1, r_2, r_3, r_4)$, thus the covariance matrix takes the form

$$\gamma = \begin{pmatrix} 0 & \langle r_1, r_2\rangle & \langle r_1, r_3\rangle & \langle r_1, r_4\rangle \\ -\langle r_1, r_2\rangle & 0 & \langle r_2, r_3\rangle & \langle r_2, r_4\rangle \\ -\langle r_1, r_3\rangle & -\langle r_2, r_3\rangle & 0 & \langle r_3, r_4\rangle \\ -\langle r_1, r_4\rangle & -\langle r_2, r_4\rangle & -\langle r_3, r_4\rangle & 0 \end{pmatrix} \tag{47}$$

where $\langle r_i, r_j\rangle := i\mathrm{Tr}\left[\rho[r_i, r_j]\right]$. Using Wick's theorem we have that

$$\mathrm{Tr}\left[\rho r_1 r_2 r_3 r_4\right] = \mathrm{Pf}\begin{pmatrix} 0 & \langle r_1, r_2\rangle & \langle r_1, r_3\rangle & \langle r_1, r_4\rangle \\ -\langle r_1, r_2\rangle & 0 & \langle r_2, r_3\rangle & \langle r_2, r_4\rangle \\ -\langle r_1, r_3\rangle & -\langle r_2, r_3\rangle & 0 & \langle r_3, r_4\rangle \\ -\langle r_1, r_4\rangle & -\langle r_2, r_4\rangle & -\langle r_3, r_4\rangle & 0 \end{pmatrix} = \tag{48}$$

$$= \langle r_1, r_2\rangle\langle r_3, r_4\rangle - \langle r_1, r_3\rangle\langle r_2, r_4\rangle + \langle r_2, r_3\rangle\langle r_1, r_4\rangle, \tag{49}$$

and

$$\mathrm{Tr}\left[\rho r_2 r_4\right] = \mathrm{Pf}\begin{pmatrix} 0 & \langle r_2, r_4\rangle \\ -\langle r_2, r_4\rangle & 0 \end{pmatrix} = \langle r_2, r_4\rangle, \tag{50}$$

and

$$\mathrm{Tr}\left[\rho r_1 r_2 r_4\right] = \mathrm{Pf}\begin{pmatrix} 0 & \langle r_1, r_2\rangle & \langle r_1, r_4\rangle \\ -\langle r_1, r_2\rangle & 0 & \langle r_2, r_4\rangle \\ -\langle r_1, r_4\rangle & -\langle r_2, r_4\rangle & 0 \end{pmatrix} = 0. \tag{51}$$

0.3.5 Diagonalisation of the correlation matrix

In subsection 0.3.2 we have seen that for any f.g.s. ρ there exists a Fermionic transformation U that diagonalises its parent Hamiltonian . With the new Dirac operators $\vec{\beta}$ the state can be expressed as a tensor product of single mode thermal states

$$\frac{e^{-\vec{\beta}^\dagger H_D \vec{\beta}}}{Z} = \frac{1}{Z}\bigotimes_{k=1}^{N} e^{-\epsilon_k (b_k^\dagger b_k - b_k b_k^\dagger)} = \bigotimes_{k=1}^{N} \frac{e^{-\epsilon_k (b_k^\dagger b_k - b_k b_k^\dagger)}}{Z_k}. \tag{52}$$

with $Z_k = \mathrm{Tr}\left[e^{-\epsilon_k (b_k^\dagger b_k - b_k b_k^\dagger)}\right]$.
Expressed with these operators the correlation matrix is diagonal. If we consider the Fock basis $\{|\vec{k}\rangle\}_{\vec{k}\in\{0,1\}^N}$ built with the action of the operators $\vec{\beta}$ on $|0\rangle$, we have that in this basis ρ assumes a diagonal form. We call U_ρ the unitary transformation that moves from the basis $\{|\vec{x}\rangle\}_{\vec{x}\in\{0,1\}^N}$ to the one of the modes $\{|\vec{k}\rangle\}_{\vec{k}\in\{0,1\}^N}$.
It is easy to see that expressed on this basis ρ has the diagonal form

$$\rho^D = U_\rho^\dagger \rho U_\rho = \bigotimes_{i=1}^{N} \begin{pmatrix} \nu_i & 0 \\ 0 & 1-\nu_i \end{pmatrix}. \tag{53}$$

The same Fermionic transformation U that diagonalises the parent Hamiltonian brings Γ in the diagonal form

$$\Gamma^D = U^\dagger \Gamma U = \begin{pmatrix} \nu_1 & 0 & \dots & & \dots & 0 \\ 0 & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \nu_N & & & \\ & & & 1-\nu_1 & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & \dots & & & \dots 0 & 1-\nu_N \end{pmatrix}, \tag{54}$$

with $\nu_i \in [0,1]$ the occupation number of the i-th free mode. To numerically obtain the diagonal form of the correlation matrix we notice that the covariance matrix γ is a real, skew-symmetric matrix, thus using theorem 1 we know that we can find an orthogonal transformation O such that

$$\gamma = O\gamma^D O^T = O\left(\bigoplus_{i=1}^{N} \begin{pmatrix} 0 & \eta_i \\ -\eta_i & 0 \end{pmatrix}\right) O^T \tag{55}$$

with $\eta_i \in [-\frac{1}{2}, \frac{1}{2}]$.
Following the same procedure of subsection 0.2.4, we can write the diagonal elements of Γ^D as

$$\nu_i = \frac{1}{2} - \eta_i. \tag{56}$$

The elements of H_D and Γ^D are related by the following formulas

$$\epsilon_k = \frac{1}{2}\ln\left(\frac{1-\nu_k}{\nu_k}\right), \tag{57}$$

$$\nu_k = \frac{1}{1+e^{2\epsilon_k}}, \tag{58}$$

with $\nu_k \in [0,1]$ and $\epsilon_k \in [-\infty, +\infty]$, where the boundary values are taken with a limit. The complete calculation can be found in appendix **??**. In (22) we defined all the ϵ_k to be positive, to use the same notation, one just has to exchange b with $\tilde{b}^\dagger$ and $b^\dagger$ with $\tilde{b}$, that is exchanging occupations with vacancies for the mode with ϵ_k negative. This corresponds to switching ν_k with $1-\tilde{\nu}_k$ and $1-\nu_k$ with $\tilde{\nu}_k$.

In general the correlation matrix Γ and the parent Hamiltonian H are related by the formula [**?**, **?**, **?**, **?**]

$$\Gamma = \frac{1}{1+e^{2H}}. \tag{59}$$

F_utilities 0.3.1: Diag_gamma(Γ)$\rightarrow \Gamma^D, U$

This function returns Γ^D, the diagonal form of the Dirac correlation matrix Γ and U the Fermionic transformation such that $\Gamma = U\Gamma^D U\dagger$.

Phisicality of a state It is known that a matrix ρ represents a valid physical density matrix if it is a positive semi-definite Hermitian matrix with trace equal to one. The condition for a matrix Γ to represent a valid physical correlation matrix of a f.g.s. is

$$\Gamma^2 - \Gamma \leq 0, \tag{60}$$

or equivalently

$$\gamma\gamma^\dagger \leq -\mathbb{I}. \tag{61}$$

These conditions are equivalent to the request that all the eigenvalues ν_i of matrix Γ have to belong to the interval $[0,1]$.

Ground states of Fermionic quadratic Hamiltonians Suppose we have a f.q.h. H and that we are interested in obtaining the correlation matrix Γ_0 associated to its ground state $|0\rangle$. In order to obtain Γ_0 we proceed by first finding the Fermionic transformation U that diagonalise H. Since our algorithm associates to each free mode of the diagonalised Hamiltonian a positive energy, in the diagonal basis the ground state is $|0\rangle\langle 0|$. The correlation matrix associated to the state $|0\rangle$ is the block matrix:

$$\Gamma_{|0\rangle\langle 0|} = \begin{pmatrix} 0 & 0 \\ 0 & \mathbb{I}_{N\times N} \end{pmatrix}. \tag{62}$$

To obtain the ground state Γ_0 we just need to move back to the original basis, thus

$$\Gamma_0 = U\Gamma_{|0\rangle\langle 0|}U^\dagger. \tag{63}$$

`F_utilities` 0.3.2: `GS_gamma(`H_D, U`)`$\rightarrow \Gamma_0$

This function returns Γ_0, the ground state of the Hamiltonian $H = UH_DU^\dagger$.

Thermal state of Fermionic quadratic Hamiltonians Suppose we have a f.q.h. H and that we are interested in obtaining the correlation matrix Γ_0 associated to the thermal state $\rho_\beta = \frac{e^{-\beta H}}{\mathrm{Tr}[e^{-\beta H}]}$.

As we did for computing the ground state, we move to the diagonal basis with the Fermionic transformation U. In the diagonal basis the thermal state has the correlation matrix

$$\Gamma^D_\beta = \begin{pmatrix} \frac{1}{1+e^{2\beta\epsilon_1}} & 0 & \dots & & \dots & 0 \\ 0 & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \frac{1}{1+e^{2\beta\epsilon_N}} & & & \\ & & & \frac{1}{1+e^{-2\beta\epsilon_1}} & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & \dots & & & \dots 0 & \frac{1}{1+e^{-2\beta\epsilon_N}} \end{pmatrix}. \tag{64}$$

To obtain the thermal state Γ_β we just need to move back to the original basis, thus

$$\Gamma_\beta = U\Gamma^D_\beta U^\dagger. \tag{65}$$

`F_utilities` 0.3.3: `Thermal_fix_beta(`H_D, U, β`)`$\rightarrow \Gamma_0$

This function returns Γ_β, the termal state at inverse temperature β of the Hamiltonian $H = UH_DU^\dagger$.

`F_utilities` 0.3.4: `Thermal_fix_energy(`H_D, U, E`)`$\rightarrow$ $\Gamma_{\beta(E)}, \beta(E), \Delta(E)$

This function variationally computes and then returns $\Gamma_{\beta(E)}$, the thermal state at inverse temperature $\beta(E)$ of the Hamiltonian $H = UH_DU^\dagger$, and $\beta(E)$ the temperature such that $\mathrm{Tr}\left[\rho_{\beta(E)}H\right] = E$ and $\Delta(E)$ the difference between the required energy E and the actual energy of the state $\Gamma_{\beta(E)}$. It outputs the precision $\Delta(E)$ and $\beta(E)$.

Energy of a Fermionic Gaussian state Consider a f.q.h H and a f.g.s. Γ. The energy of Γ with respect to H is the expectation value $\mathrm{Tr}\left[\hat{H}\rho\right]$ of the associated $\hat{H}$ computed on the associated state ρ.
In order to compute this expectation value one just needs to find the Fermionic transformation U that diagonalises H. With this, one is able to find the energies ϵ_k and the occupations $\langle b_k^\dagger b_k\rangle$ and $\langle b_k b_k^\dagger\rangle$. The correlation matrix Γ is not diagonal in the diagonal basis $\vec{\beta}$ of H, but we are just interested in its diagonal elements.
The energy $E_H(\Gamma)$ of Γ is thus

$$E_H(\Gamma) = \sum_k \epsilon_k(\langle b_k^\dagger b_k\rangle - \langle b_k b_k^\dagger\rangle). \tag{66}$$

F_utilities 0.3.5: Energy(Γ, H_D, U**)**$\rightarrow E_H(\Gamma)$

This function returns $E_H(\Gamma)$ the energy of the state Γ calculated with H. Matrices H_D and U are the output of `Diag_h(H)`.

0.3.6 Eigenvalues of ρ and eigenvalues of Γ

We have seen that the diagonal form of the correlation matrix Γ and of the density matrix ρ of a f.g.s. can be obtained respectively with a Fermionic transformation U and a unitary operation U_ρ. The Fock basis in which ρ is diagonal is the one generated by the set of operators that expresses Γ in diagonal form.
In these two basis ρ and Γ assume the forms

$$\Gamma^D = \begin{pmatrix} \nu_1 & 0 & \dots & & \dots & 0 \\ 0 & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \nu_N & & & \\ & & & 1-\nu_1 & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & \dots & & & \dots 0 & 1-\nu_N \end{pmatrix},$$

$$\rho^D = \bigotimes_{k=1}^{N} \begin{pmatrix} \nu_k & 0 \\ 0 & 1-\nu_k \end{pmatrix} = \begin{pmatrix} \pi_{\vec{0}} & 0 & \dots \\ 0 & \ddots & \vdots \\ \vdots & \dots & \pi_{\vec{1}} \end{pmatrix}. \tag{67}$$

Thus if we denote each of the 2^N eigenvalues $\pi_{\vec{x}}$ of ρ with a binary string $\vec{x} \in \{0,1\}^N$ we have that

$$\pi_{\vec{x}} = \prod_{k=1}^{N} \left(\vec{x}_k\nu_i + (1-\vec{x}_k)(1-\nu_k)\right). \tag{68}$$

It is evocative changing the order of the Dirac operators in the representation of Γ^D

$$\vec{\beta} = \left(\begin{array}{c} b_0^\dagger \\ \vdots \\ b_{N-1}^\dagger \\ b_0 \\ \vdots \\ b_{N-1} \end{array} \right) \qquad \rightarrow \qquad \vec{\tilde{\beta}} = \left(\begin{array}{c} b_0^\dagger \\ b_0 \\ b_1^\dagger \\ b_1 \\ \vdots \\ b_N^\dagger \\ b_N \end{array} \right), \tag{69}$$

this can be easily done with the Fermionic transformation $\tilde{\Gamma}^D = F^\dagger_{xp\rightarrow xx} \Gamma^D F_{xp\rightarrow xx}$. With this ordering we have

$$\tilde{\Gamma}^D = \bigoplus_{k=1}^{N} \left(\begin{array}{cc} \nu_k & 0 \\ 0 & 1-\nu_k \end{array} \right), \qquad \rho^D = \bigotimes_{k=1}^{N} \left(\begin{array}{cc} \nu_k & 0 \\ 0 & 1-\nu_k \end{array} \right). \tag{70}$$

To tensor product of density matrices corresponds a direct sum of correlation matrices.

F_utilities 0.3.6: Eigenvalues_of_rho(Γ)$\rightarrow \vec{\nu}$

This function returns the eigenvalues of the correlation matrix ρ associated to the Fermionic Gaussian state with Dirac correlation matrix Γ.

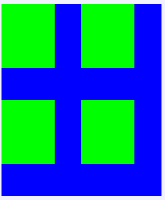
0.3.7 Reduced density matrix and tensor product of Fermionic Gaussian states

Trying to define a partial trace over Fermionic modes subspaces one soon faces what is often called the "partial trace ambiguity" [**?**, **?**] (see also the end of appendix **??**).
In the case of Fermionic Gaussian states, though, this is a much simpler task. Any reduced state formalism has to satisfy the simple criterion that the reduced density operator must contain all the information about the subsystem that can be obtained from the global state when measurements are performed only on the respective subsystem alone [**?**, **?**].
With Wick's theorem in mind it is easy to see that the correlation matrix of the reduced state on the modes $i_1, \ldots, i_m$ is just the correlation matrix $\Gamma|_{\{i_1,\ldots,i_m\}}$ and that the reduced state of a f.g.s. is a f.g.s. too.

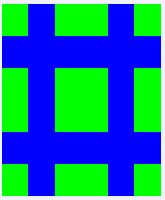
F_utilities 0.3.7: Reduce_gamma(Γ,m,i_1)$\rightarrow$ $\Gamma|_{\{i_1,\ldots,i_m\}}$

This function takes a Dirac correlation matrix Γ, a dimension of the partition m and the initial site of the partition i_1 and return $\Gamma|_{\{i_1,\ldots,i_m\}}$, the reduced correlation matrix on the contiguous modes $\{i_1,\ldots,i_m\}$ where $i_m = i_1 + m$ and periodic boundary conditions are always assumed. Examples: the green elements of the matrix $M_{6\times 6}$ are the ones returned by the function calls.

Reduce_gamma($M_{6\times 6}$,2,1)$\rightarrow$



Reduce_gamma($M_{6\times 6}$,2,3)$\rightarrow$



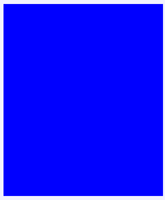
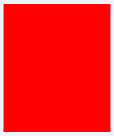
The correlation matrix $\Gamma_{A,B}$ of the tensor product of two f.g.s. Γ_A and Γ_B is obtained simply by collecting all the elements of Γ_A and Γ_B in a single well ordered correlation matrix $\Gamma_{A,B}$. The code for obtaining $\Gamma_{A,B}$ from Γ_A and Γ_B is

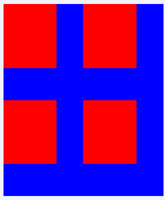
```
D_A = size(Gamma_A,1);
D_B = size(Gamma_B,1);
D   = D_A+D_B;

Gamma_AB = zeros(Complex{Float64}, D,D);
Gamma_AB = Inject_gamma(Gamma_AB,Gamma_A,1);
Gamma_AB = Inject_gamma(Gamma_AB,Gamma_B,D_A+1);
```

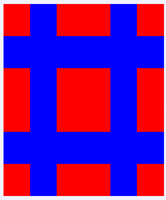
This code makes use of the function `Inject_gamma`.

F_utilities 0.3.8: Inject_gamma(Γ, Γ_{inj}, i)$\rightarrow \Gamma_{comp}$

This function takes a $2N \times 2N$ matrix Γ and a $2n \times 2n$ matrix Γ_{inj} with $n \leq N$. It returns the $2N \times 2N$ matrix Γ_{comp} as shown in the pictures.

$\Gamma =$ , $\Gamma_{inj} =$ 

`Inject_gamma(`$\Gamma, \Gamma_{inj}, 1) \rightarrow$ ,

`Inject_gamma(`$\Gamma, \Gamma_{inj}, 3) \rightarrow$ .

In the last example it is clear the systems behave with periodic boundary conditions.
If Γ is the correlation matrix of a f.g.s ρ and Γ_{inj} is the correlation matrix of a f.g.s ρ_{inj} then Γ_{comp} is the correlation matrix of the state $\mathrm{Tr}_{\mathtt{i},\ldots,\mathtt{i+n-1}}[\rho] \otimes \rho_{inj}$.

It is clear that with the ordering (69), the tensor product of two f.g.s. corresponds to the direct sum of their correlation matrices

$$\rho_{A,B} = \rho_A \otimes \rho_B \ \rightarrow \ \tilde{\Gamma}_{A,B} = \tilde{\Gamma}_A \oplus \tilde{\Gamma}_B. \tag{71}$$

0.3.8 Correlation matrices of translational invariant states

We consider a state ρ of a system of N sites and all its reduced density matrices ρ_A, where A is any possible set of sites of the system. We denote with $A + m$ the set of sites $A + m = \{j = i + m | i \text{ is a site of } A\}$, that is a translation of all site of A by m sites. When we will assume Periodic Boundary Conditions (PBC) we will allow for translations "over the border" of the system, in the sense that when $i + m > N$ (or $i + m < 1$) we will substitute it with $i + m \rightarrow \mathrm{mod}(i + m - 1, N + 1) + 1$. This is interpreted as connecting the first site with the last site of the system. Thus for PBC all translations are allowed. When we will assume Open Boundary Conditions (OBC) only translations within the system. This means that if $i \in A$ and $i+m > N$ (or $i+m < 1$), then the subset $A+m$ is not an allowed subset of sites.

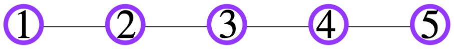


Figure 1: System with Open Boundary Conditions. If the state is translational invariant, then the reduced density matrix on sites 1 and 2 is the same as the one on sites 3 and 4, but is different from the one on sites 5 and 1

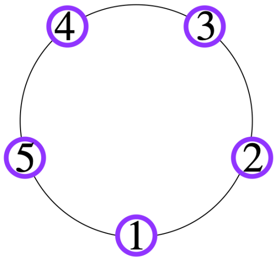


Figure 2: System with Periodic Boundary Conditions. If the state is translational invariant then the reduced density matrix on sites 1 and 2 or sites 3 and 4 or even sites 5 and 1 are all the same

A *translational invariant state* is a state such that for every A we have $\rho_A = \rho_{A+m}$ for every allowed m.
This property easily translates to correlation matrices of states. For the two point correlators of a translational invariant state we have that $\langle a_j^\dagger a_l \rangle = \langle a_{j+m}^\dagger a_{l+m} \rangle$ and $\langle a_j a_l \rangle = \langle a_{j+m} a_{l+m} \rangle$ for every m. The specific correlator is thus individuated just by the difference of the sites of the first and second operator $\Delta := l - j$ with $\Delta \in [-(N-1), N-1]$. Using this, we substitute $\langle a_j^\dagger a_l \rangle \to \langle a^\dagger a \rangle_\Delta$ and analogously $\langle a_j a_l \rangle \to \langle aa \rangle_\Delta$. We now focus on $\Gamma^{a^\dagger a}$, explicitly expressing it, we have

$$\Gamma^{a^\dagger a} \begin{pmatrix} \langle a^\dagger a \rangle_0 & \langle a^\dagger a \rangle_1 & \langle a^\dagger a \rangle_2 & \dots & \langle a^\dagger a \rangle_{N-1} \\ \langle a^\dagger a \rangle_{-1} & \langle a^\dagger a \rangle_0 & \langle a^\dagger a \rangle_1 & \dots & \langle a^\dagger a \rangle_{N-2} \\ \langle a^\dagger a \rangle_{-2} & \langle a^\dagger a \rangle_{-1} & \langle a^\dagger a \rangle_0 & \dots & \langle a^\dagger a \rangle_{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \langle a^\dagger a \rangle_{-(N-1)} & \langle a^\dagger a \rangle_{-(N-2)} & \langle a^\dagger a \rangle_{-(N-3)} & \dots & \langle a^\dagger a \rangle_0 \end{pmatrix}. \tag{72}$$

Matrix with this structure are called *Toeplitz matrices.*
If we further require the system to have PBC, we have that the parameter Δ is restricted to the range $[0, N-1]$. Consider for example the specific instance of the correlator $\langle a^\dagger a \rangle_{1-N} = \langle a_N^\dagger a_1 \rangle$, because of the translational invariance property of the system and because of the PBC we know that $\langle a_N^\dagger a_1 \rangle =$

$\langle a^\dagger_{N+1} a_{1+1}\rangle = \langle a^\dagger_1 a_2\rangle = \langle a^\dagger a\rangle_1$ (see figure 2).
With PBC, $\Gamma^{a^\dagger a}$ has the form

$$\Gamma^{a^\dagger a} = \begin{pmatrix} \langle a^\dagger a\rangle_0 & \langle a^\dagger a\rangle_1 & \langle a^\dagger a\rangle_2 & \dots & \langle a^\dagger a\rangle_{N-1} \\ \langle a^\dagger a\rangle_{N-1} & \langle a^\dagger a\rangle_0 & \langle a^\dagger a\rangle_1 & \dots & \langle a^\dagger a\rangle_{N-2} \\ \langle a^\dagger a\rangle_{N-2} & \langle a^\dagger a\rangle_{N-1} & \langle a^\dagger a\rangle_0 & \dots & \langle a^\dagger a\rangle_{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \langle a^\dagger a\rangle_1 & \langle a^\dagger a\rangle_2 & \langle a^\dagger a\rangle_3 & \dots & \langle a^\dagger a\rangle_0 \end{pmatrix}. \tag{73}$$

We see that $\Gamma^{a^\dagger a}$ is the circulant matrix (see appendix **??**) characterised by the circulant vector $\langle \vec{a^\dagger a}\rangle = \left(\langle a^\dagger a\rangle_0, \langle a^\dagger a\rangle_1, \langle a^\dagger a\rangle_2, \dots, \langle a^\dagger a\rangle_{N-1}\right)$. Following the same reasoning, we see that $\Gamma^{aa^\dagger}$ is a circulant matrix characterised by the circulant vector $\langle \vec{aa^\dagger}\rangle = \left(\langle aa^\dagger\rangle_0, \langle aa^\dagger\rangle_1, \langle aa^\dagger\rangle_2, \dots, \langle aa^\dagger\rangle_{N-1}\right)$. Matrices Γ^{aa} and $\Gamma^{a^\dagger a^\dagger}$ are circulant skew-simmetric matrices, often called *skew-circulant matrices.* If N is even Γ^{aa} and $\Gamma^{a^\dagger a^\dagger}$ are specified by the circulant vectors

$$\langle \vec{aa}\rangle = \left(\langle aa\rangle_0, \langle aa\rangle_1, \langle aa\rangle_2, \dots, \langle aa\rangle_{\frac{N}{2}-1}, 0, -\langle aa\rangle_{\frac{N}{2}-1}, -\langle aa\rangle_{\frac{N}{2}-2}, \dots, \langle aa\rangle_1\right)$$

and

$$\langle \vec{a^\dagger a^\dagger}\rangle = \left(\langle a^\dagger a^\dagger\rangle_0, \langle a^\dagger a^\dagger\rangle_1, \langle a^\dagger a^\dagger\rangle_2, \dots, \langle a^\dagger a^\dagger\rangle_{\frac{N}{2}-1}, 0, -\langle a^\dagger a^\dagger\rangle_{\frac{N}{2}-1}, -\langle a^\dagger a^\dagger\rangle_{\frac{N}{2}-2}, \dots, \langle a^\dagger a^\dagger\rangle_1\right).$$

If N is odd Γ^{aa} and $\Gamma^{a^\dagger a^\dagger}$ are specified by the circulant vectors

$$\langle \vec{aa}\rangle = \left(\langle aa\rangle_0, \langle aa\rangle_1, \langle aa\rangle_2, \dots, \langle aa\rangle_{\frac{N-1}{2}}, -\langle aa\rangle_{\frac{N-1}{2}-1}, -\langle aa\rangle_{\frac{N-1}{2}-2}, \dots, \langle aa\rangle_1\right)$$

and

$$\langle \vec{a^\dagger a^\dagger}\rangle = \left(\langle a^\dagger a^\dagger\rangle_0, \langle a^\dagger a^\dagger\rangle_1, \langle a^\dagger a^\dagger\rangle_2, \dots, \langle a^\dagger a^\dagger\rangle_{\frac{N-1}{2}}, -\langle a^\dagger a^\dagger\rangle_{\frac{N-1}{2}-1}, -\langle a^\dagger a^\dagger\rangle_{\frac{N-1}{2}-2}, \dots, \langle a^\dagger a^\dagger\rangle_1\right).$$

Eigenvalues using the properties of circulant-matrices In appendix **??** we show the general form of the eigenvalues of a circulant matrix. For Γ^{aa},$\Gamma^{a^\dagger a^\dagger}$matrices $\Gamma^{a^\dagger a}$,$\Gamma^{aa^\dagger}$ we have that their respective eigenvalues $\lambda_k^{a^\dagger a}$,$\lambda_k^{aa^\dagger}$ are

$$\lambda_k^{a^\dagger a} = \sum_{\Delta=0}^{N-1} e^{i\frac{2\pi}{N}\Delta k}\langle a^\dagger a\rangle_\Delta \qquad \lambda_k^{aa^\dagger} = \sum_{\Delta=0}^{N-1} e^{i\frac{2\pi}{N}\Delta k}\langle aa^\dagger\rangle_\Delta. \tag{74}$$

For matrices Γ^{aa},$\Gamma^{a^\dagger a^\dagger}$ we have that their respective eigenvalues λ_k^{aa},$\lambda_k^{a^\dagger a^\dagger}$

are

$$\begin{aligned} \lambda_k^{aa} &= \begin{cases} 2\sum_{\Delta=0}^{\frac{N}{2}-1} e^{i\frac{2\pi}{N}k\Delta}\langle aa\rangle_\Delta & \text{if } N \text{ even} \\ (1+e^{-i\frac{\pi}{N}})\sum_{\Delta=0}^{\frac{N}{2}-1} e^{i\frac{2\pi}{N}k\Delta}\langle aa\rangle_\Delta & \text{if } N \text{ odd} \end{cases}, \\ \lambda_k^{a^\dagger a^\dagger} &= \begin{cases} 2\sum_{\Delta=0}^{\frac{N}{2}-1} e^{i\frac{2\pi}{N}k\Delta}\langle a^\dagger a^\dagger\rangle_\Delta & \text{if } N \text{ even} \\ (1+e^{-i\frac{\pi}{N}})\sum_{\Delta=0}^{\frac{N}{2}-1} e^{i\frac{2\pi}{N}k\Delta}\langle a^\dagger a^\dagger\rangle_\Delta & \text{if } N \text{ odd} \end{cases}. \end{aligned} \tag{75}$$

We notice that the eigenvalues of Γ^{aa},$\Gamma^{a^\dagger a^\dagger}$ comes in pairs $\lambda_k^{aa} = -\lambda_{k+\lceil\frac{N}{2}\rceil}^{aa}$ and $\lambda_k^{a^\dagger a^\dagger} = -\lambda_{k+\lceil\frac{N}{2}\rceil}^{a^\dagger a^\dagger}$ as expected from the property of skew-symmetric matrices (see appendix **??**).

Eigenvalues using the Fourier transform on a linear lattice We introduce the Fourier transforms on a linear lattice

$$f_k = \frac{1}{\sqrt{N}}\sum_{j=1}^{N} e^{i\frac{2\pi}{N}kj}a_j, \qquad f_k^\dagger = \frac{1}{\sqrt{N}}\sum_{j=1}^{N} e^{-i\frac{2\pi}{N}kj}a_j^\dagger, \tag{76}$$

with inverse transformations

$$a_j = \frac{1}{\sqrt{N}}\sum_{k=1}^{N} e^{-i\frac{2\pi}{N}kj}f_k, \qquad a_j^\dagger = \frac{1}{\sqrt{N}}\sum_{k=1}^{N} e^{i\frac{2\pi}{N}kj}f_k^\dagger. \tag{77}$$

It is easy to see that the Fourier modes $\{f_k, f_k^\dagger\}_k$ obey to the CAR and are valid Dirac operators.

Now we perform the substitutions (0.5.1) in the expression of $\Gamma^{a^\dagger a}$ and we further exploit the translational invariance ($\langle a^\dagger a\rangle\Delta = \frac{1}{N}\sum_{j=1}^{N}\langle a_j^\dagger a_{j+\Delta}\rangle$) to obtain

$$\langle a^\dagger a\rangle_\Delta = \frac{1}{N^2}\sum_{j}^{N}\sum_{k,k'} e^{i\frac{2\pi}{N}k'\Delta}e^{i\frac{2\pi}{N}(k-k')j}\langle f_k^\dagger f_{k'}\rangle. \tag{78}$$

Collecting the Kronecker delta (see appendix **??**) we can express the elements of $\Gamma^{a^\dagger a}$ as

$$\langle a^\dagger a\rangle_\Delta = \frac{1}{N}\sum_{k=1}^{N} e^{-i\frac{2\pi}{N}k\Delta}\langle f_k^\dagger f_k\rangle. \tag{79}$$

With the same procedure we obtain

$$\begin{aligned} \langle a^\dagger a\rangle_\Delta &= \frac{1}{N}\sum_{k=1}^{N} e^{-i\frac{2\pi}{N}k\Delta}\langle f_k^\dagger f_k\rangle, \qquad & \langle aa^\dagger\rangle_\Delta &= \frac{1}{N}\sum_{k=1}^{N} e^{-i\frac{2\pi}{N}k\Delta}\langle f_k f_k^\dagger\rangle \\ \langle aa\rangle_\Delta &= \frac{1}{N}\sum_{k=1}^{N} e^{-i\frac{2\pi}{N}k\Delta}\langle f_k f_{N-k}\rangle, \qquad & \langle a^\dagger a^\dagger\rangle_\Delta &= \frac{1}{N}\sum_{k=1}^{N} e^{-i\frac{2\pi}{N}k\Delta}\langle f_{N-k}^\dagger f_k^\dagger\rangle. \end{aligned} \tag{80}$$

with inverse transformations

$$\begin{aligned} \langle f_k^\dagger f_k\rangle &= \sum_{\Delta=1}^{N} e^{i\frac{2\pi}{N}k\Delta}\langle a^\dagger a\rangle_\Delta, & \langle f_k f_k^\dagger\rangle &= \sum_{\Delta=1}^{N} e^{i\frac{2\pi}{N}k\Delta}\langle a a^\dagger\rangle_\Delta \\ \langle f_k f_{N-k}\rangle &= \sum_{\Delta=1}^{N} e^{i\frac{2\pi}{N}k\Delta}\langle a^\dagger a\rangle_\Delta, & \langle f_k^\dagger f_{N-k}^\dagger\rangle &= \sum_{\Delta=1}^{N} e^{-i\frac{2\pi}{N}k\Delta}\langle a a^\dagger\rangle_\Delta. \end{aligned} \tag{81}$$

We can easily identify

$$\begin{aligned} \lambda_k^{a^\dagger a} &= \langle f_k^\dagger f_k\rangle, & \lambda_k^{aa^\dagger} &= \langle f_k f_k^\dagger\rangle \\ \lambda_k^{aa} &= \langle f_k f_{N-k}\rangle, & \lambda_k^{a^\dagger a^\dagger} &= \langle f_k^\dagger f_{N-k}^\dagger\rangle. \end{aligned} \tag{82}$$

At last, we note that the Fourier transform does not mix creation and annihilation operators and can be implemented directly on the vector of Dirac operators $\vec{\alpha}$ with the Fermionic transformation U_ω that has the block diagonal form

$$U_\omega = \begin{pmatrix} W & 0 \\ 0 & \bar{W} \end{pmatrix}, \tag{83}$$

where W is the matrix implementing the discrete Fourier transform (see appendix **??**) and it acts separately on the creation and annihilation operators sectors of $\vec{\alpha}$.

For an example of diagonalisation of translational invariant matrices see e.g. subsection 0.4

F_utilities 0.3.9: Build_Fourier_matrix(N)→ U_ω

This function returns the Fermionic transformation U_ω for a system of N sites.

0.3.9 Product Rule

It will result useful to compute the product $\rho = \rho_1\rho_2$ of the density matrices of two Fermionic Gaussian states. We observe that the commutator of two quadratic terms of Majorana operators $\vec{r}$ is always again a quadratic operator or zero

$$[r_i r_j, r_k r_l] = \delta_{k,i} r_l r_j + \delta_{k,j} r_i r_l - \delta_{i,l} r_k r_j - \delta_{l,j} r_i r_k. \tag{84}$$

This is also valid for Dirac operators. We say that the commutator of two monomials of Dirac operators of degree at most 2 is a polynomial of Dirac operators of degree at most 2. Using this observation together with the Baker-Campbell-Hausdorff formula (equation B.C.H.0 in appendix **??**), it is easy to see that ρ, the product of two f.g.s., is always a f.g.s.

$$\rho = \frac{e^{-\hat{H}}}{Z}, \tag{85}$$

with $\hat{H}$ given by the B.C.H.0.
It is possible to derive the covariance matrix γ of ρ directly from the covariance matrices γ_1 and γ_2 of the states ρ_1 and ρ_2. This formula appears in [?] where a more detailed description, considering even pathological cases, is given. If we assume that $\mathbb{I} - \gamma_1$ and $\mathbb{I} - \gamma_2$ are invertible then we have

$$\gamma = \mathbb{I} - (\mathbb{I} - \gamma_2)\frac{1}{\mathbb{I} + \gamma_1\gamma_2}(\mathbb{I} - \gamma_1). \tag{86}$$

F_utilities 0.3.10: `Product(`Γ_1, Γ_2`)`$\rightarrow \Gamma$

This function returns the correlation matrix Γ corresponding to the f.g.s $\rho = \rho_1\rho_2$, where ρ_1 and ρ_2 are characterised by the correlation matrices Γ_1 and Γ_2.

0.3.10 Information measures

Von Neumann Entropies The von Neumann entropy of a quantum state described by the density matrix ρ is

$$S(\rho) = -\mathrm{Tr}\left[\rho \ln(\rho)\right]. \tag{87}$$

In terms of the eigenvalues λ of ρ, the von Neumann entropy reads as

$$S(\rho) = -\sum_{\lambda} \lambda \ln(\lambda) \tag{88}$$

If ρ is a f.g.s. of a system with N sites, since the von Neumann entropy is invariant under unitary transformation of the state, substituting in (87) the product form (70) and using the fact that the von Neumann entropy is additive for product states, the von Neumann entropy becomes a function of the eigenvalues ν_i of the correlation matrix Γ and it is the sum of just $2N$ terms

$$S(\Gamma) \equiv S(\rho) = -\sum_{k=1}^{N} \left[\nu_k \ln(\nu_k) + (1-\nu_k)\ln(1-\nu_k)\right]. \tag{89}$$

F_utilities 0.3.11: `VN_entropy(`Γ`)`$\rightarrow S$

This function returns S, the `Float64` value of the von Neumann Entropy of the state described by the Dirac correlation matrix Γ.

Purity A state is pure if its correlation matrix Γ is such that [?]

$$\Gamma^2 = \Gamma, \tag{90}$$

or, equivalently,

$$\gamma^2 = -\mathbb{I} \tag{91}$$

The purity of a state ρ is defined as

$$\text{Purity}(\rho) \equiv Tr\left[\rho^2\right]. \tag{92}$$

We have that:

$$\text{Purity}(\rho) = \prod_{i=1}^{N-1} \frac{1}{2}\left(1 + \tanh(\epsilon_i)^2\right), \tag{93}$$

$$\text{Purity}(\Gamma) = \prod_{i=1}^{N-1} \left(2\left(\nu_i - 1\right)\nu_i + 1\right), \tag{94}$$

$$\text{Purity}(\gamma) = \prod_{i=1}^{N-1} \left(2\eta_i^2 + \frac{1}{2}\right), \tag{95}$$

the value of the purity is the same if computed with any of these equations. For more details see appendix **??**.

`F_utilities` 0.3.12: `Purity(`Γ`)`$\rightarrow p$

This function returns p the purity of the Fermionic Gaussian state with Dirac correlation matrix Γ.

Entanglement Contour In 2014 Chen and Vidal [**?**] introduced the entanglement contour "a tool for identifyng which real-space degrees of freedom contribute, and how much, to the entanglement of a region A with the rest of the system B". We consider the state of a system on a chain of N sites, we divide the chain into two complementary partitions, partition A and partition B. Now suppose partitions A and B are entangled and that there exists a measure $\mathcal{E}(A, B)$ that quantifies the amount of entanglement between A and B. The entanglement contour $c_A(i)$ of partition A tells us how much each site i of partition A contributes to the total amount of entanglement betwen A and B. Furthermore summing $c_A(i)$ over all the sites of A one should obtain exactly $\mathcal{E}(A, B)$.

Chen and Vidal state five reasonable properties that define when a function is a contour function. In the same paper they show that these five properties do not identify a unique contour function, but instead a class of functions. In chapter **??** we presented these five properties and introduced the general concept of entanglement contour, here we are going to focus on a specific entanglement contour defined for Fermionic Gaussian states. First of all we restrict to pure states. For a pure state, it is known that a good measure of entanglement between two complementary partition A and B is the entanglement entropy, that is the von Neumann entropy $\mathcal{E}(A, B) = S(A)$ of the reduced state on A.

We consider an Hilbert space $\mathcal{H}$ divided in the two complementary partitions

$\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, each partition with N_A and N_B sites respectively. The Schmidt decomposition (see section **??**) of a pure state $|\psi^{A,B}\rangle$ in $\mathcal{H}$ is

$$|\psi^{A,B}\rangle = \sum_i \sqrt{p_i}|\psi_i^A\rangle \otimes |\psi_i^B\rangle, \tag{96}$$

with $p_i \geq 0$, $\sum_i p_i = 1$ and

$$\rho^A \equiv Tr_B\left[|\psi^{A,B}\rangle\langle\psi^{A,B}|\right] = \sum_i p_i|\psi_i^A\rangle\langle\psi_i^A|. \tag{97}$$

The entanglement entropy for this choice of partition is thus $S(A) = -\sum_i p_i ln(p_i)$. Factorising the Hilbert space $\mathcal{H}_A$ in its tensor product structure $\mathcal{H}_A = \bigotimes_{j\in A}\mathcal{H}_j$, we individuate in each local Hilbert space $\mathcal{H}_j$ a site of the partition A. We remind that ρ^A cannot be expressed as a product state over this factorisation of $\mathcal{H}_A$ and that the von Neumann entropy is not additive. Thus the von Neumann entropy computed on each site is not a good entanglement contour function.
We know from 0.3.7 that ρ_A is a f.g.s., thus we can express the entanglement entropy $S(A)$ as the sum of the von Neumman entropy of each mode in A,

$$S(A) = \sum_{k=1}^{N_A} S_k = -\sum_{k=1}^{N_A}\left[\nu_k \ln(\nu_k) + (1-\nu_k)\ln(1-\nu_k)\right]. \tag{98}$$

Each mode k, associated to the Dirac operators $\beta_k = b_k^\dagger$, $\beta_{k+N_A} = b_k$, is connected to the real space modes associated the Dirac operators $\alpha_i = a_i^\dagger$, $\alpha_{i+N_A} = a_i$ by the Fermionic transformation U such that

$$\beta_k = \sum_{i=1}^{N_A} U_{k,i}\alpha_i. \tag{99}$$

We want to use this equation to find how much a fixed mode k contributes to a fixed site i. We call this contribution $p_i(k)$ and we define it as

$$p_i(k) \coloneqq \frac{1}{2}\left[|U_{k,i}|^2 + |U_{k+N_A,i+N_A}|^2 + |U_{k,i+N_A}|^2 + |U_{k+N_A,i}|^2\right]. \tag{100}$$

The entanglement contour for partition A is thus defined as

$$c_A(i) \coloneqq \sum_{k=1}^{N_A} p_i(k)S_k. \tag{101}$$

It is easy to see that each of the $p_i(k)$ is positive and that

$$\sum_{k=1}^{N_A} p_i(k) = 1, \tag{102}$$

as

$$\sum_{k=1}^{N_A} p_i(k) = \frac{1}{2}\left[\sum_{l=1}^{2N_A} U_{i,l}U^*_{l,i} + U_{i+N_A,l}U^*_{l,i+N_A}\right] = \frac{1}{2}\left((UU^\dagger)_{i,i} + (UU^\dagger)_{i+N_A,i+v}\right) = 1,$$

since U is unitary. Thus one has the desired property

$$\sum_{i=1}^{N_A} c_A(i) = S(A). \tag{103}$$

F_utilities 0.3.13: Contour(Γ_A)$\rightarrow \vec{c_A}$

This function returns the vector $\vec{c_A}_i = c_A(i)$ of the entanglement contour of the correlation matrix Γ_A.

0.3.11 Examples

We will use the function

F_utilities 0.3.14: Random_NNHamiltonianr(N)$\rightarrow H$

Generate a random f.q.h. Hamiltonian for a system of N sites with just nearest neighbour interactions.

Computing the energies of H In this program we compute the energies ϵ_k and the ground state of a random nearest neighbours Hamiltonian on a linear lattice of $N = 64$ sites generated with the function `Random_NNHamiltonianr(64)`.

```
using F_utilities;
using PyPlot;
using LinearAlgebra;

const Fu = F_utilities;

N = 64;

#Generate and diagonalise the hamiltonian
H = Fu.Random_NNhamiltonian(N)
H_D, U_D = Fu.Diag_h(H)

#Print the energy modes epsilon_k
figure("Energies")
plot(1:N,diag(H_D)[1:N])
xlabel(L"$k$")
ylabel(L"$\epsilon_k$")
```

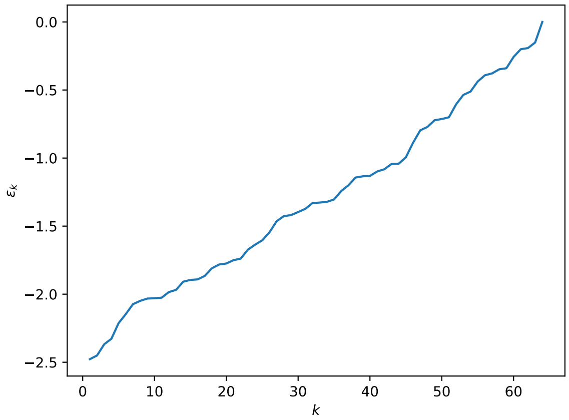


Figure 3: Output of examples 0.3.11. In the plot are represented the eigenvalues of a random nearest-neighbour Hamiltonian computed with the code `Random_NNHamiltonian(N)` for a system with $N = 64$ sites.

Computing the entanglement contour of a partition of a ground state

In this program we compute the entanglement contour and the entropy of a partition of $N_A = 32$ sites in the bulk of the ground state of a random nearest neighbours Hamiltonian on a linear lattice of $N = 64$ sites.

```
using F_utilities;
using PyPlot;
using LinearAlgebra;

const Fu = F_utilities;

N = 64;
H = Fu.Random_NNhamiltonian(N);
H_D, U_D = Fu.Diag_h(H);

Gamma = Fu.GS_Gamma(H_D, U_D);
println("The energy of the ground state is: ", Fu.
    ↪ Energy_fermions(Gamma,H_D, U_D));

N_A = 32;
#I consider the reduced state over the sites 17,18,...,48
Gamma_A = Fu.Reduce_gamma(Gamma,N_A,17);
#I compute the entangement entropy
S_A = Fu.VN_entropy(Gamma_A);
#I compute the contour of partition A
c_A = Fu.Contour(Gamma_A);
```

```
lbl_title   = string(L"$S(A) = $", S_A);
lbl_legend  = string(L"$\sum_{i=1}^{N_A} c_A(i) = $", sum(c_A
    ↪ ));
figure("Contour of A")
title(lbl_title)
plot(1:N_A, c_A, marker="o", label=lbl_legend);
xlabel("i")
ylabel(L"$c_A(i)$")
legend();
```

Output:

```
The energy of the ground state is:  -83.1750144099933
```

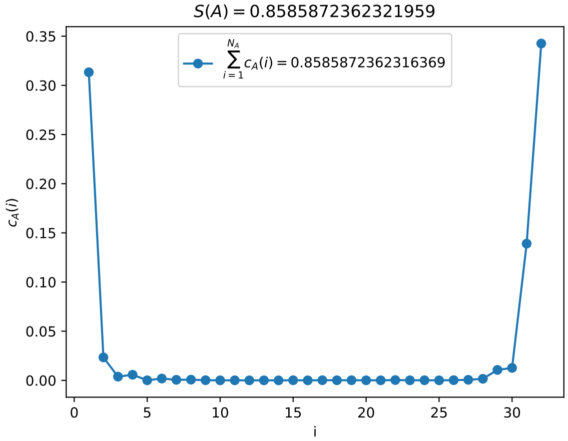


Figure 4: Output of examples 0.3.11. In the plot it is represented the entanglement contour of the reduced density matrix of the first 32 contiguous site of a linear chain, when the global state is the ground state of a nearest-neighbour Hamiltonian on a linear chain of $N = 64$ sites. Note how for a nearest neighbour Hamiltonian, the contour is higher on the boundary of the partition.

0.3.12 Time Evolution

We learned about Hamiltonians and states. Now it is time to put these two ingredients together and finally talk about the unitary evolution of Fermionic Gaussian states.

We start stating that *the space of Fermionic Gaussian states is closed under evolution induced by Fermionic quadratic Hamiltonians.*

The best way for seeing this is using the Majorana operators representation. We consider a general f.q.h. $\hat{H}$ and a generic f.g.s. $\rho = \frac{e^{-\hat{H}_\rho}}{Z}$ with both $\hat{H}$ and $\hat{H}_\rho$ of the form (18). Using standard notation we call $\rho(t)$ the state ρ at time t

defined as

$$\rho(t) := e^{-it\hat{H}}\rho e^{it\hat{H}} = \frac{e^{-it\hat{H}}e^{-\hat{H}_\rho}e^{it\hat{H}}}{Z}. \tag{104}$$

As already observed in subsection 0.3.9, the commutator of two quadratic monomial of Dirac operators is a polynomial at most quadratic in Dirac operators. Using this observation together with the Baker-Campbell-Hausdorff formula (equation B.C.H.0 in appendix **??**), it is easy to see that $\rho(t)$ has the form

$$\rho(t) = \frac{e^{-\hat{H}_{\rho(t)}}}{Z}, \tag{105}$$

with $\hat{H}_{\rho(t)}$ a Fermionic quadratic Hamiltonian. Thus $\rho(t)$ is again a Gaussian state proving that the space of Gaussian states is closed under evolution induced by Fermionic quadratic Hamiltonians.
We will now compute an explicit formula for the time evolution of the correlation matrix $\Gamma(t)$ of the f.g.s. state $\rho(t)$. The first step is computing the time evolution of the creation and annihilation operators in the Heisenberg picture. We denote with $\vec{\beta}$ the vector of Dirac operators that diagonalises H. The annihilation and creation operators b_k and $b_k^\dagger$ evolved at time t are (see appendix **??**)

$$b_k(t) = e^{-i\hat{H}t}b_k e^{i\hat{H}t} = e^{-i2\epsilon_k t}b_k, \tag{106}$$

$$b_k^\dagger(t) = e^{-i\hat{H}t}b_k^\dagger e^{i\hat{H}t} = e^{i2\epsilon_k t}b_k^\dagger. \tag{107}$$

In compact form this can be written as

$$\vec{\beta}(t) = e^{i2H_D t}\vec{\beta}. \tag{108}$$

It is easy now to compute the time evolution of the correlators $\langle\vec{\beta}\vec{\beta}^\dagger\rangle$

$$\langle\vec{\beta}(t)\vec{\beta}^\dagger(t)\rangle = \langle e^{i2H_D t}\vec{\beta}\vec{\beta}^\dagger e^{-i2H_D t}\rangle. \tag{109}$$

Thus, if U is the Fermionic transformation such that $\vec{\beta} = U\vec{\alpha}$, the Fermionic transformation implementing the time evolution of $\vec{\alpha}$ is $U^\dagger e^{i2H_D t}U = e^{i2Ht}$. We finally obtain that the correlation matrix Γ evolves with H as

$$\Gamma(t) = e^{i2Ht}\Gamma e^{-i2Ht}. \tag{110}$$

F_utilities 0.3.15: Evolve(Γ**, H_D, U, t)**$\rightarrow \Gamma(t)$

This function returns the correlation matrix Γ evolved at time t with H. Matrices H_D and U are the output of `Diag_h(H)`.

0.4 Hopping model

We consider the translational invariant hopping Hamiltonian for a system of N sites

$$\hat{H} = \sum_{i=1}^{N-1} [a_i^\dagger a_{i+1} - a_i a_{i+1}^\dagger] + \delta[a_N^\dagger a_1 - a_N a_1^\dagger], \tag{111}$$

with $\delta = 1$ for periodic boundary conditions and $\delta = 0$ for open boundary conditions.
The compact form (16) of $\hat{H}$ is specified by the two circulant matrices (see **??**)

$$A = \begin{pmatrix} 0 & \frac{1}{2} & 0 & \dots & 0 & \delta\frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \dots & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \delta\frac{1}{2} & 0 & 0 & \dots & \frac{1}{2} & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix} \tag{112}$$

As seen in subsection 0.3.8 and appendix **??**) we know that H is diagonalised with a Fourier transformation. Indeed, if we express the hopping Hamiltonian (111) in term of the Fourier modes (0.5.1) we obtain

$$\hat{H} = \sum_{k=1}^{N} \phi_k (f_k^\dagger f_k - f_k f_k^\dagger), \tag{113}$$

where

$$\phi_k = \cos(\frac{2\pi}{N} k). \tag{114}$$

The Fermionic transformation that diagonalises the Hamiltonian is U_ω as defined in (83).

0.4.1 Numerical diagonalisation

In the following code we show how to initialise and diagonalise the hopping Hamiltonian using functions of `F-utilities`. We perform these calculations using two different methods. The energies computed with both methods are reported in figure **??**, the ground states correlations matrices are identical.

```
using F_utilities;
using PyPlot;
using LinearAlgebra;

const Fu = F_utilities;



N =127;

H   = Fu.Build_hopping_hamiltonian(N,PBC=true);
```

```
U_omega = Fu.Build_Fourier_matrix(N);
D_omega = U_omega'*H*U_omega;
D,U = Fu.Diag_h(H);


figure("Energies")
plot(diag(real.(D_omega))[(N+1):(2*N)],label="Method Fourier"
    ↪ );
plot(real.(diag(D))[(N+1):(2*N)], label="Method Diag_h");
xlabel(L"$k$");
ylabel(L"$\epsilon_k$");
legend();

Gamma_omega = Fu.GS_gamma(D_omega,U_omega);
Gamma   = Fu.GS_gamma(D,U);
println("")
println("Energy GS Method Fourier:      ",Fu.Energy(
    ↪ Gamma_omega,(D_omega,U_omega)))
println("En GS Method Diag_h:           ",Fu.Energy(Gamma,(D,
    ↪ U)))
```

Output:

```
Energy GS Method Fourier:  -80.85277253991693
En GS Method Diag_h:  -80.85277253997737
```

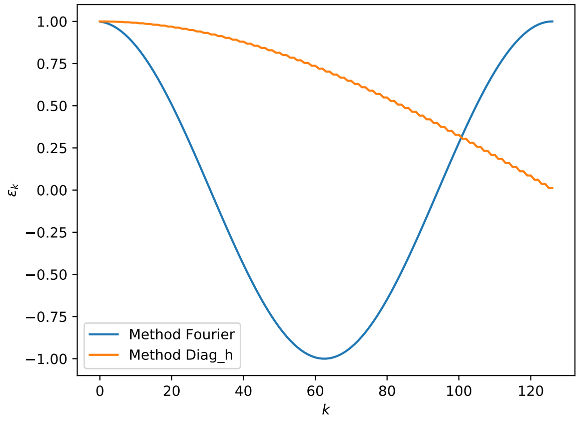


Figure 5: In this example we diagonalised the Hamiltonian with two different methods. Using the Fourier transform method we obtain the energies specified in (114). These energies are both positive and negative. Using the `Diag_h` method of `F_utilities` we obtain just positive energies. The difference in the diagonal energies comes from the fact that `Diag_h`, for every eigenmode with negative energy, substitutes creation and annihilation operators in order to redefine the energy as positive, and then reorders the modes in order to have the energies in descending order. If we diagonalise with the Fourier transform then the ground state is obtained filling all the modes with negative energy. If we diagonalise with `Diag_h` then the ground state corresponds to the empty state.

`F_utilities` 0.4.1: `Build_hopping_Hamiltonian(`N`, PBC=true)`$\to H$

This functions return the Hamiltonian H if dimension $2N \times 2N$ for the hopping model. If `PBC=false` it return the hopping Hamiltonian with open boundary conditions

For the numerical diagonalisation of the Hamiltonian we used two methods, the analytical one using the Fourier modes, and the numerical one introduced in the previous subsection.
These methods return the hopping Hamiltonian in the diagonal forms

$$\hat{H}_\omega = \sum_{k=1}^{N} \phi_k (f_k^\dagger f_k - f_k f_k^\dagger), \qquad \hat{H} = \sum_{k=1}^{N} \epsilon_k (b_k^\dagger b_k - b_k b_k^\dagger), \tag{115}$$

where the differences in the energies are due to the fact that `Diag_h` considers all the energies ϵ_k positive, thus defines $b_k = f_k^\dagger$ and $b_k^\dagger = f_k$ for each k such that $\phi_k < 0$ (that corresponds to flipping the sign of ϕ_k when it is negative, such that the corresponding $\epsilon_k = -\phi_k$), and then reorder the modes such that to modes with smaller k correspond biggest energies.

0.4.2 Time Evolution

For the hopping model we analytically obtained the Fermionic transformation U_ω that diagonalises the Hamiltonian. This allows us to give an analytical expression for the time evolution of the correlation matrix.
Expressing the correlation matrix Γ in term of the operators $\vec{\phi}$ and computing

the time evolution with the diagonal Hamiltonian (113) we obtain

$$\langle a_l^\dagger a_m\rangle(t) = \frac{1}{N}\sum_{k,k'}\sum_{x,y} e^{i2(\phi_k-\phi_{k'})t} e^{i\frac{2\pi}{N}(k(l-x)-k'(m-y))}\langle a_x^\dagger a_y\rangle \tag{116}$$

$$\langle a_l a_m\rangle(t) = \frac{1}{N}\sum_{k,k'}\sum_{x,y} e^{-i2(\phi_k+\phi_{k'})t} e^{-i\frac{2\pi}{N}(k(l-x)+k'(m-y))}\langle a_x a_y\rangle. \tag{117}$$

Because of the block diagonal structure of U_ω there is not mixing of the two types of correlators during the evolution of the correlation matrix.

Time evolution of translational invariant states Let us consider a translational invariant state Γ. In subsection 0.3.8 we expressed Γ in term of the Fourier modes $f_k^\dagger, f_k$. Using the diagonal form (115) of the Hopping Hamiltonian to compute the time evolution of the correlators of Γ expressed as in (80) we have that Γ evolves as

$$\begin{aligned}
\langle a^\dagger a\rangle_\Delta(t) &= \frac{1}{N}\sum_{k=1}^{N} e^{-i\frac{2\pi}{N}k\Delta}\langle f_k^\dagger f_k\rangle, & \langle a a^\dagger\rangle_\Delta(t) &= \frac{1}{N}\sum_{k=1}^{N} e^{-i\frac{2\pi}{N}k\Delta}\langle f_k f_k^\dagger\rangle\\
\langle a a\rangle_\Delta(t) &= \frac{1}{N}\sum_{k=1}^{N} e^{i4\phi(k)t}e^{-i\frac{2\pi}{N}k\Delta}\langle f_k f_{N-k}\rangle, & \langle a^\dagger a^\dagger\rangle_\Delta(t) &= \frac{1}{N}\sum_{k=1}^{N} e^{-i4\phi(k)t}e^{-i\frac{2\pi}{N}k\Delta}\langle f_{N-k}^\dagger f_k^\dagger\rangle.
\end{aligned} \tag{118}$$

in the following code we numerically compute the time evolution induced by a hopping Hamiltonian on a random translational invariant gaussian state with exponentially decaying correlation functions. We consider a linear system of $N = 50$ sites and evolve it for $N_{\text{steps}} = 100$ steps of $\delta_{steps} = 0.1$.

```
using F_utilities;
using PyPlot;
using LinearAlgebra;

const Fu = F_utilities;



N=50;
N_steps = 100;
delta_steps = 0.1;

#Build the circulant vector for the adaa part of the Gamma
    ↪ with exponential decaying correlations
adaa = zeros(Complex{Float64},N);#
for i=1:div(N,2)
    adaa[i] = exp(-i*0.15)*(rand()+im*rand())
end
adaa[((div(N,2))+1):N]= reverse(adaa[1:div(N,2)]);
```

```
#Build the translational invariant adaa part of the Gamma
Gamma_adaa = Fu.Circulant(adaa);
Gamma_adaa = (Gamma_adaa+Gamma_adaa’)/2.

#Build the circulant vector for the aa part of the Gamma
aa = zeros(Complex{Float64},N);
aa[2] = rand()+im*rand();
aa[3] = rand()+im*rand();
#Build the translational invariant aa part of the Gamma
Gamma_aa = Fu.Circulant(aa)
Gamma_aa = (Gamma_aa-transpose(Gamma_aa))/2.;

#Build the translational invariant Gamma
Gamma= zeros(Complex{Float64}, 2N,2N);
Gamma[(1:N),(1:N)]          = Gamma_adaa;
Gamma[(1:N).+N,(1:N).+N]    = (I-Gamma_adaa)’;
Gamma[(1:N).+N,(1:N)]       = Gamma_aa;
Gamma[(1:N),(1:N).+N]       = -conj(Gamma_aa);
Fu.Print_complex_matrix("Gamma",Gamma)

H   = Fu.Build_hopping_hamiltonian(N,true);
D,U = Diag_h(H);


Gamma_evolved   = copy(Gamma);
adaa        = zeros(Complex{Float64}, N_steps)
aa          = similar(adaa);

#Start the time evolution cycle
#at each cycle it saves the value of two correalotors
adaa[1]     = Gamma_evolved[1,2];
aa[1]       = Gamma_evolved[N+1,2];
for t=2:N_steps
    global Gamma_evolved = Fu.Evolve(Gamma_evolved,(D,U),
        ↪ delta_steps);
    adaa[t]     = Gamma_evolved[1,2];
    aa[t]       = Gamma_evolved[N+1,2];
end

figure("Evolutions")
plot(real.(adaa), color="black", label=L"$\mathfrak{R}(\
    ↪ langle a_1^{\dagger}a_2 \rangle)(t)$");
plot(imag.(adaa), color="black",linestyle="--", label=L"$\
    ↪ mathfrak{I}(\langle a_1^{\dagger} a_2 \rangle)(t)$");
plot(real.(aa), color="purple", label=L"$\mathfrak{R}(\langle
    ↪  a_1a_2 \rangle|(t))$");
```

```
plot(imag.(aa), color="purple", linestyle="--", label=L"$\mathfrak{I}(\langle a_1 a_2 \rangle)(t)$");
legend()
xlabel("timestep")
```

Output:

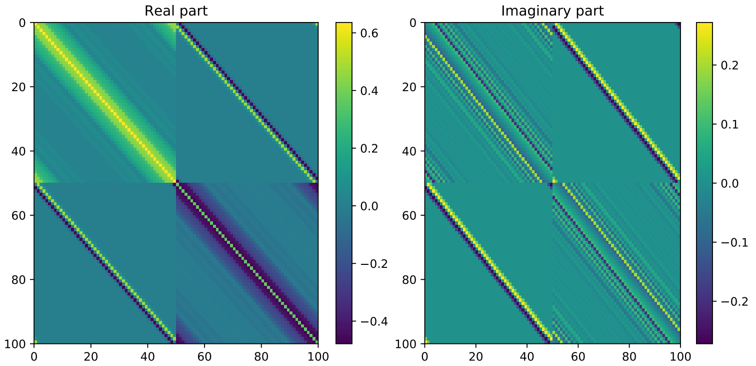


Figure 6: This is a representation of the real and imaginary part (left and right plots) of the elements of the correlation matrix Γ of a translational invariant state with exponentially decaying correlations. The element (i, j) corresponds to $\Gamma_{i,j}$. The exponential decay of the correlation is evident from the fading of the colours moving to matrix elements farer from the diagonal.

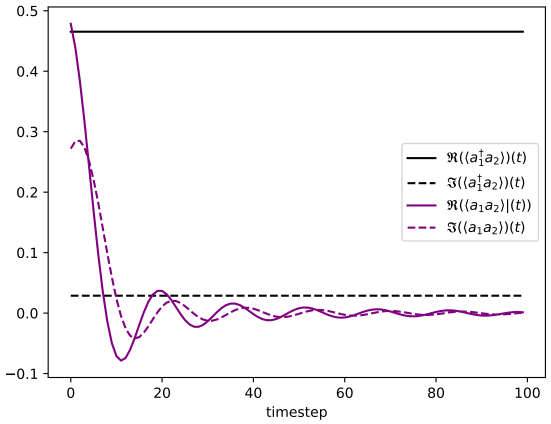


Figure 7: The time evolution induced by the Hopping Hamiltonian of the real and imaginary part of $\langle a_1^\dagger a_2 \rangle$ and $\langle a_1 a_2 \rangle$ of the translational invariant state specified in the code. The expectation values evolve as predicted by equations (118).

0.5 Transverse Field Ising Model

The Hamiltonian of the Transverse Field Ising model (TFI) has the form

$$\hat{H} = -\sum_{n=1}^{N-1} \sigma_n^x \sigma_{n+1}^x - g_I \sigma_N^x \sigma_1^x - \cot(\theta) \sum_{n=1}^{N} \sigma_n^z, \tag{119}$$

where N is the number of sites, σ_i^α with $\alpha = x, y, z$ are the Pauli matrices at the i-th site and $\cot(\theta)$ is the magnetic field, with $0 < \theta < \frac{\pi}{2}$.
The parameter g_I encodes the boundary conditions of the Ising model: here we consider $g_I = -1, 0, +1$, corresponding, respectively, to anti-periodic, open and periodic boundary conditions.
The model is called *transverse* field Ising model because the field interacts with the spins with σ_i^z, while the spins interact between each others with $\sigma_i^x \sigma_{i+1}^x$.
The TFI Hamiltonian can be exactly diagonalised using a Jordan-Wigner transformation (see appendix **??**) mapping spin operators to spinless fermions [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**].
In term of fermions the Hamiltonian has the form

$$\hat{H} = -\sum_{n=1}^{N-1} (a_n^\dagger - a_n)(a_{n+1} + a_{n+1}^\dagger) + g_I P (a_N^\dagger - a_N)(a_1 + a_1^\dagger) - \cot(\theta) \sum_{n=1}^{N} (a_n^\dagger a_n - a_n a_n^\dagger), \tag{120}$$

where $P = \prod_{n=1}^{N} (1 - 2a_n^\dagger a_n)$ is the parity operator introduced in 0.1.2.
We notice that becuase of the presence of the parity operator P, the TFI Hamiltonian cannot be directly mapped to a f.q.h.. Nonetheless, since the Hamiltonian $\hat{H}$ commutes with the parity operator P we can diagonalise $\hat{H}$ and $\hat{P}$ simultaneously. On the diagonal basis of P we have that $\hat{H}$ has the block diagonal form $\hat{H} = \hat{H}_e \oplus \hat{H}_o$, where $\hat{H}_e$, called even sector Hamiltonian, corresponds to the eigenvalue $+1$ of P and $\hat{H}_o$, the odd sector Hamiltonian, corresponds to the eigenvalue -1 of P. We can then proceed to diagonalise the Hamiltonian on the two sectors independently (see section **??** and figure **??**).
The Hamiltonians of the two sectors are:

$$\hat{H}_{\substack{e \\ o}} = -\sum_{n=1}^{N-1} (a_n^\dagger - a_n)(a_{n+1} + a_{n+1}^\dagger) \pm g_I (a_N^\dagger - a_N)(a_1 + a_1^\dagger) - \cot(\theta) \sum_{n=1}^{N} (a_n^\dagger a_n - a_n a_n^\dagger). \tag{121}$$

Finally, we see that on each parity sector the TFI Hamiltonian is mapped to a f.q.h..
A bit of confusion can raise from considering the boundary conditions. The boundary conditions of the TFI Hamiltonian do not correspond to the boundary conditions of the Fermionic Hamiltonian. In fact, let us consider the three f.q.h.

$$\hat{H}(g_F) = -\sum_{n=1}^{N-1} (a_n^\dagger - a_n)(a_{n+1} + a_{n+1}^\dagger) - g_F (a_N^\dagger - a_N)(a_1 + a_1^\dagger) - \cot(\theta) \sum_{n=1}^{N} (a_n^\dagger a_n - a_n a_n^\dagger), \tag{122}$$

where the boundary conditions are encoded by the parameter $g_F = -1, 0. + 1$ and corresponds respectively to antiperiodic, open and periodic boundary conditions of the Fermionic Hamiltonian. The correspondences between spin model and Fermionic model are collected in table 1. Written in compact form (17), Hamiltonian (122), is specified by the matrices

$$A = -\frac{1}{2}\begin{pmatrix} 2\cot(\theta) & 1 & 0 & \dots & 0 & g_F \\ 1 & 2\cot(\theta) & 1 & 0 & \dots & 0 \\ 0 & 1 & 2\cot(\theta) & 1 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ g_F & 0 & 0 & \dots & 1 & 2\cot(\theta) \end{pmatrix} \tag{123}$$

and

$$B = -\frac{1}{2}\begin{pmatrix} 0 & -1 & 0 & \dots & 0 & g_F \\ 1 & 0 & -1 & 0 & \dots & 0 \\ 0 & 1 & 0 & -1 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ -g_F & 0 & 0 & \dots & 1 & 0 \end{pmatrix}. \tag{124}$$

We already know how to numerically diagonalise this Hamiltonian. We will present here the standard method for analytical diagonalisation, introducing the Bogoliubov transformations, and we will compare the results with the numerical diagonalisation.

$g_I = 1$ (TFI periodic)	Even sector	$g_F = -1$, (f.q.h. antiperiodic)
	Odd Sector	$g_F = +1$, (f.q.h periodic)
$g_I = 0$ (TFI open)	Even/Odd sector	$g_F = 0$, (f.q.h. open)
$g_I = -1$ (TFI antiperiodic)	Even sector	$g_F = +1$, (f.q.h. periodic)
	Odd Sector	$g_F = -1$, (f.q.h antiperiodic)

Table 1: Corrspondences between spin models and Fermionic models

0.5.1 Analytical diagonalisation of the TFI Hamiltonian

In this subsection we will see how to diagonalise the three Hamiltonians (122) analytically. For a complete and detailed treatment we refer to [**?**, **?**] or the more recent review [**?**].

Antiperiodic and periodic boundary condition Fermionic Hamiltonian
Let us first consider the case of antiperiodic and periodic boundary conditions, $g_F = -1, +1$ (APBC and PBC respectively). Both cases can be brought to the

form:

$$\hat{H} = -2\sum_{k>0}\left[(\cot(\theta)+\cos(\frac{2\pi}{N}k))(f_k^\dagger f_k - f_k f_k^\dagger) + i\sin(\frac{2\pi}{N}k)(f_k^\dagger f_{-k}^\dagger - f_{-k}f_k)\right] + \\ -(1+\cot(\theta))(f_0^\dagger f_0 - f_0 f_0^\dagger) - (\cot(\theta)-1)(f^\dagger_{-\frac{N}{2}} f_{-\frac{N}{2}} - f_{-\frac{N}{2}} f^\dagger_{-\frac{N}{2}}), \tag{125}$$

where

$$f_k = \frac{1}{\sqrt{N}}\sum_{j=1}^{N} e^{i\frac{2\pi}{N}kj} a_j, \qquad f_k^\dagger = \frac{1}{\sqrt{N}}\sum_{j=1}^{N} e^{-i\frac{2\pi}{N}kj} a_j^\dagger, \tag{126}$$

with inverse transformations

$$a_j = \frac{1}{\sqrt{N}}\sum_{k} e^{-i\frac{2\pi}{N}kj} f_k, \qquad a_j^\dagger = \frac{1}{\sqrt{N}}\sum_{k} e^{i\frac{2\pi}{N}kj} f_k^\dagger. \tag{127}$$

and

$$k = \frac{-N+1}{2}, \frac{-N+1}{2}+1, \dots, \frac{N-1}{2} \qquad \text{for } g_F = -1 \text{ and } N \text{ even or } g_F = 1 \text{ and } N \text{ odd} \\ k = \frac{-N}{2}, \frac{-N}{2}+1, \dots, \frac{N}{2}-1 \quad \text{for } g_F = +1 \text{ and } N \text{ even or } g_F = -1 \text{ and } N \text{ odd}. \tag{128}$$

Terms with $f_0^\dagger, f_0$ and $f_{-\frac{N}{2}}, f^\dagger_{-\frac{N}{2}}$ in (125) are present just when $k = 0$ and $k = -\frac{N}{2}$ are allowed.

The different quantisations of the k in the two cases are justified in [**?**] and can be understood by intuition noting that with the first quantisation one would have $a_{N+1} = -a_1$, while with the second $a_{N+1} = a_1$. We can write the Hamiltonian in the compact form

$$\hat{H} = \sum_{k>0} \begin{pmatrix} f_k^\dagger & f_{-k} \end{pmatrix} h_k \begin{pmatrix} f_k \\ f_{-k}^\dagger \end{pmatrix} - (1+\cot(\theta)(f_0^\dagger f_0 - f_0 f_0^\dagger), \tag{129}$$

with

$$h_k = \begin{pmatrix} -2(\cot(\theta)+\cos(\frac{2\pi}{N}k)) & 2i\sin(\frac{2\pi}{N}k) \\ -2i\sin(\frac{2\pi}{N}k) & 2(\cot(\theta)+\cos(\frac{2\pi}{N}k)) \end{pmatrix}. \tag{130}$$

This divides the modes space in sectors that couple each k with $-k$. For each of these sectors we have the unitary transformation

$$U_k = \begin{pmatrix} s_k & -it_k \\ -it_k & s_k \end{pmatrix}, \tag{131}$$

such that it diagonalises h_k as

$$U_k^\dagger h_k U_k = \begin{pmatrix} \epsilon_k & 0 \\ 0 & -\epsilon_k \end{pmatrix}, \tag{132}$$

with eigenvalues

$$\epsilon_k = 2\sqrt{1+\cot(\theta)^2 + 2\cot(\theta)\cos(\frac{2\pi}{N}k)}. \tag{133}$$

The elements of U_k are defined as

$$s_k = \frac{\sin(\frac{2\pi}{N}k)}{\sqrt{\epsilon_k(\epsilon_k/2 + \cot(\theta) + \cos(\frac{2\pi}{N}k))}}, \tag{134}$$

$$t_k = \frac{\epsilon_k/2 + \cot(\theta) + cos(\frac{2\pi}{N}k)}{\sqrt{\epsilon_k(\epsilon_k/2 + \cot(\theta) + cos(\frac{2\pi}{N}k))}}. \tag{135}$$

This defines the Fermionic transformation of all the Fourier modes that read as

$$f_k = s_k b_k - i t_k b^\dagger_{-k}, \tag{136}$$

$$f^\dagger_{-k} = s_k b^\dagger_{-k} - i t_k b_k. \tag{137}$$

This transformation is called Bogoliubov-Valatin transformation [**?**, **?**], and sometimes s_k and t_k are expressed respectively as $\cos(\phi_k)$ and $\sin(\phi_k)$, with ϕ_k called Bogoliubov angle. One has that for PBC and APBC each f.q.h. of the form (125) is characterised by the choice of the quantisation of k and a particular Bogoliubov angle.
We finally obtain the diagonal form of the Hamiltonian

$$\hat{H} = \sum_{k\neq -\frac{N}{2},0} \frac{\epsilon_k}{2}(b^\dagger_k b_k - b_k b^\dagger_k) - (1-\cot(\theta))(f^\dagger_0 f_0 - f_0 f^\dagger_0) - (\cot(\theta)-1)(f^\dagger_{-\frac{N}{2}} f_{-\frac{N}{2}} - f_{-\frac{N}{2}} f^\dagger_{-\frac{N}{2}}). \tag{138}$$

Open boundary condition Fermionic Hamiltonian For the open boundary conditions (OBC) form of Hamiltonian (122) there is not a clear meaning for the term a_{N+1}, thus we will not apply any Fourier transform. We will not show here the procedure for the diagonalisation, we refer to [**?**, **?**] or the more recent [**?**] for it. We have that the energies ϵ_k of the Hamiltonian in diagonal form will be

$$\epsilon_k = \sqrt{1+\cot(\theta)^2 + 2\cot(\theta)\cos(\phi_k)} \tag{139}$$

with $\{\phi_k\}_{k=1}^N$ the roots of equation

$$\frac{\sin((N+1)\phi)}{\sin(N\phi)} = -\frac{1}{\cot(\theta)}, \tag{140}$$

in the interval $0 \leq \phi_k \leq \pi$.

0.5.2 Ground state

In the case of OBC the ground state is easily found using the function `GS_gamma()` with the Hamiltonian (122) imposing $g_F = 0$.

For computing the ground state in the case of APBC or PBC we need to know if the ground state is even or odd or if it is a superposition of states with different parities. It is known that, at finite dimension, with N even, for the antiperiodic Ising model the ground state is in the odd sector, while for the periodic Ising model the ground state is in the even sector. When N is odd, for the antiperiodic Ising model the ground state is in the even sector and for the periodic Ising model the ground state is in the odd sector [**?**].
In the thermodynamic limits, the energy difference between the two sectors goes to zero, the ground state becomes degenerate. Here we present a code for finding the correct sector of the ground state and for verifying that as N grows the energy difference between the ground state of the two sectors goes to zero.

```
using F_utilities;
using PyPlot;
using LinearAlgebra;

const Fu = F_utilities;

N   = 10;
theta   = pi/8;

H_APBC          = Fu.TFI_Hamiltonian(N, theta; PBC=-1);
HD_APBC, U_APBC = Fu.Diag_h(H_APBC);
NE_APBC         = diag(HD_APBC);

H_PBC           = Fu.TFI_Hamiltonian(N, theta; PBC=+1);
HD_PBC, U_PBC   = Fu.Diag_h(H_PBC);
NE_PBC          = diag(HD_PBC);

H_OBC           = Fu.TFI_Hamiltonian(N, theta; PBC=0);
HD_OBC, U_OBC   = Fu.Diag_h(H_OBC,2);
NE_OBC          = diag(HD_OBC);

AE_APBC = zeros(Float64, 2*N);
AE_PBC  = similar(AE_APBC);
AE_OBC  = similar(AE_APBC);

#The solutions of equation sin((N+1)*phi)/sin(N*phi)=-1/cot(
    ↪ theta)
phi = [0.293377974249272,0.586547314382234,
        0.879273168816649,1.17126278605144,
        1.46212217642804,1.75129510871389,
        2.03798675412035,2.32111594487769,
        2.59947341172037,2.87247738375037]
for n=1:N
```

```
    AE_APBC[n]  = sqrt(1+cot(theta)^2+2*cot(theta)*cos(2*pi
        ↪ *((1-N)/2+n-1)/N));
    AE_PBC[n]   = sqrt(1+cot(theta)^2+2*cot(theta)*cos(2*pi
        ↪ *((-N)/2+n-1)/N));
    AE_OBC[n]   = sqrt(1+cot(theta)^2+2*cot(theta)*cos(phi[n
        ↪ ]));
end

AE_APBC[(N+1):(2*N)]    = -AE_APBC[1:N];
AE_PBC[(N+1):(2*N)]     = -AE_PBC[1:N];
AE_OBC[(N+1):(2*N)]     = -AE_OBC[1:N];


fig = plt.figure("Comparison Analitical and Numerical Results
    ↪ ",figsize=(10, 6), dpi=80)
plt.subplots_adjust(wspace=0, hspace=0)
ax1 = plt.subplot2grid((21,10), (0,0), colspan=10, rowspan=7)
    ↪ ;
ax1.set_title("Comparison Analitical and Numerical")
ax1.plot(sort(AE_APBC)[11:20], color="black",  marker="o",
        markersize=10, mfc="none" ,   label="Analytical APBC"
            ↪ );
ax1.plot(sort(NE_APBC)[11:20], color="red", marker="+",
        markersize=10,  linestyle="None", label="Numerical
            ↪ APBC");
ax1.xaxis.set_ticklabels([])
yticks([1.5,2,2.5,3],[L"$1.5$",L"$2$",L"$2.5$",L"$3$"],
    ↪ fontsize=15)
ax1.set_ylabel(L"$\epsilon_k$",fontsize=18)
legend();
ax2 = plt.subplot2grid((21,10), (7,0), colspan=10, rowspan=7)
    ↪ ;
ax2.plot(sort(NE_PBC)[11:20],  color="purple",  marker="o",
        markersize=10, mfc="none" , label="Analytical PBC" );
ax2.plot(sort(AE_PBC)[11:20], color="green", marker="+",
        markersize=10,  linestyle="None", label="Numerical
            ↪ PBC");
ax2.xaxis.set_ticklabels([])
yticks([1.5,2,2.5,3],[L"$1.5$",L"$2$",L"$2.5$",L"$3$"],
    ↪ fontsize=15)
ax2.set_ylabel(L"$\epsilon_k$",fontsize=18)
legend();
ax3 = plt.subplot2grid((21,10), (14,0), colspan=10, rowspan
    ↪ =7);
ax3.plot(sort(NE_OBC)[11:20],   color="blue",  marker="o",
        markersize=10, mfc="none" , label="Analytical OBC" );
```

```
ax3.plot(sort(AE_OBC)[11:20],  color="orange", marker="+",
        markersize=10,  linestyle="None", label="Numerical
            ↪ OBC");
ax3.set_ylabel(L"$\epsilon_k$",fontsize=18)
ax3.set_xlabel(L"$k$",fontsize=18)
xticks([0,1,2,3,4,5,6,7,8,9],[L"$1$",L"$2$",L"$3$",L"$4$",L"
    ↪ $5$",L"$6$",L"$7$",L"$8$",L"$9$",L"$10$"],fontsize=15)
yticks([1.5,2,2.5,3],[L"$1.5$",L"$2$",L"$2.5$",L"$3$"],
    ↪ fontsize=15)
legend();
tight_layout();

GS_APBC     = Fu.GS_gamma(HD_APBC,U_APBC);
GS_PBC      = Fu.GS_gamma(HD_PBC,U_PBC);
E_GS_APBC   = Fu.Energy(GS_APBC,(HD_APBC,U_APBC));
E_GS_PBC    = Fu.Energy(GS_PBC,(HD_PBC,U_PBC));

println("Ground State Energies");
println("G_F=-1 :         ", E_GS_APBC);
println("G_F=+1 :         ", E_GS_PBC);
```

Output:

```
Ground State Energies
G_F=-1 :  -25.18934650837823
G_F=+1 :  -25.189223629491178
```

We have $N = 10$ so the ground state is expected to be in the GF=-1 sector. The computed energies confirm this.

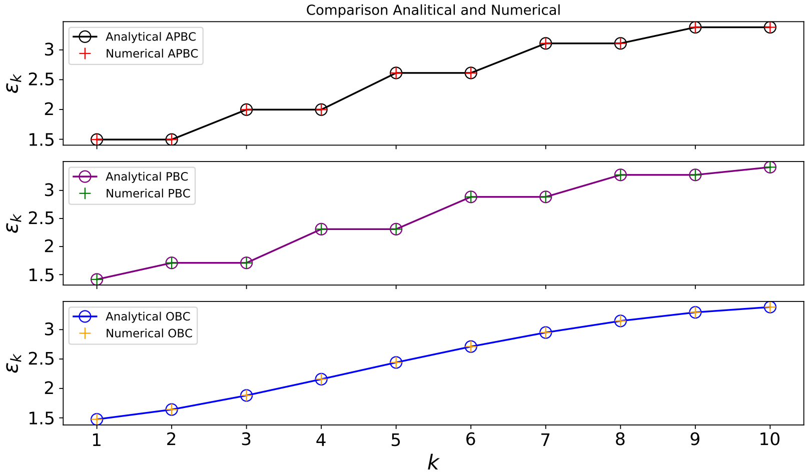


Figure 8: Output of the code 0.5.2. The three plots represent the analytical and numerical values of the free mode energies ϵ_k of the Hamiltonian (122) computed with antiperiodic, periodic and free boundary conditions. We see that the energies computed with `F_utilities` correspond to the one computed analytically.

Degenerancy of the ground state In the following code we check that the ground state energies of the TFI hamiltonians (122) with $g_F = \pm 1$ converge to the same value as the dimension of the system grows.

```
using F_utilities;
using PyPlot;

const Fu = F_utilities;

theta   = pi/8;
Delta_E = zeros(Float64, 47)

for N=4:50
        H_APBC          = Fu.TFI_Hamiltonian(N, theta; PBC
            ↪ =-1);
        HD_APBC, U_APBC = Fu.Diag_h(H_APBC);
        H_PBC           = Fu.TFI_Hamiltonian(N, theta; PBC
            ↪ =+1);
        HD_PBC, U_PBC   = Fu.Diag_h(H_PBC);

        E_GS_APBC   = Fu.Energy(Fu.GS_gamma(HD_APBC,U_APBC),(
            ↪ HD_APBC,U_APBC));
        E_GS_PBC    = Fu.Energy(Fu.GS_gamma(HD_PBC,U_PBC),(
            ↪ HD_PBC,U_PBC));

        global Delta_E[N-3]= abs(E_GS_APBC-E_GS_PBC);
end

figure("|E_GS(GF=+1)-E_GS(GF=-1)|")
plot(4:50, log10.(abs.(Delta_E)));
xlabel(L"$N$");
ylabel(L"$\log|E_{GS}(g_F=+1,N)-E_{GS}(g_F=-1,N)|$");
tight_layout();
```

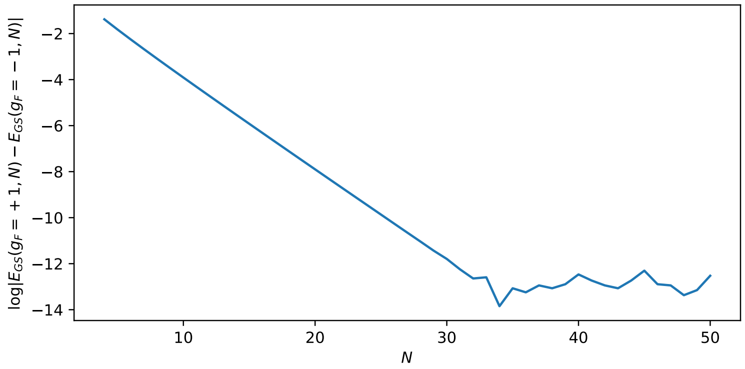


Figure 9: The ground states of Hamiltonians (122) for $g_F = \pm 1$ converge exponentially to the same value. The ground state of the antiperiodic and of the periodic transverse field Ising model is degenerate in the thermodynamic limit.

0.5.3 Time Evolution

As in the case of the Hopping model, even for the Hamiltonian (122) it is possible to explicitly compute the time evolution of the correlation matrix elements. We focus here on the case of $g_F = -1$ and N even in order to simplify the analitical form. The principal difference with the Hopping model is that, in the case of the Fermionic TFI, the transformation that diagonalises the Hamiltonian is not a simple Fourier transform, but it is a composition of a Fourier transform and a Bogoliubov transformation. We exemplify how to obtain an analitical form for the time evolution of the term $\langle a_1^\dagger a_1\rangle$ of a translational invariant correlation matrix. As a first step, exploiting the translational invariance of the state and moving to the Fourier modes, we write

$$\begin{aligned}\langle a_1^\dagger a_1\rangle = \frac{1}{N}\sum_{n=1}^{N}\langle a_n^\dagger a_n\rangle = \frac{1}{N^2}\sum_{n=1}^{N}\sum_{k=\frac{-N+1}{2}}^{\frac{N+1}{2}}\sum_{k'=\frac{-N+1}{2}}^{\frac{N+1}{2}} e^{-i\frac{2\pi}{N}n(k-k')}\langle f_k^\dagger f_{k'}\rangle = \\ = \frac{1}{N}\sum_{k=\frac{-N+1}{2}}^{\frac{N+1}{2}}\langle f_k^\dagger f_k\rangle. \qquad (141)\end{aligned}$$

We then move to the Bogoliubov modes with the transformation (136) obtaining

$$\langle a_1^\dagger a_1\rangle = \frac{1}{N}\sum_{k=\frac{-N+1}{2}}^{\frac{N+1}{2}}\left[s_k^2\langle b_k^\dagger b_k\rangle + t_k^2\langle b_{-k} b_{-k}^\dagger\rangle + i s_k t_k(\langle b_{-k} b_k\rangle - \langle b_k^\dagger b_{-k}^\dagger\rangle)\right]. \qquad (142)$$

In this basis the Hamiltonian is diagonal, thus the time evolution easily computed as

$$\langle a_1^\dagger a_1\rangle(t) = \frac{1}{N}\sum_{k=\frac{-N+1}{2}}^{\frac{N+1}{2}} \left[s_k^2\langle b_k^\dagger b_k\rangle + t_k^2\langle b_{-k} b_{-k}^\dagger\rangle + i s_k t_k (e^{i2\epsilon_k t}\langle b_{-k} b_k\rangle - e^{-i2\epsilon_k t}\langle b_k^\dagger b_{-k}^\dagger\rangle)\right] . \tag{143}$$

To obtain the expression of $\langle a_1^\dagger a_1\rangle(t)$ in terms of correlators of the operators $a^\dagger, a$ we just have to map the $b^\dagger, b$ to $a^\dagger, a$.

0.6 Benchmarking with Fermionic Gaussian states

Fermionic Gaussian states can be used as a tool for benchmarking algorithms. We will see how tools developed for general quantum states can be translated to the language of correlation matrices. To understand the idea behind the benchmarking, let us take an explicit example. In the next subsection we are going to see the imaginary time evolution of Fermionic Gaussian states. One usually uses the imaginary time evolution of a state for computing the ground state of an Hamiltonian. Of course, in the case of f.g.s., computing the ground state is not the main purpose, as we know already how to compute it for any f.q.h.. Knowing already the exact results allows us to compare the algorithm for the imaginary time evolution with the exact results and get good insight in what we should expect in a context where the exact result is not known.

If we have an algorithm acting on some generic quantum state, we can try to translate it in the formalism of f.g.s. and benchmark it to the exact results.

In this subsection we present the translation of some well known algorithm in the language of correlation matrices. The purpose of this subsection is not to benchmark these algorithm, but instead, to translate some important existing algorithms. This will provide us the translations of the possible building blocks of any novel and more complex experimental algorithm.

0.6.1 Imaginary-time evolution

In order to find the ground state of a non-degenerate Hamiltonian H one can use the following equality

$$|GS\rangle = \lim_{\tau\to\infty}\frac{e^{-H\tau}|\psi\rangle}{||e^{-H\tau}|\psi\rangle||} \tag{144}$$

starting from a generic state $|\psi\rangle$ with $\langle GS|\psi\rangle \neq 0$.

To see this, let us consider the orthononormal basis $\{|E_i\rangle\}_i$ generated by the collection of the eigenvectors of H, with eigenvalues $\{E_i\}_i$ such that $0 \leq E_0 \lneqq E_1 \leq E_2 \leq$, where $\mathcal{H}$ is the Hilbert space on which H act.

Expanding $|\psi\rangle$ on this basis one obtains $|\psi\rangle = \sum_i c_i|E_i\rangle$, with $c_0 \neq 0$ from the fact that $\langle GS|\psi\rangle \neq 0$. One can thus see that eq (144) is just a projection to the ground state:

$$\lim_{\tau\to\infty} \frac{e^{-H\tau}|\psi\rangle}{||e^{-H\tau}|\psi\rangle||} = \lim_{\tau\to\infty} \sum_i \frac{e^{-E_i\tau} c_i}{\sqrt{\sum_i e^{-2E_i\tau}|c_i|^2}} |E_i\rangle = \tag{145}$$

$$= \lim_{\tau\to\infty} \sum_i \frac{e^{-\frac{E_i}{E_0}\tau} c_i}{\sqrt{\sum_i e^{-2\frac{E_i}{E_0}\tau}|c_i|^2}} |E_i\rangle = |E_0\rangle, \tag{146}$$

and thus that $\lim_{\tau\to\infty} \frac{e^{-H\tau}}{||e^{-H\tau}||}$ is the projector on the ground state:

$$\lim_{\tau\to\infty} \frac{e^{-H\tau}}{||e^{-H\tau}||} = \lim_{t\to\infty} \frac{\sum_i e^{-E_i\tau}|E_i\rangle\langle E_i|}{\sqrt{\sum_i e^{-2E_i\tau}}} = \tag{147}$$

$$= \lim_{\tau\to\infty} \frac{\sum_i e^{-\frac{E_i}{E_0}\tau}|E_i\rangle\langle E_i|}{\sqrt{\sum_i e^{-2\frac{E_i}{E_0}\tau}}} = |E_0\rangle\langle E_0|. \tag{148}$$

The imaginary-time evolution is directly related to the power method presented in section **??**. The eigenvenvector associated to the smallest eigenvalue E_0 of H is the eigenvector associated to the biggest eigenvalue of e^{-H} and this can be approximately obtained using the power method by computing $(e^{-H})^N|\psi\rangle$, a procedure that in the limit of $N \to \infty$ is analogous to equation (144).
Applying the same method to the density matrix one can obtain the ground state ρ_{GS} of a non degenerate Hamiltonian H from a general density matrix ρ such that $Tr[\rho\rho_{GS}] \neq 0$ as

$$\rho_{GS} = \lim_{\tau\to\infty} \frac{e^{-H\tau}\rho e^{-H\tau}}{Tr\left[\rho e^{-2H\tau}\right]}. \tag{149}$$

We refer to the method for obtaining the ground state using (144) as performing an *imaginary time evolution*.
This is the case because, if for the time evolution operator $U(t) = e^{-iHt}$ for the Hamiltonian H, we select $t = -i\tau$ we obtain the operator $e^{-H\tau}$ that is the one of eq (144). One can thus write in a non-formal way $|GS\rangle = \lim_{t\to -i\infty} \frac{|\psi(t)\rangle}{||\psi(t)\rangle||}$.
It is important to keep in mind that the operator $e^{-H\tau}$ is not unitary and for this reason it does not preserve the norm of the state and one has to renormalise it.

0.6.2 Numerical Imaginary time evolution

In the numerical approach to imaginary time evolution one faces some difficulties.
Almost in all cases one is forced to evolve the state step by step renormalising every time, performing a discrete imaginary time evolution.

This procedure does not allow to reach infinite time in a finite amount of time steps, thus one has to find a criterion to stop the evolution when the convergence is accurate up to some confidence parameter. To check if the reached state is the expected state is tricky and theoretically impossible in most of the cases since one does not always have the exact value of the energy of the ground state.
A method for checking the convergence is to check the energy difference between two steps of the discrete imaginary time evolution. Once the difference in energy between two steps is lower than an acceptable value ϵ, one decides that the algorithm converged.
It is not always the case though. It is also possible that the approximate imaginary time evolution stops at some plateaux and thus it tricks the algorithm in believing in a false convergence to the ground state.

Imaginary time evolution of the correlation matrix The imaginary time evolution of the correlation matrix is defined as

$$\Gamma_{i,j}(\tau) = Tr\left[\rho(\tau)\vec{\alpha}_i\vec{\alpha}_j^\dagger\right] = \tag{150}$$

$$= \frac{Tr\left[e^{-\hat{H}\tau}\rho e^{-\hat{H}\tau}\vec{\alpha}_i\vec{\alpha}_j^\dagger\right]}{Tr\left[e^{-\hat{H}\tau}\rho e^{-\hat{H}\tau}\right]}. \tag{151}$$

Obtaining an explicit form for $\Gamma(\tau)$ just in term of H and $\Gamma(0)$ is not easy. Following the reasoning made for the real time evolution, one can compute the imaginary time evolution in Heisenberg picture with $e^{-\hat{H}\tau}$ of the operator $\vec{\alpha}_i\vec{\alpha}_j^\dagger$. Using the Baker-Campbell-Hausdorff formula (i.e. $e^A B e^A = \sum_{n=0}^{\infty}\frac{1}{n!}\underbrace{\{A,...\{A,}_{n}B\underbrace{\}...\}}_{n}$ i.e. *B.C.H.2* in Appendix **??**) and moving in the diagonal basis with Dirac operators $\vec{\beta}$ one can write the Hamiltonian as $\hat{H} = \sum_k \epsilon(k)\left(b_k^\dagger b_k - b_k b_k^\dagger\right)$.
Thus one has

$$e^{-\hat{H}\tau}\vec{\beta}_l\vec{\beta}_j^\dagger e^{-\hat{H}\tau} = \sum_{n=0}^{\infty}\frac{-\tau^n}{n!}\underbrace{\{\hat{H},...\{\hat{H},}_{n}\vec{\beta}_l\vec{\beta}_j^\dagger\underbrace{\}...\}}_{n}. \tag{152}$$

Since $b_l^\dagger b_j \hat{H} = (\hat{H} + 2\Delta_{l,j})b_l^\dagger b_j$, we cannot simplify this expression as in the case of real time evolution.
To obtain a numerical algorithm for the imaginary time evolution one has to realise that, for each value of τ, $\Gamma(\tau)$ is just the correlation matrix of the f.g.s.

$$\rho(\tau) = \frac{e^{-\hat{H}\tau}\rho e^{-\hat{H}\tau}}{\mathrm{Tr}\left[e^{-\hat{H}\tau}\rho e^{-\hat{H}\tau}\right]}, \tag{153}$$

and this can be seen as the state obtained by correctly normalising the matrix product of the density matrices of three states. The trick for obtaining the correlation matrix $\Gamma(\tau)$ is thus using the product rule (see subsection 0.3.9) of the initial f.g.s. ρ and the thermal state $\rho_{\beta=\tau} = \frac{e^{-H\tau}}{\mathrm{Tr}\left[e^{-\hat{H}\tau}\right]}$. This allows us to compute the imaginary time evolution of Fermionic Gaussian states.

`F_utilities` 0.6.1: `Evolve_imag(`Γ`, H_D, U,` τ`)`$\rightarrow \Gamma(\tau)$

This function returns the correlation matrix Γ evolved at imaginary time τ with H. Matrices H_D and U are the output of `Diag_h(H)`.

0.6.3 Fermionic Gaussian States with Fixed Bond Dimension

The compression of correlation matrices of f.g.s. in a similar fashion of matrix product states (MPS) has been introduced in [**?**].

Let us consider a pure Fermionic Gaussian state completely described by the $N \times N$ matrix $\Lambda_{i,j} = \langle a_i^\dagger a_j \rangle$. Since Λ is a pure state, its eigenvalues are either 1 (the mode is occupied) or 0 (the mode is unoccupied). This high degeneracy can be exploited mixing occupied (or unoccupied) eigenstate for finding a basis in which these modes are localised. In systems with a limited entanglement structure we expect to be able to find a basis in which eigenstates are localised. This fact can be justified as follows. Let Λ be the ground state of a 1D local Hamiltonian. We consider the partition of the first ℓ sites of the system. The state of the partition is described by the $\ell \times \ell$ correlation matrix Λ_ℓ. In ground states of 1D local Hamiltonians, the entanglement of partitions of the systems of dimension ℓ grows at most as $\log(\ell)$. This means that for growing values of ℓ we expect to find eigenvalues of Λ_ℓ closer and closer to 1 or 0. Now suppose that diagonalising Λ_ℓ, the eigenvalue associated to the eigenvector $\vec{v}$ is ~ 1. This makes $\vec{v}$ also an approximate eingenvalue of Λ.

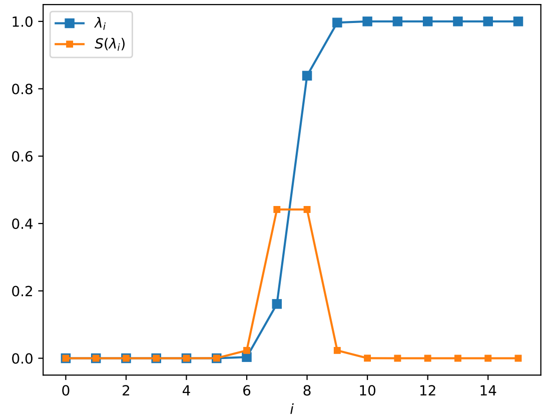


Figure 10: In blue the eigenvalues λ_i of the reduced state $\Lambda_{\ell=16}$ of the ground state of a hopping Hamiltonian with $N = 500$. In orange the von Neumann entropy $S(\lambda_i)$ of the mode associated to each eigenvalue λ_i. The total von Neumann entropy of the partition $\Lambda_{\ell=16}$ is given by the sum of the entropies of each mode (see (89)). Since the entropy of a partition Λ_ℓ is bounded by $\log(\ell)$, with growing ℓ the added modes must have associated eigenvalues close to 0 or 1.

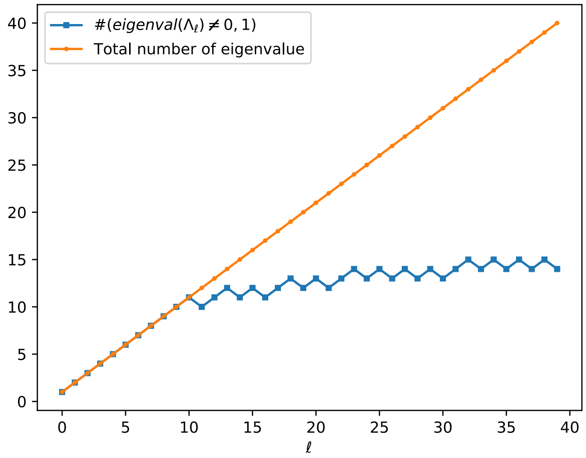
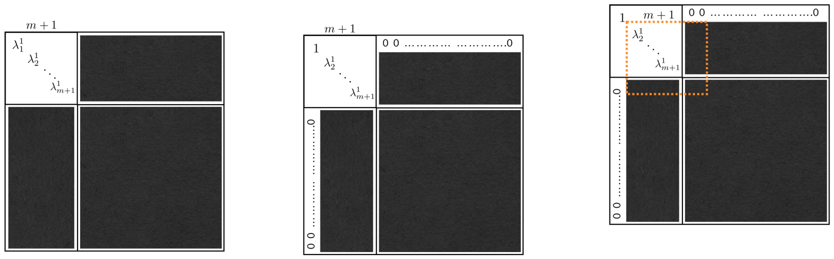


Figure 11: In blue the number of eigenvalues of Λ_ℓ that are 0 or 1 up to machine precision, in orange the total number of eigenvalues of Λ_ℓ in the ground state of a hopping Hamiltonian with $N = 500$. When the dimension of the partition is $\ell > 10$ the entanglement saturates and the number of eigenvalues equal to 1 or 0 starts growing linearly.

In [**?**], developing on this idea, the authors are able to construct a compression algorithm for correlation matrices and directly map it to the MPS representa-

tion of the state.
Here we will illustrate a method for obtaining the correlation matrix of a f.g.s. expressed as an MPS with fixed bond dimension D.

Let us consider a pure f.g.s. $|\psi\rangle$ on a system with N sites with associated $N \times N$ correlation matrix Λ. We denote with $|\psi^D\rangle$ the state obtained representing $|\psi\rangle$ with an MPS of fixed bond dimension D. We are interested in the correlation matrix Λ^D of the state $|\psi^D\rangle$.
For a bipartition having bond dimension D corresponds to having Schmidt rank D [**?**]. If a state $|\psi\rangle$ has bond dimension $D' > D$, we can approximate it at bond dimension D by setting to 0 the lower Schmidt coefficients and renormalising the state. With the formalism of correlation matrices we cannot directly manipulate the single Schmidt coefficients, but we can approximate low entangled modes with product modes. Approximating an entangled mode with a product mode corresponds to setting half of the Schmidt coefficients to zero. With this insight we can devise the following algorithm for obtaining Λ^D.
We proceed as follows. We consider $\Lambda_{1,\dots,m+1}$, the correlation matrix of the partition of the first ℓ sites, where $m = \lceil \log_2(D) \rceil$. We will refer to m as the bond dimension of the correlation matrix.
We diagonalise it as $\Lambda_{1,\dots,m+1} = U_1 D^1_{1,\dots,m+1} U_1^\dagger$, the diagonal elements of $D_{1,\dots,\ell}$ are organised such that, the top left element λ^1_1 is the closest to 0 or 1. Suppose $\lambda^1_1 \sim 1$. We expand U_1 to be $N \times N$ adding ones on the diagonal and we have that the top left element of $\Lambda^1 = U_1^\dagger \Lambda U_1$ is λ^1_1. We set the first column and first row of Λ^1 to 0 and $(\Lambda^1)_{1,1}$ to 1 (because $\lambda^1_1 \sim 1$). We then proceed diagonalising $\Lambda^1_{2,\dots,m+2} = U_2 D^2_{2,\dots,m+2} U_d^\dagger$. Suppose this time the top left element of D^2 is $\lambda^2_1 \sim 0$. We set the second column and second row of $\Lambda^2 = U_2^\dagger \Lambda^1 U_2$ to 0 and $(\Lambda^2)_{2,2}$ to 0 (because $\lambda^2_1 \sim 0$). Iterating this procedure $N - \ell$ times we obtain a correlation matrix $\Lambda^{(N-m1)}$ with $N - m$ diagonal elements equal to 0 or 1. We proceed with the same procedure decreasing the dimension of the reduced system everytime until after N steps we obtain a diagonal matrix with diagonal elements equal to 0 or 1. Returning to the original basis applying all the transformation $\{U_i\}_{i=1,\dots,N}$ to Λ^N we obtain the correlation matrix associated to the state $|\psi^D\rangle$. We report a schematic representation of the algorithm in figures 12 and 13.

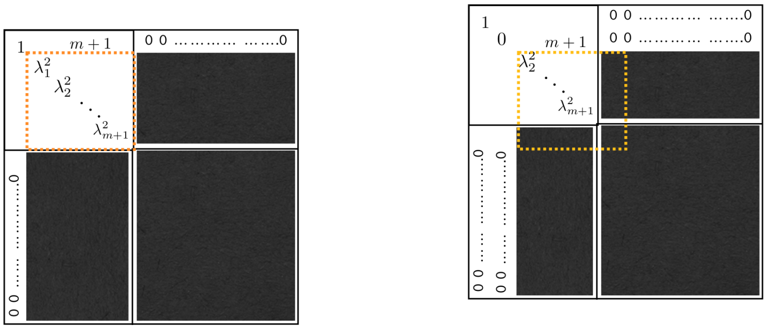


(a) Step 1: Diagonalise the subsystem $\Lambda^1_{1,\dots,m+1}$. The eigenvalues are ordered such that $\min(|1-\lambda_1^1|, |\lambda_1^1|) \leq \min(|1-\lambda_2^1|, |\lambda_2^1|) \leq \cdots \leq \min(|1-\lambda^1_{m+1}|, |\lambda^1_{m+1}|)$.

(b) Step 2: Since $\lambda_1^1 \sim 1$ ($\lambda_1^1 \sim 0$) we set it to 1 (0). We set to zero all the other elements of the first row and first column of Λ. This will be the an eigenvalue of Λ.

(c) Step 3: We move to the second subsystem $\Lambda^2_{2,\dots,m+2}$. We note that now the correalation matrix is represented in a mixed basis and the lower indices do not exactly represent the sites of the system.

Figure 12: Steps of the algorithm for reducing the bond dimension of a Fermionic Gaussian state. The big squares represent the correlation matrix Λ. We repeat this procedure $(N-m)$ times then we continue for m steps reducing by one the dimension of the reduced system at each step. At the end one obtain a diagonal matrix with diagonal elements equal to 1 or 0.



(a) Step 4: We diagonalise $\Lambda^2_{2,\dots,\ell+1}$

(b) Step6: After resetting λ_1^2, we move to $\Lambda^3_{3,\dots,m+3}$

.

Figure 13: Steps 4 and 6 of the algorithm for reducing the bond dimension of a Fermionic Gaussian state.

In figure 14 we plot the number of eigenvalues different from 0 and 1 for different partitions of the system for the ground state Γ of a hopping Hamiltonian of a system of $N = 100$ sites, and for Γ with bond dimension reduced to m.
We know that for ground states of $1D$ local Hamiltonians the amount of entanglement (measured by the von Neumann entropy S) of any region of an MPS of bound dimension D is bounded by $S \leq \log(D)$. In figure 15 we plot the value of the entropy of different regions of the ground state of a random Hamiltonian. The entropy is indeed bounded by $S \leq \log(D)$ with $D = 2^m$.
Combining this method with the imaginary time evolution algorithm one can construct the time evolving block decimation algorithms on the space of correlations matrices.
Together with the algorithm for reducing the bond dimension of one dimensional systems on the space of correlation matrices, in `F_utilities` we include the algorithm for reducing the bond dimension of specific two dimensional systems. In particular we focussed on two dimensional systems where the Hamiltonian can be sectorised. This algorithm differs from the one for one dimensional systems in the way it handles the lowest Schmidt's coefficients of different sectors. Taking advantage of the sectorisation of the Hamiltonian, the algorithm becomes more complex, but at the same time more efficient.

0.6.4 Reduction of the bond dimension of a two dimensional system divided in sectors

We consider the Fermionic quadratic Hamiltonian

$$\hat{H} = \sum_x \sum_y \left[a^\dagger_{x,y} a_{x+1,y} + a^\dagger_{x+1,y} a_{x,y} + (-1)^x \left(a^\dagger_{x,y} a_{x,y+1} + a^\dagger_{x,y+1} a_{x,y} \right) \right], \tag{154}$$

defined on the on an $L \times L$ lattice on a cylinder with periodic boundary conditions along the x direction and open boundary conditions along the y direction as in figure 0.6.4. Because of the boundary conditions we have that L must be even.
Substituting the Fourier operators

$$a^\dagger_{x,y} = \frac{1}{\sqrt{L}} \sum_k e^{-i\frac{2\pi}{L}kx} c^\dagger_{k,y}, \tag{155}$$

where, because of the boundary conditions, we choose k as

$$k = -\frac{L}{2}, -\frac{L}{2}+1, \ldots, \frac{L}{2}-1, \tag{156}$$

the Hamiltonian becomes as

$$\begin{aligned}\hat{H} = \sum_{k<0} \Big[\sum_y 2\cos(\frac{2\pi}{L}k) \left(c^\dagger_{k,y} c_{k,y} - c^\dagger_{k+\frac{L}{2},y} c_{k+\frac{L}{2},y} \right) + \\ + c^\dagger_{k,y} c_{k+\frac{L}{2},y+1} + c^\dagger_{k+\frac{L}{2},y} c_{x,y+1} + c^\dagger_{k+\frac{L}{2},y+1} c_{k,y} + c^\dagger_{k,y+1} c_{k+\frac{L}{2},y} \Big].\end{aligned} \tag{157}$$

In this form the Hamiltonian is divided in $\frac{L}{2}$ sectors, each one corresponding to the couples of values of $\{k, k+\frac{L}{2}\}$. This means that the eigenstates of this Hamiltonain are product states of states defined on each $\{k, k+\frac{L}{2}\}$ sector. Thus, these eigenstates, instead of being described by a $2L^2 \times 2L^2$ correlation matrix, are instead described by just a collection of $\frac{L}{2}$ correlation matrices of dimension $2L \times 2L$, where each of these correlation matrices corresponds to a stripe of the cylinder in the (k, y) space.
We call states of this kind, sectorised states.

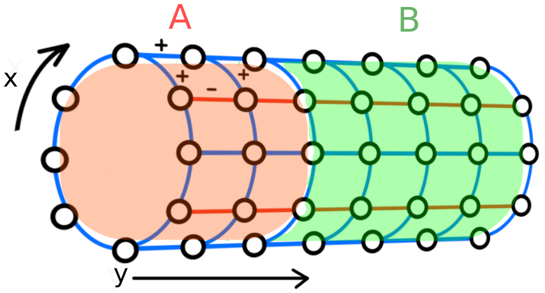


Figure 14: Lattice with periodic boundary conditions along the x direction. Red lines correspond to negative couplings, blue lines correspond to positive couplings. Two possible partitions A and B are higlighted in red and green respectively.

Performing the reduction of the bond dimension of a sectorised state, one can expolit the sectorisation property in order to reduce the computational cost of the operation.

Consider for example the ground state of Hamiltonian (154). Once we move to the Fourier basis along the x direction this becomes a sectorised state. Mimicking the encoding of this quantum state with a tensor network corresponds to fixing its Schmidt rank relatively to some iterative partition scheme. We choose a partition scheme that increasingly cuts the cylinder perpendularly to y. Since the Fourier transformation we applied mixes only Dirac operator corresponding to the same value of y, this partition scheme is a geometric partition scheme (it is equivalent on the (x, y) space and the (k, y) space). Step l of the partition scheme divides the system in a partition A consisting of all the elements corresponding to $y = \{1, \ldots, l\}$ and a partition B corresponding to all the elements corresponding to $y \in \{l+1, \ldots, L\}$ analogously of the iterative partition scheme used for the one dimensional system of the last section.

Choosing this partition scheme allows us to exploit a parallel implementation of the algorithm for the reduction of the bond dimension of the state. In fact, instead of considering the full $2L^2 \times 2L^2$ correlation matrix describing the state we consider the $\frac{L}{2}$ correlation matrices of dimension $2L \times 2L$. The first step of the algorithm consists in partitioning the system at step $m+1$ of the partition scheme. These corresponds to taking the first $2(m+1)$ elements of each correlation matrix. One then proceed diagonalising these subsystem. This step is analogous to step 1 of the one dimensional system presented in the last section, with the difference that now we are acting on $\frac{L}{2}$ correlation matrices simultaneously. This returns $L(m+1)$ eigenvalues from the $\frac{L}{2}$ sectors. Step 2 consists in considering all these $L(m+1)$ eingenvalues *together*, selecting the $2\frac{L}{2}$ closest to 0 or 1 and then approximating them with 0 or 1 respectively in the respective correlation matrix and setting them to product state with the rest of the system analogously to what was done in step 2 of the one dimensional case. Here the difference with the one dimensional case of the last section consists in the fact that we are setting to product state L modes, not just one, and that we are choosing them from all the correlation matrices. Step 3 consists in moving to the partition scheme $m+2$, enlarging the first partitions. Differently from the one dimensional case, here one has to keep track of the number of approximations performed in each sectors before diagonalising. As in the one dimensional case, the algorithm then proceed iteratively returning after at most L steps, the correlation matrix $\Gamma(m)$ of a state with reduced bond dimension with the respect to the partitions along the chosen spatial direction. Considering larger values of m, the correlation matrix $\Gamma(m)$ converges towards Γ, the exact correlation matrix.

F_utilities 0.6.2: RBD(Γ, m)$\rightarrow \Gamma(m)$

This function returns the correlation matrix $\Gamma(m)$ obtained reducing the bond dimension of Γ to m.

F_utilities 0.6.3: RBD_csectors($\vec{\Gamma}$, L_x, L_y, m)$\rightarrow \vec{\Gamma}(m)$

This function returns the correlation matrices $\vec{\Gamma}(m)$ obtained reducing the bond dimension to mL of the system on the $L_x \times L_y$ cylinder described by the correlation matrices $\vec{\Gamma}$. Each correlation matrix must cointains information of two values of k and must be organised following the order $\vec{\alpha} = (a_{k,1}, a_{k+\frac{L}{2},1}, a_{k,2}, a_{k+\frac{L}{2},2}, \dots)$

```
using F_utilities;
using PyPlot;
using LinearAlgebra;

const Fu    = F_utilities;
const LinA  = LinearAlgebra;


N           = 100;
H           = Fu.Build_hopping_hamiltonian(N);
HD, U       = Fu.Diag_h(H);
Gamma       = Fu.GS_gamma(HD,U);
Gamma_RBD   = Fu.RBD(Gamma,5)

prod_modes_border       = zeros(Int64, N);
prod_modes_RBD_border   = zeros(Int64, N);
prod_modes_bulk         = zeros(Int64, div(N,2));
prod_modes_RBD_bulk     = zeros(Int64, div(N,2));
for l=1:N
      DA,UA             = Fu.Diag_gamma(Fu.Reduce_gamma(Gamma
          ↪  ,l,1));
      DA_RBD,UA_RBD     = Fu.Diag_gamma(Fu.Reduce_gamma(
          ↪  Gamma_RBD,l,1));
      prod_modes_border[l]     = count(i->i!=0, round.(real.(
          ↪  LinA.diag(DA)[1:l]),digits=14));
      prod_modes_RBD_border[l] = count(i->i!=0, round.(real.(
          ↪  LinA.diag(DA_RBD)[1:l]),digits=14));
end
for l=1:div(N,2)
      DA,UA                   = Fu.Diag_gamma(Fu.Reduce_gamma
          ↪  (Gamma,l, div(N,2)));
```

```
        DA_RBD,UA_RBD            = Fu.Diag_gamma(Fu.Reduce_gamma
            ↪ (Gamma_RBD,l, div(N,2)));
        prod_modes_bulk[l]       = count(i->i!=0, round.(real.(
            ↪ LinA.diag(DA)[1:l]),digits=14));
        prod_modes_RBD_bulk[l]   = count(i->i!=0, round.(real.(
            ↪ LinA.diag(DA_RBD)[1:l]),digits=14));
end
figure("prod_eigenvalues");
plot(1:N,prod_modes_border, marker="s", markersize=3, label=L
    ↪ "$\#(eigenval(\Gamma_{1,\dots,\ell}\neq 0,1)$");
plot(1:div(N,2),prod_modes_bulk, marker="s", markersize=3,
    ↪ label=L"$\#(eigenval(\Gamma_{\frac{N}{2},\dots,\frac{N
    ↪ }{2}+\ell})\neq 0,1)$");
plot(1:N,prod_modes_RBD_border, marker="s", markersize=3,
    ↪ label=L"$\#(eigenval(\Gamma(m=5)_{1,\dots,\ell}\neq
    ↪ 0,1)$");
plot(1:div(N,2),prod_modes_RBD_bulk, marker="s", markersize
    ↪ =3, label=L"$\#(eigenval(\Gamma(m=5)_{\frac{N}{2},\
    ↪ dots,\frac{N}{2}+\ell}\neq 0,1)$");
axvline(div(N,2), linestyle="--", linewidth=0.5, color="gray"
    ↪ )
xlabel(L"$\ell$")
grid(axis="y", linestyle="--")
legend();
```

Output:

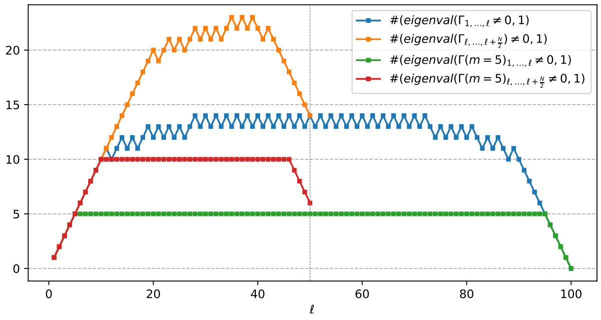


Figure 15: Eigenvalues different from 0 and 1 for different partitions of the system for the ground state Γ of a hopping Hamiltonian of a system of $N = 100$ sites, and for Γ with bond dimension reduced to m. The blue dots correspond to partitions of Γ with first site at the boundary of the system and with dimension ℓ. The orange dots correspond to partitions of Γ with first site at the boundary of the system and with dimension ℓ. The green dots are anologous to the blue dots, but computed for the state $\Gamma(m = 5)$ obtained reducing the bond dimension of Γ to $m = 5$. Red dots are anologous to the orange dots, but computed for $\Gamma(m = 5)$. As expected the number of eigenvalues different from 0 and 1 are bounded as $\#eigenval(\neq 0, 1) \leq 2m$.

```
using F_utilities;
using PyPlot;
using LinearAlgebra;

const Fu    = F_utilities;
const LinA  = LinearAlgebra;

function Random_hamiltonian(N)
  A   = rand(N,N)+im*rand(N,N);
  A   = (A+A')/2.;
  bd  = rand(N-1).+im*rand(N-1);
  B   = Tridiagonal(bd, zeros(Complex{Float64}, N), -bd);
  H   = zeros(Complex{Float64}, 2*N, 2*N);
  H[(1:N),(1:N)]          = -conj(A);
  H[(1:N).+N,(1:N)]       = -conj(B);
  H[(1:N),(1:N).+N]       = B;
  H[(1:N).+N,(1:N).+N]    = A;

  return H;
end

N           = 100;
H           = Random_hamiltonian(N);
HD, U       = Fu.Diag_h(H);
Gamma       = Fu.GS_gamma(HD,U);
Gamma_RBD   = Fu.RBD(Gamma,1)
S_modes_border       = zeros(Float64, N);
S_modes_RBD_border   = zeros(Float64, N);
S_modes_bulk         = zeros(Float64, div(N,2));
S_modes_RBD_bulk     = zeros(Float64, div(N,2));
for l=1:N
      S_modes_border[l]     = Fu.VN_entropy(Fu.Reduce_gamma(
          ↪ Gamma,l,1));
      S_modes_RBD_border[l] = Fu.VN_entropy(Fu.Reduce_gamma(
          ↪ Gamma_RBD,l,1));
```

```
end
for l=1:div(N,2)
      S_modes_bulk[l]         = Fu.VN_entropy(Fu.Reduce_gamma
          ↪ (Gamma,l, div(N,2)));
      S_modes_RBD_bulk[l]     = Fu.VN_entropy(Fu.Reduce_gamma
          ↪ (Gamma_RBD,l, div(N,2)));
end
figure("Entropies");
plot(1:N,log.(abs.(S_modes_border)), marker="s", markersize
    ↪ =3, label=L"$F=S(\Gamma_{1,\dots,\ell}\neq 0,1)$");
plot(1:div(N,2),log.(abs.(S_modes_bulk)), marker="s",
    ↪ markersize=3, label=L"$F=S(\Gamma_{\frac{N}{2},\dots,\
    ↪ ell+\frac{N}{2}})\neq 0,1)$");
plot(1:N,log.(abs.(S_modes_RBD_border)), marker="s",
    ↪ markersize=3, label=L"$F=S(\Gamma(m=1)_{1,\dots,\ell}\
    ↪ neq 0,1)$");
plot(1:div(N,2),log.(abs.(S_modes_RBD_bulk)), marker="s",
    ↪ markersize=3, label=L"$F=S(\Gamma(m=1)_{\frac{N}{2},\
    ↪ dots,\ell+\frac{N}{2}}\neq 0,1)$");
axvline(div(N,2), linestyle="--", linewidth=0.5, color="gray"
    ↪ )
axhline(log(log(2)), linestyle="-.", color="red", label="F=
    ↪ log(D)")
axhline(log(2*log(2)), linestyle="-.", color="red", label="F
    ↪ =2*log(D)")
xlabel(L"$\ell$")
ylabel(L"$\log(F)$")
# grid(axis="y", linestyle="--")
legend();
```

Output:

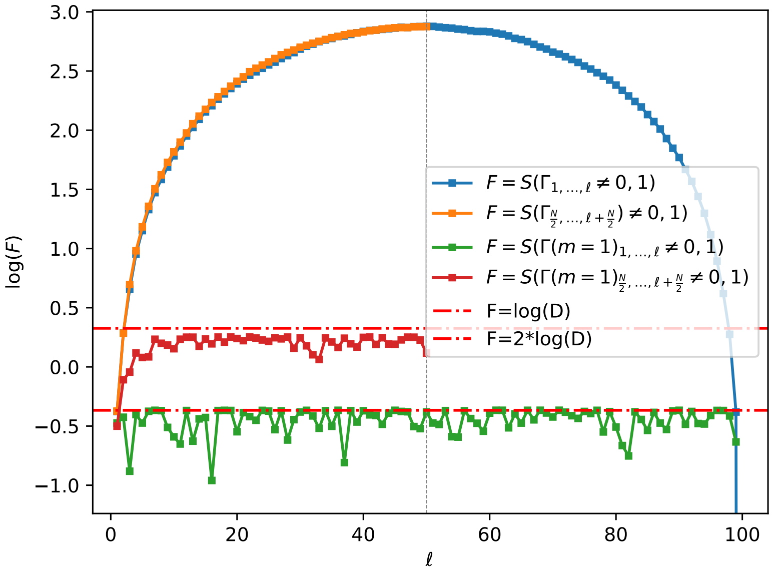


Figure 16: The entropy of different regions of the ground state of a random Hamiltonian. The blue dots correspond to partitions of dimension ℓ with first site at the boundary of the chain. The orange dots correspond to partitions of dimension ℓ with first site in the middle of the chain. The green dots are anologous to the blue dots but computed for the state $\Gamma(m = 1)$ obtained reducing the bond dimension of Γ to $m = 1$. Red dots are anologous to the orange dots, but computed for $\Gamma(m = 1)$. As expected since the Hamiltonian is random and long range, the entropy of the partitions almost always saturates. The red dash-dotted horizontal lines represent the upper bound for the entropy of a partition (starting at the border or not respectively for $\log(D)$ and $2\log(D)$). As we can see the entropy is always bounded by $S \leq \log(D)$ with $D = 2^m$ as expected.