Approximating ground states with free-fermionic states on a quantum computer

Honours Programme Research Project

Author: Ignacio Fernández Graña
Supervisor: Jordi Tura i Brugués
Second examiner: Johannes Borregaard

Abstract

Approximating the ground state of quantum-many body systems is a central topic in a wide range of fields. Quantum computing is thought to offer a substantial advantage tackling this problem relative to classical computers. We introduce two methods to approximately prepare the ground state of a free-fermion Hamiltonian on a quantum computer. First, we propose an adaptative algorithmic cooling method, where parametrized operators are sequentially applied and optimized to monotonically decrease the energy of an initial state. Next, we design a more refined implementation inspired in a diagonalization algorithm for antisymmetric matrices, known as Paardekooper algorithm, which significantly reduces the number of rotations required to prepare the ground state for small system sizes. We numerically study both methods for small system sizes n and show they can asymptotically reach the ground state in $\mathcal{O}(n^2)$ operators. Finally, we go beyond the free-fermion Hamiltonian and extend the algorithm to a model with a ground state that is manifestly non-Gaussian, namely the Sachdev–Ye–Kitaev model.

Contents

1 Introduction

The simulation of quantum many-body systems is a central topic in the field of condensed matter and quantum chemistry. Unfortunately, simulating these systems with classical computers quickly becomes intractable even for moderate system sizes, due to the exponential scaling of their Hilbert space. On the other hand, quantum computers offer an exceptional platform for quantum many-body system simulation [1, 2], providing an exponential save in memory cost. In particular, approximating the ground state of such models has been identified as one of the key potential applications quantum computing, as a wide range of interesting phenomena in physics happen at low energies. Preparing the ground state of a Hamiltonian can be a difficult task- in fact, this problem is the paradigmatic example of the QMA-complete complexity class, which is, roughly speaking, the analogue of NP-complete for quantum computers [3]. Furthermore, ground states are also closely related to optimization tasks, as the solution of many combinatorial problems can be naturally encoded in the ground state of a Hamiltonian [4].

In this work we focus on a very simple quantum many-body model, namely a non-interacting system of n fermions, also known as free-fermion systems. Free-fermion systems are known to be exactly solvable in a efficient way [5], which allows us to numerically benchmark the methods proposed in this work. We propose two methods to approximately prepare the ground state of a free-fermion system in a quantum computer. The building block of the methods is an unitary operator that acts on the fermionic space by inducing an orthogonal change of basis. More specifically, the matrix of this change of basis is a 2D rotation known as a *Givens rotation*. As a consequence, this operators maps a free-fermion Hamiltonian into another free-fermion Hamiltonian, which guarantees that the degree of the Hamiltonian is preserved.

The first method uses a variational approach, where we randomly apply the Givens rotations sequentially, optimizing the angle of the rotations so to minimize the energy at each step. This way, the energy of the state is a monotonically decreasing function of time. Moreover, according to the variational principle this function is lower bounded by the ground state energy, which guarantees convergence within the parity subspace of the initial state. The convergence is limited to the parity subspace of the initial state because the parity of a state remains invariant under the action of these Givens rotations. We refer to this procedure as *adaptative algorithmic cooling*. There exist various algorithmic cooling approaches, the most usual of which works by coupling the state to some ancillary qubits that are repeatedly measured to extrac heat from the system [6, 7]. Our strategy does not require any auxiliar qubits, and resembles that of adaptative quantum circuits [8, 9], where the structure and length of the ansatze are not fixed. In our case, the quantum circuit grows one operator at a time until the required accuracy is reached, although many variants of this idea are straightforward to implement. This allows to greatly simplify the classical optimization of the ansatz parameters, as only one parameter is optimized at a time. This is specially advantageous since the training of variational algorithms is known to be NP hard in a wide variety of cases, including that of free fermions [10].

The second method is inspired by the use of Givens rotations in computational algebra algorithms. Givens rotations are frequently used to perform matrix decompositions and to solve the eigenvalue problem of (anti-)symmetric matrices [11]. Our method is based on an algorithm, proposed by Paardekooper [12], to semi-diagonalize antisymmetric matrices via similarity transformations using Givens rotations. Paardekooper algorithm uses Givens rotations to iteratively annihilate the off-diagonal elements of the matrix, bringing the matrix to a block diagonal form. This allows us to order the Givens rotations through an algebraic mindset rather than randomly, resulting in a reduced number of rotations required to achieve the ground state for small system sizes, in addition to avoid any classical optimization.

Finally, we show that the algorithmic cooling procedure can be extended to approximate the best Gaussian state estimation of the ground state of more complex fermionic models. Gaussian states are quantum states that are the ground state of a free-fermion Hamiltonian [13]. They are a central topic in quantum many-body physics, and are known to provide a reasonable approximation for more complex models [14]. Gaussian states are frequently used as a starting point to more refined algorithms- a famous example are Slater determinants, which are a subset of Gaussian states, and are used in the

Hartree-Fock state method in quantum chemistry [15]. In our case, we extend the algorithm to a quantum many-body system that has a ground state which is manifeslty non-Gaussian [16], namely the Sachdev–Ye–Kitaev (SYK) model. The SYK model describes $2n$ Majoranas with all to all random interactions, where the magnitude of the interactions are normally distributed. The SYK model has become an important tool for the study of quantum gravity as it is the simplest known example of AdS/CFT duality [17–19]. The AdS/CFT (Anti-de Sitter/Conformal Field Theory) duality conjectures the relationship between the physics of quantum many-body systems and the physics of gravity in higher dimensions. Indeed, the SYK model is known to share common properties with the physics of the near horizon region of near extremal black holes. It is also a primary platform in the study of quantum chaos and scrambling of information [20–23], as the model is maximally chaotic yet amenable to exact analyis in the large n limit. Extending our algorithm to the SYK model allows us to quantify the non-Gaussianity of its ground state for small system sizes.

The numerical simulations of this work were implemented in Python 3.8[1]. For the symbolic calculations in second quantization we used the Mathematica package SNEG [24].

This work is structured as follows. In Section 2 we review the free-fermion model and its exact diagonalization. In Section 3 present the algorithms to approximately prepare the ground state of a free-fermion Hamiltonian. In Section 4 we extend this method to provide a Gaussian approximation of the SYK model. Finally, in Section 5 we conclude the work and discuss possible future outlooks.

2 Free-fermion model

A free-fermion model corresponds to a Hamiltonian of the form:

$$\hat{\mathcal{H}} = \sum_{0\leq i\neq j<n} A_{ij}\hat{a}_i^\dagger \hat{a}_j + B_{ij}\hat{a}_i\hat{a}_j + \text{ h.c.}, \tag{1}$$

where $A_{ij}, B_{ij} \in \mathbb{C}$ and h.c. stands for hermitian conjugate. The operators $\hat{a}_i(\hat{a}_i^\dagger)$ are the annihilation (creation) Dirac operators of a fermionic system with n modes. The annihilation operator $\hat{a}_i$ acting on the fermionic vacuum $|\Omega\rangle$ satisfies $\hat{a}_i|\Omega\rangle = 0\ \forall i$, while the creation operator $\hat{a}_i^\dagger$ populates the i-th mode $\hat{a}_i^\dagger|\Omega\rangle = |1_i\rangle$. Dirac operators satisfy the following Canonical Anticommutation Relations (CARs):

$$\left\{\hat{a}_i, \hat{a}_j^\dagger\right\} = \delta_{i,j}\hat{\mathbb{1}}, \quad \{\hat{a}_i, \hat{a}_j\} = 0 \quad \forall i,j\ , \tag{2}$$

where $\delta_{i,j}$ stands for the Dirac delta function. The Hamiltonian described in equation (1) can also be expressed in terms of Majorana fermions [5, 25]

$$\hat{\mathcal{H}} = \frac{\mathrm{i}}{2}\sum_{i,j=0}^{n-1}\sum_{\alpha,\beta=0}^{1} H_{i,\alpha;j,\beta}\hat{c}_{i,\alpha}\hat{c}_{j,\beta}, \tag{3}$$

where the $2n$ Majorana operators $\hat{c}_{i,\alpha}$ are defined as

$$\hat{c}_{i,\alpha} = \mathrm{i}^\alpha(\hat{a}_i + (-1)^\alpha\hat{a}_i^\dagger). \tag{4}$$

The coupling matrix H is a $2n\times 2n$ matrix with elements denoted by $H_{i,\alpha;j,\beta}$. The indices $i,j = 0,\dots n-1$ index each of the 2×2 blocks of the matrix, while the indices $\alpha,\beta = 0,1$ index the elements within each of the 2×2 blocks. Therefore, in the usual matrix notation, the element $H_{i,\alpha;j,\beta}$ corresponds to the $H_{2i+\alpha,2j+\beta}$ element, as shown in Figure 1.

Note that the Majorana operators satisfy the CARs

$$\{\hat{c}_{i,\alpha}, \hat{c}_{j,\beta}\} = 2\delta_{\alpha,\beta}\delta_{i,j}\hat{\mathbb{1}}. \tag{5}$$

[1]The code is available at https://github.com/inafergra/VQA-Fermionic-Systems

$$H = \begin{bmatrix} H_{0,0;0,0} & H_{0,0;0,1} & \dots & H_{0,0;n,0} & H_{0,0;n,1} \\ H_{0,1;0,0} & H_{0,1;0,1} & \dots & H_{0,1;n,0} & H_{0,1;n,1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ H_{n,0;0,0} & H_{n,0;0,1} & \dots & H_{n,0;n,0} & H_{n,0;n,1} \\ H_{n,1;0,0} & H_{n,1;0,1} & \dots & H_{n,1;n,0} & H_{n,1;n,1} \end{bmatrix}$$

Figure 1: Form of the matrix $H_{i,\alpha;j,\beta}$.

We define the global parity operator $\hat{\mathbb{P}}$ as

$$\hat{\mathbb{P}} := \prod_{i=0}^{n-1} \mathrm{i}\hat{c}_{i,0}\hat{c}_{i,1}. \tag{6}$$

Since $[\hat{\mathbb{P}}, \hat{\mathcal{H}}] = 0$, its ground state has a well defined parity $p = \pm 1$ given by the eigenvalue of $\hat{\mathbb{P}}$. Thus, the Hilbert space is splitted in two subspaces with opposite parity, $\mathcal{H} = \mathcal{H}_{+1} \oplus \mathcal{H}_{-1}$. Intuitively, the parity relates to the number of occupied Dirac modes in the system, and it will be even ($p = +1$) if an even number of modes are occupied and odd ($p = -1$) if an odd number of modes are occupied.

2.1 Exact diagonalization

Free-fermion models are known to be exactly solvable [5]. The matrix H can be taken real antisymmetric without loss of generality due to (5). By *Darboux theorem* [26], every real antisymmetric matrix H admits a Williamson decomposition $H = OJO^T$, where $O \in \mathcal{O}(2n)$ is an orthogonal transformation and J is block diagonal:

$$J = \bigoplus_{k=0}^{n-1} \begin{pmatrix} 0 & \epsilon_k \\ -\epsilon_k & 0 \end{pmatrix}. \tag{7}$$

Changing the sign of one of the ϵ's corresponds to an inversion in the parity of $\hat{\mathcal{H}}$, as the determinant of H will invert sign. The orthogonal transformation O is uniquely determined up to the signs of the ϵ's. This O induces a new set of Majorana operators $\{\hat{d}_{k,\gamma}\}$

$$\hat{d}_{k,\gamma} = \sum_{i\beta} O_{i,\beta;k,\gamma}\hat{c}_{i,\beta} \tag{8}$$

that diagonalize the Hamiltonian

$$\hat{\mathcal{H}} = \mathrm{i}\sum_{k=0}^{n-1} \epsilon_k \hat{d}_{k,0}\hat{d}_{k,1}. \tag{9}$$

Eq. (9) represents a diagonal Hamiltonian as the operators $\{\mathrm{i}\hat{d}_{k,0}\hat{d}_{k,1}\}$ mutually commute. The new operators $\{\hat{d}_{k,\gamma}\}$ still satisfy the CARs in Eq. (5). The minimal eigenvalue E_0 of $\hat{\mathcal{H}}$ is given by

$$E_0 = \sum_{k=0}^{n-1} s_k \epsilon_k \ , \tag{10}$$

achieved on a simultaneous eigenstate of all the operators $\{\mathrm{i}\hat{d}_{k,0}\hat{d}_{k,1}\}$, with respective eigenvalue $s_k = -sign(\epsilon_k)$. The second energy level is given by $E_1 = E_0 + 2\min_k |\epsilon_k|$. In the non-degenerate case, the matrix O can be found using the Spectral Theorem: since $H = O^T JO$ is antisymmetric, then H^2 is symmetric and diagonalizes as $H^2 = O^T DO$ with D a diagonal matrix with eigenvalues $-\epsilon_k$ [5].

3 Solving the free-fermion model

In this section we present two algorithms to approximately prepare the ground state of the free-fermion Hamiltonian. We fist introduce the building blocks of the algorithms (Section 3.1) and show how to map fermions into sping operators to implement the algorithms in a quantum computer (Section 3.2). Next, we introduce the adaptative algorithmic cooling method (Section 3.3). We finalize the section by presenting an algorithm based on an algorithm to semi-diagonalize antisymmetric matrices known as Paardekooper algorithm (Section 3.4).

3.1 Givens rotations

The building blocks of our method are the operators corresponding to the time evolution of the individual terms of the free-fermion Hamiltonian (3) $\mathrm{i}\hat{c}_{k,\gamma}\hat{c}_{l,\delta}$, with $k = 0, \ldots, n-1$ and $\gamma = 0, 1$, i.e., the time evolution operator is

$$\hat{G}^{(k,\gamma;l,\delta)}(t) = e^{\mathrm{i}\left(\mathrm{i}\hat{c}_{k,\gamma}\hat{c}_{l,\delta}\right)t} = e^{-\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t}, \tag{11}$$

with $k, l = 1 \ldots n$ and $\gamma, \delta = 0, 1$. From the CARs (5) it follows that the product $\hat{c}_{k\gamma}\hat{c}_{l\delta}$ is anti-Hermitian, so the operator $e^{-\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t}$ is unitary and represents a valid quantum operation.

In appendix A we show that the operator $\hat{G}^{(k,\gamma;l,\delta)}(t)$ acting on the Hilbert space of n fermionic modes induces a change of basis

$$\left(\hat{G}^{(k,\gamma;l,\delta)}(t)\right)^{\dagger} \hat{c}_{i,\alpha} \left(\hat{G}^{(k,\gamma;l,\delta)}(t)\right) = \sum_{j,\beta} G^{(k,\gamma;l,\delta)}_{i,\alpha;j,\beta}(t)\hat{c}_{j,\beta} \tag{12}$$

given by the rotation $G^{(k,\gamma;l,\delta)}(t) \in S\mathcal{O}(2n)$. We have denoted the operator $\hat{G}^{(k,\gamma;l,\delta)}(t)$ acting on the fermionic Hibert space with a hat, and the associated rotation matrix $G^{(k,\gamma;l,\delta)}(t)$ with no hat. The Hamiltonian (3) transforms upon this change of basis as

$$\left(\hat{G}^{(k,\gamma;l,\delta)}(t)\right)^{\dagger} \hat{\mathcal{H}} \left(\hat{G}^{(k,\gamma;l,\delta)}(t)\right) = \frac{\mathrm{i}}{2} \sum_{i,j=0}^{n-1} \sum_{\alpha,\beta=0}^{1} H'_{i,\alpha;j,\beta}\hat{c}_{i\alpha}\hat{c}_{j\beta} \ , \tag{13}$$

so it can still be written as quadratic in Majorana operators, but with a different coupling matrix H' given by

$$H' = \left(G^{(k,\gamma;l,\delta)}(t)\right)^{T} H \left(G^{(k,\gamma;l,\delta)}(t)\right). \tag{14}$$

The rotation $G^{(k,\gamma;l,\delta)}(t)$ can be written in matrix notation (see appendix A) as

$$G^{(k,\gamma;l,\delta)}(t) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos(2t) & \cdots & \sin(2t) & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -\sin(2t) & \cdots & \cos(2t) & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & \ldots & 0 & \cdots & 1 \end{bmatrix}, \tag{15}$$

where the diagonal elements are all 1 except from $G^{(k,\gamma;l,\delta)}_{k,\gamma;k,\gamma}(t) = G^{(k,\gamma;l,\delta)}_{l,\delta;l,\delta}(t) = \cos(2t)$ and the off-diagonal elements are zero except from $G^{(k,\gamma;l,\delta)}_{k,\gamma;l,\delta}(t) = -G^{(k,\gamma;l,\delta)}_{l,\delta;k,\gamma}(t) = \sin(2t)$. A matrix of this form is known as a *Givens rotation*. The product $\left(G^{(k,\gamma;l,\delta)T}(t)\right) H \left(G^{(k,\gamma;l,\delta}(t)\right)$ rotates the $2k+\gamma$ row of H with the $2l+\delta$ row and the $2k+\gamma$ column with the $2l+\delta$ column by an angle $2t$.

3.2 Implementation in a quantum computer

Here we briefly discuss how the operator $\hat{G}^{(k,\gamma;l,\delta)}(t) = e^{-\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t}$can be implemented in a quantum circuit. The first step is to map the second-quantized fermionic operators to spin operators. This mapping must be isospectral, i.e., the eigenspectrum of the operators must be conserved. Moreover, an additional constraint needs to be taken account. As we saw before, different fermionic modes anticommute (CARs 2), while spin operators on different sites commute. Thus, the mapping should also preserve the fermionic CARs so that both formalisms are equivalent. A variety of such mappings exist, but here we employ the famous Jordan-Wigner transformation, which maps a Majorana fermion $\hat{c}_{k,\gamma}$ into

$$\hat{c}_{k,\gamma} \leftrightarrow (-1)^{\gamma} \left(\prod_{j=0}^{k-1} \sigma_z^{(j)} \right) \sigma_{x+\gamma}^{(k)}, \quad \gamma \in \{0,1\}. \tag{16}$$

where σ_x^j, σ_y^j, σ_z^j the Pauli matrices acting on qubit j, and we denote $x+1=y$ for short. The JW transformation intuitively maps the (un)occupied k-th fermionic mode into the $(|0\rangle)|1\rangle$ computational state of qubit k. This means n qubits are needed to map a system of n fermionic modes. According to Eq. (17), the product $\mathrm{i}\hat{c}_{k,\gamma}\hat{c}_{l,\delta}$ is mapped into a Pauli string P acting non trivially in $|k-l|$ qubit, so the average length is $\mathcal{O}(n)$:

$$\hat{c}_{k,\gamma}\hat{c}_{l,\delta} \leftrightarrow \mathrm{i}(-1)^{\delta}\sigma_{x+1-\gamma}^{(k)} \left(\prod_{j=k+1}^{l-1} \sigma_z^{(j)} \right) \sigma_{x+\delta}^{(l)}, \quad \gamma,\delta \in \{0,1\}, \tag{17}$$

where we assumed, without loss of generality, that $k<l$. In other words, implementing $e^{-\hat{c}_{k,\gamma}\hat{c}_{jl,\delta}t}$ requires implementing the exponential of a Pauli string e^{iPt}, which can be efficiently simulated via CNOT gates and single qubit rotations [1, 2, 27].

Let us illustrate this with an example. Take, for instance, the product $\hat{c}_{k,0}\hat{c}_{k+1,1}$, which is mapped by (17) into $\mathrm{i}\sigma_x^{(k)}\sigma_x^{(k+1)}$. The exponential $e^{-\hat{c}_{k,0}\hat{c}_{k+1,1}} \leftrightarrow e^{-\mathrm{i}\sigma_x^{(k)}\sigma_x^{(k+1)}}$ can then be implemented by the quantum circuit showed in Figure 2. This can be done in general for any product.

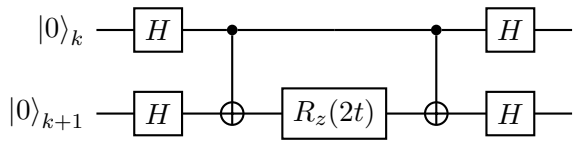

Figure 2: Quantum circuit implementing $e^{it\sigma_x^{(k)}\otimes\sigma_x^{(k+1)}}$. H represents a Hadamard gate and $R_z(t) = e^{it\sigma_z/2}$ a rotation of t around the z axis.

3.3 Adaptative algorithmic cooling of the free-fermion model

As a first step, we apply the Givens rotations in a random order. We choose as an initial state the fermionic vacuum $|\Omega\rangle$, which according to the JW transformation (17) maps into the $|0\rangle^{\otimes n}$ state in the computational basis, i.e. $|\Omega\rangle \leftrightarrow |0\rangle^{\otimes n}$. We sequentially apply Givens rotations to fermionic vacuum optimizing the angles t_k of the rotations such that the energy is minimized after each rotation, as shown in Figure 3. We refer to this procedure as *adaptative algorithmic cooling*. It is an adaptative scheme as Givens rotations are applied until the desired accuracy is reached. We use the Nelder-Mead optimizer [28] to optimize the angles. For the numerical simulations, the elements of the coupling matrix H are drawn from a normal distribution $\mathcal{N}(0,1)$. The indices of the rotations, $(k,\gamma;l,\delta)$, are taken randomly, making sure that $k \neq l$ and $\gamma \neq \delta$ to ensure the unitarity of the operator.

Since $G^{(k,\gamma;l,\delta)}(t) \in SO(2n)$ is a rotation, the parity of the initial state, defined in Eq. (6), is conserved by $\hat{G}^{(k,\gamma;l,\delta)}(t)$, and thus one cannot reach the ground state by only applying operators of

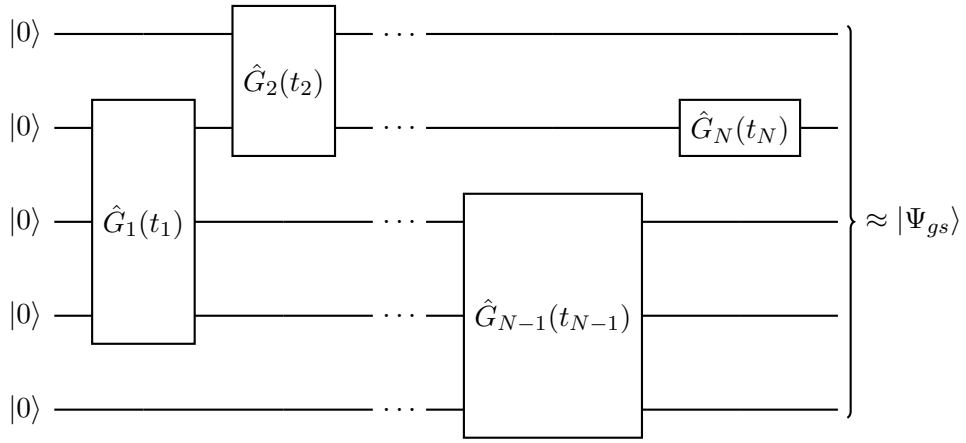

Figure 3: Scheme of the algorithmic cooling procedure consisting on $i = 1, \ldots, N$ cooling operators. Operators of the form $\hat{G}_i(t_i) = \hat{G}^{(k_i,\gamma_i;l_i,\delta_i)}(t_i) = e^{-\hat{c}_{k_i,\gamma_i}\hat{c}_{l_i,\delta_i}t_i}$ are applied sequentially, and the times t are optimized such that the energy decreases monotonically until the desired accuracy is reached. The output state is then an approximation of the ground state $|\Psi_{gs}\rangle$.

this form if the initial state has a different parity than the ground state. Since one cannot predict the parity of the ground state beforehand, this means the algorithm can get stuck and converge to the first excited state instead of the ground state. This issue can easily be overcome by executing the algorithm twice with initial states of opposite parity, e.g. $|0\rangle^{\otimes n}$ $(p = +1)$ and $|1\rangle \otimes |0\rangle^{\otimes n-1}$ $(p = -1)$, which will output the ground state and the first excited state. We can then pick the state with the lowest energy as the ground state. We show typical runs of the algorithm in Figure 4 for a system $n = 10$. The algorithm successfully converges to the ground state or the first excited state dependending on the parity. Later in Section 3.5 we will further study the number of rotations needed to prepare the ground state. It is also worth mentioning that, when implementing in a quantum circuit, the number of measurements needed to achieve enough precision in the energy will increased more operators are added. This is due to the fact that, when close to the ground state energy, the energy does not vary much with each rotation, meaning that a high precision will be needed to keep track of the energy variations, which translates in a large number of measurements.

3.4 Paardekooper-based algorithm

In the previous section we saw that we can "cool" a free-fermion model by sequentially applying Givens rotations to drive the system to the ground state. As a first step, the plane on which the rotations acted and the order on which the rotations were applied was determined by randomly choosing the four indices $(k, \gamma; l, \delta)$ for each of the rotations. The angle of the rotation was then optimized to minimize the energy.

A natural question is if a more optimal sequence of such rotations exists, such that the system can be more effectively driven to the ground state using a reduced number of those rotations. In this work, we take inspiration from computational algebra methods to answer this question. Givens rotations are frequently used in computational algebra algorithms - for instance, they can be used to perform the QR decomposition of a square matrix $A = QR$, where A is a square matrix, $Q \in \mathcal{O}(n)$ is orthogonal and R is an upper triangular matrix. *Jacobi-type algorithms* make use Givens rotations to iteratively diagonalize $(2n \times 2n)$ symmetric matrices [29, 30], requiring in general $(2n-1)(2n-2)/2$ rotations (which can be greatly reduced if the matrix is sparse). Although the symmetric eigenvalue problem is far more studied, similar ideas have been developped for the antisymmetric case as well [11], where the goal is to bring an antisymmetric matrix to the Williamson form in (7).

The antisymmetric case will be the main focus of this section, as the coupling matrix H is anti-

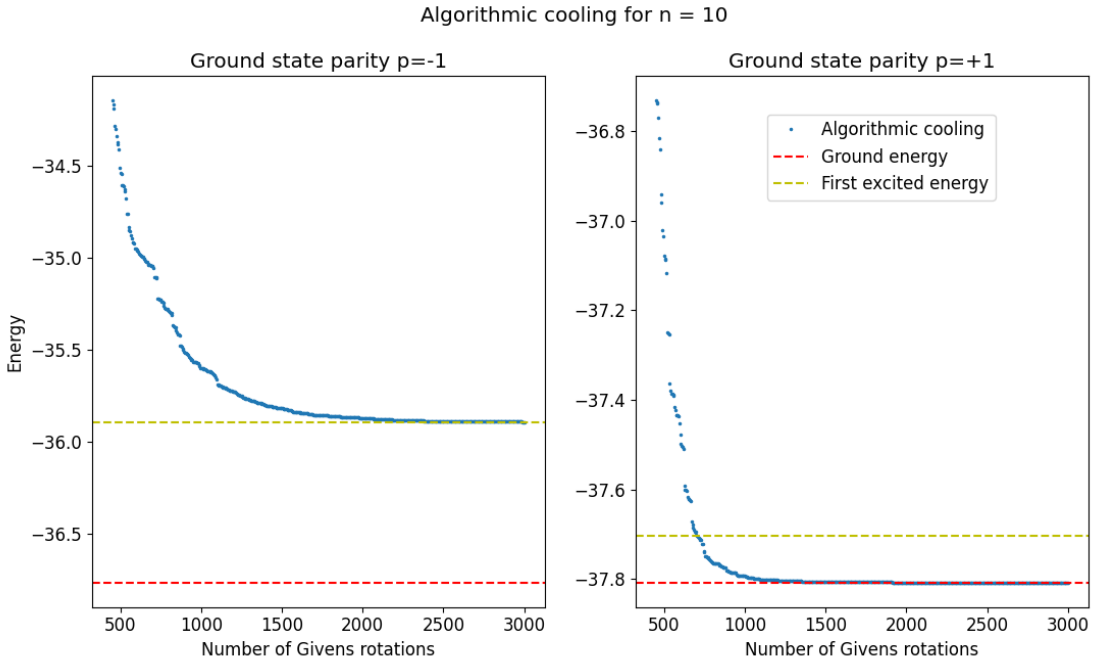

Figure 4: Two different runs of the algorithmic cooling of the free-fermion Hamiltonian of size $n = 10$ by applying Givens rotations. The initial state is the fermionic vacuum $|\Omega\rangle = |0\rangle^{\otimes n}$, which has a parity $p = +1$. On the left, we show a run for an instance of the Hamiltonian where the ground state has different parity ($p = -1$) to the initial state, and therefore it is not possible to reach the ground state by only using Givens roations. The first excited state, which in this case has parity $p = +1$, is reached instead. On the right, the ground state and the initial state both have parity $p = +1$, so the ground state can be attained.

symmetric. To bring the system to the ground state (in fact, to any eigenstate), the matrix H must be brought into the Williamson form (7), a procedure we refer as *semi-diagonalization*. In Section 2.1, we saw that the antisymmetric coupling matrix H of a free-fermion Hamiltonian can be brought to a Williamson form by an orthogonal matrix $O \in \mathcal{O}(2n)$. This matrix O can be decomposed as a sequence of Givens rotations $O = \prod_i G_i$ as long as $O \in S\mathcal{O}(2n)$, this is, as long as O is also a rotation. If O is a reflection then this decomposition exclusively in terms of Givens rotations would no longer be possible, and we would have to add a reflection term in the decomposition that changes the parity, as we saw in the previous section in the case when the algorithmic cooling converged the first excited state.

3.4.1 Paardekooper algorithm

Various algorithms exist to solve the antisymmetric eigenvalue problem [11]. The algorithm first proposed by Paardekooper [12] brings an anti-symmetric matrix H to an (approximate) Williamson form by repeatedly anhilitating 2×2 off-diagonal blocks of the original matrix H using Givens rotations. Paardekooper algorithm aims to reduce the *off-diagonal norm* τ of the matrix

$$\tau(H) = \sqrt{\sum_{i\neq j} \|H_{ij}\|_F} \quad , \tag{18}$$

where H_{ij} denotes each of the $n(n-1)$ off-diagonal (2×2) blocks of H and $\|\cdot\|_F$ denotes Frobenius norm, $\|H_{i,j}\|_F = \sqrt{\sum_{\alpha=0}^{1}\sum_{\beta=0}^{1}|H_{i,\alpha;j,\beta}|^2}$. A step of the algorithm consists of the orthogonal similarity transformation

$$H' = P_{pq}^T H P_{pq} \tag{19}$$

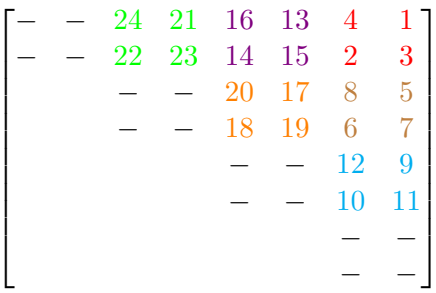

Figure 5: Illustration of the pivot strategy used in this work, for an 8×8 matrix. The elements are labeled according to the order they are zeroed-out. A single Givens rotations zeroes out one element. Each color represents an off-diagonal 2×2 block, which is annhilated by a Jacobi annihilator. Since H is antisymmetric and we apply similarity transformations of the type GHG^T, the lower-triangular part of the matrix is zeroed out simultaneously with the upper part.

which zeroes out the off-diagonal blocks H_{pq} and H_{pq}, since the H is antisymmetric. The matrix P_{pq} is referred to as the *Jacobi annihilator* of the 2×2 H_{pq}, and it consists of 4 successive Givens rotations

$$P_{pq} = G^{(p,0;q,0)}\left(t_1/2\right) \cdot G^{(p,1;q,1)}\left(t_2/2\right) \cdot G^{(p,0;q,1)}\left(t_3/2\right) \cdot G^{(p,1;q,0)}\left(t_4/2\right) \tag{20}$$

The angles t_i, $i = 1...4$ are presented in appendix B. The main idea of the algorithm is to run over all the 2×2 off-diagonal blocks H_{pq} of the matrix H, applying the respective Jacobi annihilator P_{pq} in order to reduce the off-diagonal norm $\tau(H)$. The algorithm would be straightforward if the only action of the Jacobi annhilator P_{pq} would be to zero out the H_{pq} block. Unfortunately, this is not the case as the Givens rotations will rotate the entire columns and rows contained in the H_{pq} block. In order words, as we move through the matrix zeroing out the 2×2 blocks we will also fill in previously created zeroes. Therefore, in general we will need to iteratively run through the whole matrix more than once . We refer as a run through all of the blocks H_{pq} as a sweep, and denote the number of sweeps used as j.

After annihilating the off-diagonal block H_{pq}, $\tau(H)$ is reduced by

$$\tau(H)^2 \to \tau(H)^2 - 2 \cdot ||H_{pq}||_F \tag{21}$$

which garantees that $\tau \to 0$ when the number of sweeps j tends to infinity.

A final step is to establish *pivot strategy* that sets the order in which the 2×2 blocks are annihilated. The convergence and performance of the algorithm depends on the considered strategy. The current best strategy is the one proposed in [31], providing a cubic convergence under certain conditions. In this work, we employ a naive pivot strategy going column by column, as shown in Figure 5.

Once we have set a pivot strategy, Paardekooper algorithm provides a matrix Ω representing a sweep of the algorithm. Iterating this sweep j times gives rise to a matrix we denote

$$U^{(j)}_{paard} = \prod_{r=1}^{j} \Omega_r \ . \tag{22}$$

After applying this operator j times, the norm of the off-diagonal blocks of the coefficients, $\tau(H)$, will be reduced. For $j \to \infty$, this norm tends to zero so the operator $U^{(\infty)}$ will semi-diagonalize H,

$$\left(U^{(\infty)T}_{paard}\right) H \left(U^{(\infty)}_{paard}\right) = \bigoplus_{k=0}^{n-1} \begin{pmatrix} 0 & \epsilon_k \\ -\epsilon_k & 0 \end{pmatrix} \tag{23}$$

this is, in the limit $j \to \infty$, $U^{(j)}$ is equivalent to the orthogonal matrix O used in the Williamson decomposition (7). By equivalent we mean that the ϵ's in in the Williamson form are equal up to their

sign. For finite j, the output state of the algorithm, $|\Phi\rangle_{out}$, is approximately an eigenstate of the $\hat{\mathcal{H}}$. It is important to note that we do not have any control over which of the eigenstates is the final state, as this depends on the signs of the ϵ_k after the diagonalization (recall that the transformation O is unique up to the choice of the signs of ϵ_k). If we are interested in a particular eigenstate of the system, for instance the ground state, we must do some additional work that allows us to drive any arbitrary eigenstate to the ground state. This is done in the next section.

3.4.2 Greedy algorithm

Once the coupling matrix is in (approximate) Williamson form, it means that we have transformed the initial state into an (approximate) arbitrary eigenstate of the system. In order to get to the ground state, we use Givens rotations of the form $G^{(k,\gamma;k+m,\delta)}(t=\frac{\pi}{2}) = G(\frac{\pi}{2})$, with $k = 0, \dots n-1$, $m = 0, \dots, n-k-1$ and $\gamma, \delta = 0, 1$ has the form

$$G^{(k,\gamma;k+m,\delta)}\left(\frac{\pi}{2}\right) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & -1 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & -1 & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \cdots & 1 \end{bmatrix} \tag{24}$$

with $G^{(k,\gamma;k+m,\delta)}_{k,\gamma;k,\gamma}(\frac{\pi}{2}) = G^{(k,\gamma;k+m,\delta)}_{k+m,\delta;k+m,\delta}(\frac{\pi}{2}) = -1$. The only action of acting on the coupling matrix as $G^{(k,\gamma;k+m,\delta)}(\frac{\pi}{2})^T H G^{(k,\gamma;k+m,\delta)}(\frac{\pi}{2})$, is to invert the sign of a number of ϵ_k in the diagonal blocks of H. Specifically, the columns and rows corresponding to the k-th and $(k+m)$-th blocks will invert sign. Using this kind of Givens rotations, we can devise a greedy-type algorithm where we apply rotations of this type only if the energy is reduced. Recall that the ground energy (10) is achieved if the signs are such that . A scheme of the algorithm is depicted in Algorithm 1. This methods converges to either the ground state (if the initial state has the same parity as the ground state) or to the first excited state otherwise. As we mentioned in the case of the algorithmic cooling, the solution is again to repeat the algorithm twice with initial states of different parity, picking as a ground state the one with the lowest energy.

Applying Paardekooper algorithm followed by the greedy algorithm successfully achieves the ground state of the free-fermion Hamiltonian, again if the parity of the ground state and the initial state are the same. We refer to this method as the *Paardekooper-based* algorithm.

3.5 Numerical results

In Figure 6 we show the number of Givens rotations needed to achieve an energy error $|E - E_{exact}|/E_{exact}$ smaller than 1% for both the algorithmic cooling and the Paardekooper-based algorithm. We plot the results in a loglog scale to visually display the polynomial scaling. We perform a polynomial fit, showcasing that in both cases the number of rotations scales almost quadratically. Specifically,the algorithmic cooling scales as $\mathcal{O}(1.81n^{1.92})$, while Paardekooper algorithm scales as $\mathcal{O}(0.67n^{2.12})$. Thus, for small sytem size n Paardekooper algorithm is more efficient due to the smaller multiplicative coefficient in the scaling ($0.67 < 1.81$). Nevertheless, the scaling exponent is smaller in the algorithmic cooling case ($1.92 < 2.12$), hence for large n the algorithmic cooling will require less rotations. It is interesting to see that the heuristics of the algorithmic cooling is more efficient than the Paardekooper algorithm for large sizes. Specifically, the polynomial fits shows that Paardekooper is more efficient only for sizes $n < 284$.

Algorithm 1 Greedy algorithm

```
#Begin with H in Williamson form
E = Energy(H)
for k ← 0 to 2n − 1 do
  for m ← 0 to 2n − k − 1 do
    for a, b ← 0 to 1 do
      G = GivensRotation(k; k + m; π/2)
      H′ = G^T H G
      E′ = Energy(H′)
      if E′ < E then
        H = H′
        E = E′
      end if
    end for
  end for
end for
```

The number of Givens rotations used in the Paardekooper-based algorithm is calculated as $4jn(n-1)/2 + S$, where j is the number of sweeps used and n the number of fermionic Dirac modes in the system, as there are $n(n-1)/2$ independent 2×2 off-diagonal blocks in the matrix and it takes 4 Givens rotations to zero each of them out. S is the number of Givens applied in the greedy algorithm, which is dependent on the eigenstate in which we end up after semi-diagonalizing the matrix.

4 Gaussian approximation of the SYK model

So far, both algorithms presented in the previous section have been proved to successfully approximately prepare the ground state of a free-fermion Hamiltonian, as long as the initial state is in the parity subspace of the ground state. Ground states of such Hamiltonians are known in the literature as *Gaussian states* and can be formally defined as follows [14]. Let U_r be an unitary operator acting on the Hilbert space of N Majorana modes as

$$(\hat{U}_r)^\dagger \hat{c}_{i,\alpha}(\hat{U}_r) = \sum_{j,\beta} R_{i,\alpha;j,\beta}\hat{c}_{j,\beta} \tag{25}$$

with $R \in \mathcal{O}(n)$ an orthogonal matrix. Then, a state $|\psi\rangle$ is Gaussian if it can be written as $|\psi\rangle = \hat{U}_r|\Omega\rangle$, with $|\Omega\rangle$ being the fermionic vacuum. We note that the unitary $\hat{U}_r$ is completely determined by the matrix R up to the phase. Looking back to Eq. (37), we notice that the building block of our algorithm, the operator $\hat{G}(t)$, corresponds to the special case of U_r when R is a Givens rotation. This means that by using operators $\hat{G}(t)$ we are restricted to transformations of the type (25) where R has determinant $+1$, and therefore the final state of the algorithms is restricted to the subspace of Gaussian states that lie in the same parity subspace than the initial state.

In this section we review how the adaptative algorithmic cooling procedure proposed in this work can be be extended to yield a Gaussian state approximation to the ground state of other fermionic models apart from free-fermion. We use the Sachdev–Ye–Kitaev (SYK) model as a testbed [17]. The SYK model has recently drawn a considerable amount of attention since it was proposed as a dual model for particular quantum theories of gravity [18, 19]. Digital [32, 33] and analog [34] quantum simulations of the SYK model have been proposed.

In [16], strong evidence was presented showing that the approximation ratio to the ground state energy of Gaussian states for the SYK model, $r = E_{Gauss}/E_{exact}$, goes to zero for increasing system size: $r \to 0$ for $n \to \infty$. In other words, this means that the difference between the exact ground energy and the best Gaussian approximation to the ground state energy will tend to infinite as the

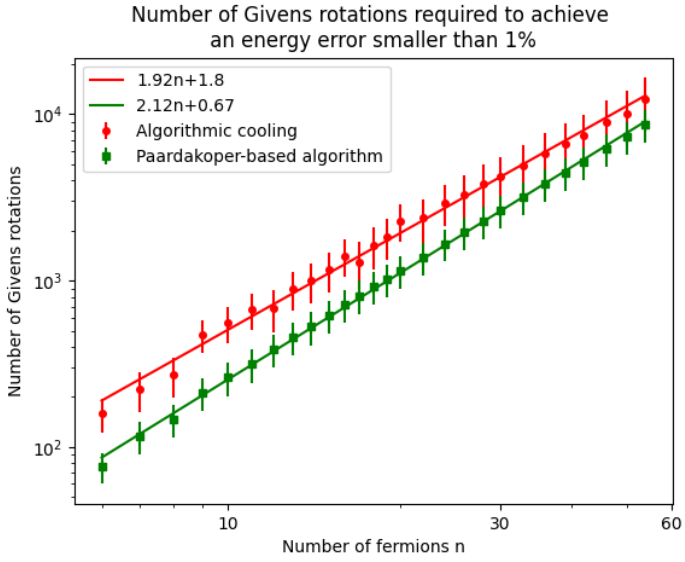

Figure 6: In red, the average number of Givens rotations needed to achieve an energy error $|E - E_{exact}|/E_{exact} \leq 1\%$ in the free-fermion Hamiltonian when applying the Givens rotations randomly (algorithmic cooling explained in Section 3.3). In green, the average number of Givens rotations required to achieve the same energy error when using Paardekooper-based algorithm, followed by the Greedy algorithm to drive the system to the ground state. Each point was calculated as the average was taken over 25 disorder realizations of the Hamiltonian. The results are fitted to a polynomial function, showing that the number of Givens rotations scales almost quadratically in both cases. The fit reveals that the Paardekooper-based algorithm is consistently more efficient n smaller than ≈ 284.

system size increases. The ground state of the SYK model is therefore, in some way, as non-Gaussian as it can be. This motivates the extension of the algorithmic cooling to the SYK model as a way to quantitive measure its non-Gaussianity, as we numerically show that no meaningful approximation of the ground state can be achieved via Gaussian transformations.

The SYK model is a system of $2n$ Majorana fermions with all possible four body interactions:

$$\hat{\mathcal{H}}_{SYK} = \sum_{i_1 \ldots i_4=0}^{n-1} \sum_{\alpha_1 \ldots \alpha_4=0}^{1} J_{i_1,\alpha_1;\ldots;i_4,\alpha_4}\ \hat{c}_{i_1,\alpha_1}\hat{c}_{i_2,\alpha_2}\hat{c}_{i_3,\alpha_3}\hat{c}_{i_4,\alpha_4}\ , \tag{26}$$

where $\hat{c}_{i,\alpha}$ are Majorana fermions defined in Eq. (43). The couplings $J_{i_1,\alpha_1;\ldots;i_4,\alpha_4}$ are drawn randomly from a Gaussian distribution $\mathcal{N}(1, 3!J^2/(2n)^3)$, and satisfy the proper antisymmetrization conditions according to the Majorana's CARs in Eq. (5) and to ensure the hermiticity of the Hamiltonian.

To avoid dealing with many indices, in this section we will rewrite the pair of indices (i, α), with $i = 0, \ldots, n-1$ and $\alpha = 0, 1$, used in the previous sections to index the Majorana fermions, as a single index $\mathbf{i} = 0, \ldots, 2n-1$ such that $\mathbf{i} = 2i + \alpha$:

$$\hat{\mathcal{H}}_{SYK} = \sum_{\mathbf{i},\mathbf{j},\mathbf{k},\mathbf{l}=0}^{2n-1} J_{\mathbf{i},\mathbf{j},\mathbf{k},\mathbf{l}}\ \hat{c}_{\mathbf{i}}\hat{c}_{\mathbf{j}}\hat{c}_{\mathbf{k}}\hat{c}_{\mathbf{l}}\ . \tag{27}$$

4.1 Adaptative algorithmic cooling of the SYK model

We extend here the adaptative algorithmic cooling presented in Section 3.3 to the SYK model. The Paardekooper-based algorithm proposed in Section 3.4 does not have a trivial extension to the SYK model, as now the couplings are stored in a 4-tensor $J_{\mathbf{i},\mathbf{j},\mathbf{k},\mathbf{l}}$, instead of a matrix that can be semi-diagonalized (although work in tensor diagonalization could be serve as a basis to extend the algorithm,

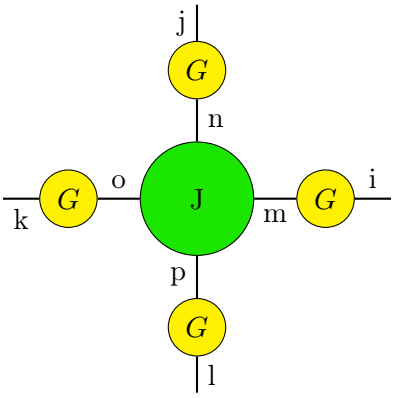

Figure 7: Tensor diagram representing the transformation in Eq. (29). Circles represent tensors while lines represent indices. A line connecting two circles means that the index is contracted between both tensors. Lines only connected to one circle stand for free indices.

as we will comment later). Using an analog approach to that shown in appendix A it is straightforward to see how the operator $\hat{G}^{(\mathbf{j}_1,\mathbf{j}_2)}(t) = e^{-\hat{c}_{\mathbf{j}_1}\hat{c}_{\mathbf{j}_2}t}$ transforms the Hamiltonian $\hat{\mathcal{H}}_{SYK}$ as

$$\left(\hat{G}^{(\mathbf{j}_1,\mathbf{j}_2)}(t)\right)^\dagger \hat{\mathcal{H}}_{SYK} \left(\hat{G}^{(\mathbf{j}_1,\mathbf{j}_2)}(t)\right) = \sum_{\mathbf{i},\mathbf{j},\mathbf{k},\mathbf{l}=0}^{2n-1} J'_{\mathbf{i},\mathbf{j},\mathbf{k},\mathbf{l}}\ \hat{c}_{\mathbf{i}}\hat{c}_{\mathbf{j}}\hat{c}_{\mathbf{k}}\hat{c}_{\mathbf{l}}, \tag{28}$$

with the new coupling tensor $J'(t)$ given by

$$J'_{\mathbf{i},\mathbf{j},\mathbf{k},\mathbf{l}} = \sum_{\mathbf{m},\mathbf{n},\mathbf{o},\mathbf{p}} J_{\mathbf{m},\mathbf{n},\mathbf{o},\mathbf{p}}\ (G^{(\mathbf{j}_1,\mathbf{j}_2)})_{\mathbf{i},\mathbf{m}}(G^{(\mathbf{j}_1,\mathbf{j}_2)})_{\mathbf{j},\mathbf{n}}(G^{(\mathbf{j}_1,\mathbf{j}_2)})_{\mathbf{k},\mathbf{o}}(G^{(\mathbf{j}_1,\mathbf{j}_2)})_{\mathbf{l},\mathbf{p}}\ , \tag{29}$$

where $G^{(\mathbf{j}_1,\mathbf{j}_2)}$ is a Givens rotation as shown in Eq. (24). A tensor diagram representing the index contractions in Eq. (29) is shown in Figure 7.

In Figure 8 we show typical runs of the algorithm for systems sizes $n = 3, 4, 5, 6$. Note that a system with n fermionic (Dirac) modes is represented by $2n$ Majorana modes. We also show the average approximation ratio to the ground state energy r for each of the sizes. The approximation ratio is lower as the system size increases, in line with the results in [16]. Only for the smallest size $n = 3$, the algorithm converges to the ground state due to the small number of free parameters at this size. As expected, the number of Givens rotations for the algorithmic cooling needed to converge also scales worse than in the free-fermion case. In particular, for the SYK model it scales roughly as $\mathcal{O}(n^4)$, corresponding with the number of free parameters in the coupling tensor. This is shown in Figure 9.

Let us now reflect on why the SYK Hamiltonian cannot be driven to ground the state through transformations of the type (29). In the free-fermion case, the coupling matrix H could be semi-diagonalized by the Williamson decomposition $H = O^T J O$, where J is block diagonal of the form in Eq. (7). The matrix is an orthogonal rotation $O \in \mathcal{O}(2n)$, and therefore this O could be decomposed (up to parity) in Givens rotations, as Givens rotations are also orthogonal. Once the coupling matrix was in a block diagonal form, we had full information about the spectrum.

It is reasonable to ask if there is an analogue procedure for the SYK model. In the SYK model the couplings are now expressed as a 4-order tensor instead of a matrix, which already poses the first challenge: what does it mean to diagonalize a tensor? Turns out that this question has given rise to a recent line of research that extends concepts of matrix diagonalization to higher order tensors [35, 36]. The fact that the ground state of the SYK model is not a Gaussian state means that the tensor bringing J to a diagonal form (whatever diagonalizing the SYK tensor J means), i.e. the analog of O in the free-fermion case, is not odeco (orthogonally decomposable) [37]. A $n \times n \times \ldots \times n (d$ times) tensor $T_{i_1,\ldots,i_d}$ is odeco iff we can write $T = \sum_{i=1}^{r} \lambda_i v_i^{\otimes d}$, where $v_1, \ldots, v_n \in \mathbb{R}^n$. In practice, the fact

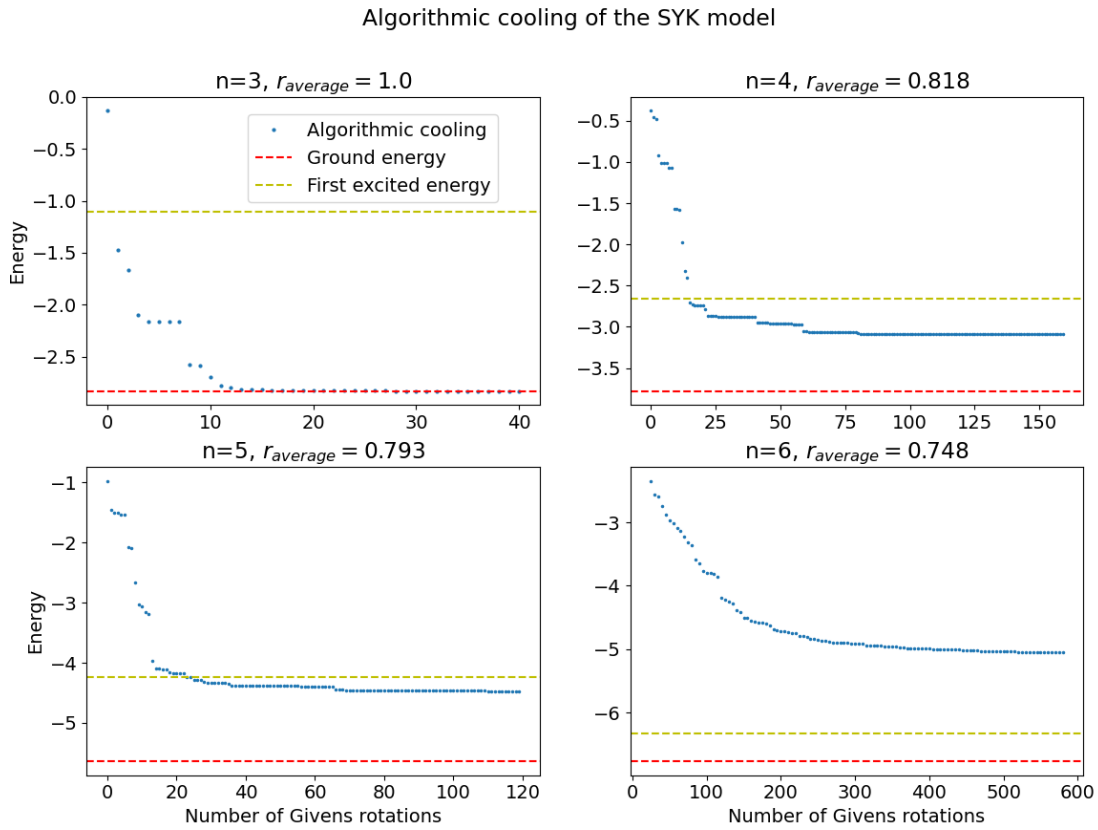

Figure 8: In each plot we show a specific run of the adaptative algorithmic cooling of the SYK Hamiltonian for system sizes $n = 3, 4, 5, 6$. For each system size we show the average approximation ratio to the ground state $r_{average} = E_{cooling}/E_{exact}$, averaged over 15 disorder realizations.

that the coupling tensor of the SYK model is not odeco means that to diagonalize the SYK model, a more general type of transformation on the tensor J than (29) has to be considered. Further research on what kind of transformations diagonalize the SYK Hamiltonian could indicate in what direction the operators need to be generalized.

5 Conclusions

In this work, we have shown how to achieve the ground state of a free-fermion Hamiltonian using operators that act on the coupling matrix of the Hamiltonian as Givens rotations. We have done so first with a variational approach, by applying the operators in an adaptative algorithmic cooling approach. Next we have used Paardekooper semi-diagonalization algorithm for antisymmetric matrices to reduce the number of Givens rotations required to reach the ground state. We have numerically shown that Paardekooper algorithm is more efficient in the number of rotations for $n < 284$. Last, we have extended the randomized algorithmic cooling to the SYK model and seen how it provides a Gaussian state approximation of the ground state, allowing us to quantify the non-Gaussianity of its ground state for small n.

Let us conclude by indicating some future outlooks of this work. In the free fermions setting, an interesting extension would be to combine both of the proposed approaches, by adding a variational part to the Paardekooper-based algorithm. For instance, adding parametrized operators of the type $e^{-\hat{c}_{k,0}\hat{c}_{k+1,1}}$, which we have seen that correspond to spin operators of the type $e^{-i\sigma_x^{(k)}\sigma_x^{(k+1)}}$, and optimizing such that the energy or the variance are minimized. This would likely help to reduce the number

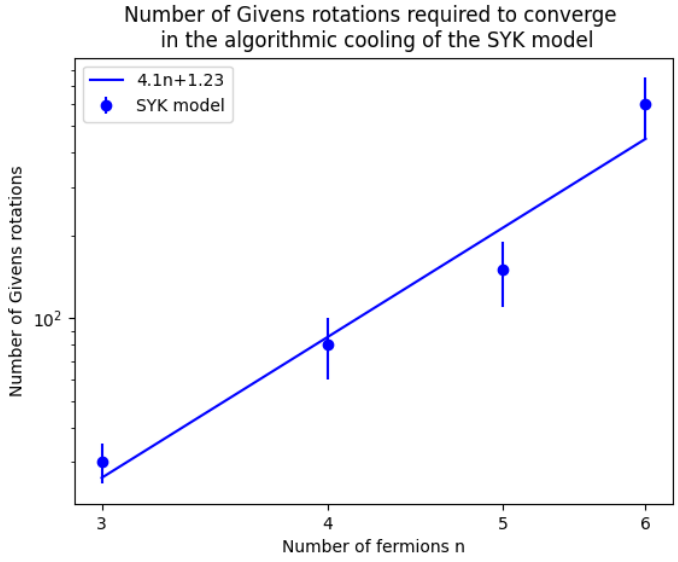

Figure 9: The number of givens rotations required for the algorithmic cooling to converge in the SYK scales roughly as $\mathcal{O}(n^4)$, like the number of free parameters in the coupling tensor $J_{\mathbf{i,j,k,l}}$. The results shown are taken over an average of 5 disorder realizations. Sizes larger than 6 fermions were not numerically tractable.

of sweeps needed in Paardekooper algorithm, effectively reducing the depth of the circuit.

Last, the family of operators used in the algorithms could be extended to provide other type of transformations apart from Givens rotations. In the free fermions case, it would be particularly useful to find transformations that translate into some type of reflections, which would allow us to invert the parity throughout the algorithms. In the SYK case, further research on tensor diagonalization could provide suggestions into how to generalize the transformations, as discussed in Section 4. The algorithms described here could also be used to provide Gaussian approximations to other fermionic models.

6 Acknowledgements

First and foremost, I would like to express my deepest thanks to Jordi Tura, who kindly accepted to supervise the project and from whom I have learned a great deal in the process. I must also thank Johannes Borregaard for acting as a second examiner. In addition, I would like to thank Stefano Polla and Emiel Koridon for fruitful discussions on the topic. Last but not least, thanks to my friends and colleagues Smit Chaudhary and Yorgos Sotiropoulos, who have been an exceptional support in this Honours Programme adventure.

References

[1] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, "Simulation of electronic structure hamiltonians using quantum computers," *Molecular Physics*, vol. 109, no. 5, 735–750, 2011, ISSN: 1362-3028. DOI: 10.1080/00268976.2011.552441. [Online]. Available: http://dx.doi.org/10.1080/00268976.2011.552441.

[2] J. T. Seeley, M. J. Richard, and P. J. Love, "The bravyi-kitaev transformation for quantum computation of electronic structure," *The Journal of Chemical Physics*, vol. 137, no. 22, p. 224 109, 2012, ISSN: 1089-7690. DOI: 10.1063/1.4768229. [Online]. Available: http://dx.doi.org/10.1063/1.4768229.

[3] J. Kempe, A. Kitaev, and O. Regev, "The complexity of the local hamiltonian problem," in *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science*, K. Lodaya and M. Mahajan, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 372–383, ISBN: 978-3-540-30538-5.

[4] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices," *Phys. Rev. X*, vol. 10, p. 021 067, 2 2020. DOI: 10.1103/PhysRevX.10.021067. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.021067.

[5] J. Tura, G. De las Cuevas, R. Augusiak, M. Lewenstein, A. Acín, and J. I. Cirac, "Energy as a detector of nonlocality of many-body spin systems," *Phys. Rev. X*, vol. 7, p. 021 005, 2 2017. DOI: 10.1103/PhysRevX.7.021005. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.7.021005.

[6] P. O. Boykin, T. Mor, V. Roychowdhury, F. Vatan, and R. Vrijen, "Algorithmic cooling and scalable nmr quantum computers," eng, *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 6, pp. 3388–3393, 2002, 11904402[pmid], ISSN: 0027-8424. DOI: 10.1073/pnas.241641898. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/11904402.

[7] D. K. Park, N. A. Rodriguez-Briones, G. Feng, R. Rahimi, J. Baugh, and R. Laflamme, "Heat bath algorithmic cooling with spins: Review and prospects," in *Electron Spin Resonance (ESR) Based Quantum Computing*, T. Takui, L. Berliner, and G. Hanson, Eds. New York, NY: Springer New York, 2016, pp. 227–255, ISBN: 978-1-4939-3658-8. DOI: 10.1007/978-1-4939-3658-8_8. [Online]. Available: https://doi.org/10.1007/978-1-4939-3658-8_8.

[8] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," *Nature Communications*, vol. 10, no. 1, p. 3007, 2019, ISSN: 2041-1723. DOI: 10.1038/s41467-019-10988-2. [Online]. Available: https://doi.org/10.1038/s41467-019-10988-2.

[9] Y.-X. Yao, N. Gomes, F. Zhang, C.-Z. Wang, K.-M. Ho, T. Iadecola, and P. P. Orth, "Adaptive variational quantum dynamics simulations," *PRX Quantum*, vol. 2, p. 030 307, 3 2021. DOI: 10.1103/PRXQuantum.2.030307. [Online]. Available: https://link.aps.org/doi/10.1103/PRXQuantum.2.030307.

[10] L. Bittel and M. Kliesch, "Training variational quantum algorithms is np-hard," *Physical Review Letters*, vol. 127, no. 12, 2021, ISSN: 1079-7114. DOI: 10.1103/physrevlett.127.120502. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.127.120502.

[11] C. F. V. L. Gene H. Golub, *Matrix Computations, 3rd ed.* London: Bantam, 1996.

[12] M. Paardekooper, "An eigenvalue algorithm for skew-symmetric matrices.," *Numerische Mathematik*, vol. 17, pp. 189–202, 1971. [Online]. Available: http://eudml.org/doc/132069.

[13] L. Hackl and E. Bianchi, "Bosonic and fermionic gaussian states from kähler structures," *SciPost Physics Core*, vol. 4, no. 3, 2021, ISSN: 2666-9366. DOI: 10.21468/scipostphyscore.4.3.025. [Online]. Available: http://dx.doi.org/10.21468/SciPostPhysCore.4.3.025.

[14] S. Bravyi, D. Gosset, R. König, and K. Temme, "Approximation algorithms for quantum many-body problems," *Journal of Mathematical Physics*, vol. 60, no. 3, p. 032 203, 2019. DOI: 10.1063/1.5085428. eprint: https://doi.org/10.1063/1.5085428. [Online]. Available: https://doi.org/10.1063/1.5085428.

[15] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, and et al., "Hartree-fock on a superconducting qubit quantum computer," *Science*, vol. 369, no. 6507, 1084–1089, 2020, ISSN: 1095-9203. DOI: 10.1126/science.abb9811. [Online]. Available: http://dx.doi.org/10.1126/science.abb9811.

[16] A. Haldar, O. Tavakol, and T. Scaffidi, "Variational wave functions for sachdev-ye-kitaev models," *Physical Review Research*, vol. 3, no. 2, 2021, ISSN: 2643-1564. DOI: 10.1103/physrevresearch.3.023020. [Online]. Available: http://dx.doi.org/10.1103/PhysRevResearch.3.023020.

[17] S. Sachdev and J. Ye, "Gapless spin-fluid ground state in a random quantum heisenberg magnet," *Phys. Rev. Lett.*, vol. 70, pp. 3339–3342, 21 1993. DOI: 10.1103/PhysRevLett.70.3339. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.70.3339.

[18] A. Kitaev, ""a simple model of quantum holography"."," *Phys. Rev. A*, vol. 104, p. 012 427, 1 2021. [Online]. Available: https://online.kitp.ucsb.edu/online/entangled15/kitaev/,https://online.kitp.ucsb.edu/online/entangled15/kitaev2/.

[19] J. Maldacena and D. Stanford, "Remarks on the sachdev-ye-kitaev model," *Physical Review D*, vol. 94, no. 10, 2016, ISSN: 2470-0029. DOI: 10.1103/physrevd.94.106002. [Online]. Available: http://dx.doi.org/10.1103/PhysRevD.94.106002.

[20] C. Sünderhauf, L. Piroli, X.-L. Qi, N. Schuch, and J. I. Cirac, "Quantum chaos in the brownian syk model with large finite n: Otocs and tripartite information," *Journal of High Energy Physics*, vol. 2019, no. 11, 2019, ISSN: 1029-8479. DOI: 10.1007/jhep11(2019)038. [Online]. Available: http://dx.doi.org/10.1007/JHEP11(2019)038.

[21] Lantagne-Hurtubise, S. Plugge, O. Can, and M. Franz, "Diagnosing quantum chaos in many-body systems using entanglement as a resource," *Physical Review Research*, vol. 2, no. 1, 2020, ISSN: 2643-1564. DOI: 10.1103/physrevresearch.2.013254. [Online]. Available: http://dx.doi.org/10.1103/PhysRevResearch.2.013254.

[22] W. Fu and S. Sachdev, "Numerical study of fermion and boson models with infinite-range random interactions," *Physical Review B*, vol. 94, no. 3, 2016, ISSN: 2469-9969. DOI: 10.1103/physrevb.94.035135. [Online]. Available: http://dx.doi.org/10.1103/PhysRevB.94.035135.

[23] R. Feng, G. Tian, and D. Wei, *Spectrum of syk model*, 2018. arXiv: 1801.10073 [math-ph].

[24] R. Žitko, "Sneg – mathematica package for symbolic calculations with second-quantization-operator expressions," *Computer Physics Communications*, vol. 182, no. 10, pp. 2259–2264, 2011, ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2011.05.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465511001792.

[25] A. Y. Kitaev, "Unpaired majorana fermions in quantum wires," *Physics-Uspekhi*, vol. 44, no. 10S, pp. 131–136, 2001. DOI: 10.1070/1063-7869/44/10s/s29. [Online]. Available: https://doi.org/10.1070/1063-7869/44/10s/s29.

[26] G. Thompson, "Normal forms for skew-symmetric matrices and hamiltonian systems with first integrals linear in momenta," 1988.

[27] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[28] M. A. Luersen and R. Le Riche, "Globalized nelder–mead method for engineering optimization," *Computers & structures*, vol. 82, no. 23-26, pp. 2251–2260, 2004.

[29] H. Rutishauser, "The jacobi method for real symmetric matrices," *Numerische Mathematik*, vol. 9, no. 1, pp. 1–10, 1966, ISSN: 0945-3245. DOI: 10.1007/BF02165223. [Online]. Available: https://doi.org/10.1007/BF02165223.

[30] G. H. Golub and H. A. van der Vorst, "Eigenvalue computation in the 20th century," *Journal of Computational and Applied Mathematics*, vol. 123, no. 1, pp. 35–65, 2000, Numerical Analysis 2000. Vol. III: Linear Algebra, ISSN: 0377-0427. DOI: https://doi.org/10.1016/S0377-0427(00)00413-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377042700004131.

[31] N. H. Rhee and V. Hari, "On the cubic convergence of the paardekooper method," *BIT Numerical Mathematics*, vol. 35, no. 1, pp. 116–132, 1995, ISSN: 1572-9125. DOI: 10.1007/BF01732981. [Online]. Available: https://doi.org/10.1007/BF01732981.

[32] R. Babbush, D. W. Berry, and H. Neven, "Quantum simulation of the sachdev-ye-kitaev model by asymmetric qubitization," *Physical Review A*, vol. 99, no. 4, 2019, ISSN: 2469-9934. DOI: 10.1103/physreva.99.040301. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.99.040301.

[33] L. García-Álvarez, I. Egusquiza, L. Lamata, A. del Campo, J. Sonner, and E. Solano, "Digital quantum simulation of minimal ads/cft," *Physical Review Letters*, vol. 119, no. 4, 2017, ISSN: 1079-7114. DOI: 10.1103/physrevlett.119.040501. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.119.040501.

[34] Z. Luo, Y.-Z. You, J. Li, C.-M. Jian, D. Lu, C. Xu, B. Zeng, and R. Laflamme, "Quantum simulation of the non-fermi-liquid state of sachdev-ye-kitaev model," *npj Quantum Information*, vol. 5, no. 1, p. 53, 2019, ISSN: 2056-6387. DOI: 10.1038/s41534-019-0166-7. [Online]. Available: https://doi.org/10.1038/s41534-019-0166-7.

[35] L. Qi, "Eigenvalues of a real supersymmetric tensor," *Journal of Symbolic Computation*, vol. 40, no. 6, pp. 1302–1324, 2005, ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2005.05.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0747717105000817.

[36] B. Sturmfels, "Tensors and their eigenvectors," 2017. [Online]. Available: https://www.youtube.com/watch?v=khsWef3RYcs.

[37] E. Robeva, "Orthogonal decomposition of symmetric tensors," *SIAM J. Matrix Anal. Appl.*, vol. 37, no. 1, 86–102, 2015. DOI: https://doi.org/10.1137/140989340.

A Evolution of the hamiltonian under $\hat{G}$

Acting with the operator $\hat{G}(t) \equiv \hat{G}^{(k,\gamma;l,\delta)}(t) = e^{-\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t}$ on the Hilbert space of n fermionic modes translates into the Hamiltonian $\hat{\mathcal{H}}$ evolving as

$$\begin{aligned}\left(\hat{G}(t)\right)^{\dagger}\hat{\mathcal{H}}\left(\hat{G}(t)\right) &= \frac{\mathrm{i}}{2}\sum_{i,j=0}^{n-1}\sum_{\alpha,\beta=0}^{1} H_{i,\alpha;j,\beta}\; e^{\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t}\hat{c}_{i,\alpha}e^{-\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t}e^{\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t}\hat{c}_{j,\beta}e^{-\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t} = \\ &= \frac{\mathrm{i}}{2}\sum_{i,j=0}^{n-1}\sum_{\alpha,\beta=0}^{1} H_{i,\alpha;j,\beta}f_{i,\alpha}^{(k,\gamma;l,\delta)}(t)f_{j,\beta}^{(k,\gamma;l,\delta)\dagger}(t)\end{aligned} \tag{30}$$

where we have used that $\hat{G}(t)$ is unitary a, $\hat{G}^{\dagger}(t)\hat{G}(t) = \mathbb{1}$, and we have renamed $e^{\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t}\hat{c_{i,\alpha}}e^{-\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t} = f_{i,\alpha}^{(k,\gamma;l,\delta)}(t)$. The above expression can be expressed in a vectorized form as follows:

$$\hat{G}^{\dagger}\hat{\mathcal{H}}\hat{G} = \frac{\mathrm{i}}{2}\;\vec{f}^{\,\dagger}(t)H\vec{f}(t) = \frac{\mathrm{i}}{2}\vec{c}^{\,T}H'\vec{c} \tag{31}$$

where $\vec{f}(t)$ is a vector containing the elements of the new Majorana basis $\{\hat{f}_{i,\alpha}\}$ and $\vec{c}$ represents the original Majorana fermion basis $\{\hat{c}_{j,\beta}\}$.

It follows that the derivative of this operator $\{\hat{f}_{i,\alpha}\}$ with respect to time is

$$\frac{\partial}{\partial t}f_{i,\alpha}^{(k,\gamma;l,\delta)}(t) = e^{-\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t}[\hat{c}_{i,\alpha}, \hat{c}_{k,\gamma}\hat{c}_{l,\delta}]e^{\hat{c}_{k,\gamma}\hat{c}_{l,\delta}t}\;. \tag{32}$$

Using the CARs (5) one gets

$$[\hat{c}_{i,\alpha}, \hat{c}_{k,\gamma}\hat{c}_{l,\delta}] = 2(\delta_{i,l}\delta_{\alpha,\delta}\hat{c}_{k,\gamma} - \delta_{i,k}\delta_{\alpha,\gamma}\hat{c}_{l,\delta}) \tag{33}$$

where δ represents Dirac delta function. Therefore

$$\frac{\partial}{\partial t}f_{i,\alpha}^{(k,\gamma;l,\delta)}(t) = 2\left(\delta_{i,l}\delta_{\alpha,\delta}\hat{f}_{k,\gamma}^{(k,\gamma;l,\delta)}(t) - \delta_{i,k}\delta_{\alpha,\gamma}\hat{f}_{l,\delta}^{(k,\gamma;l,\delta)}(t)\right) = \sum_{r,\sigma}A_{i,\alpha;r,\sigma}^{(k,\gamma;l,\delta)}f_{r,\sigma}^{(k,\gamma;l,\delta)}(t)(t) \tag{34}$$

The above equation can be also expressed in vectorial form as:

$$\dot{\vec{f}}(t) = A^{(k,\gamma;l,\delta)}\vec{f}(t) \tag{35}$$

where $A^{(k,\gamma;l,\delta)}$ is an antisymmetric matrix with only two non zero elements, $A_{k,\gamma;l,\delta}^{(k,\gamma;l,\delta)} = -2$ and $A_{l,\delta;k,\gamma}^{(k,\gamma;l,\delta)} = 2$. Eq. (35) is an ODE that can be solved by setting the initial condition $\vec{f}(0) = \vec{c}$, and has the solution

$$\vec{f}(t) = \left(e^{A^{(k,\gamma;l,\delta)}t}\right)\vec{c}\,. \tag{36}$$

or in index form:

$$\left(\hat{G}^{(k,\gamma;l,\delta)}(t)\right)\hat{c}_{i,\alpha}\left(\hat{G}^{(k,\gamma;l,\delta)\dagger}(t)\right) = \sum_{j,\beta}\left(e_{i,\alpha;j,\beta}^{A^{(k,\gamma;l,\delta)}t}\right)\hat{c}_{j,\beta} \tag{37}$$

Coming back to equation (31), we can express the transformation induced by $\hat{G}(t)$ as the coupling matrix H transforming as

$$H' = \left(e^{-A^{(k,\gamma;l,\delta)}t}\right)H\left(e^{A^{(k,\gamma;l,\delta)}t}\right). \tag{38}$$

Since the matrix $A \equiv A^{(k,\gamma;l,\delta)}$ is a $2n \times 2n$ is antisymmetric with only two non-zero elements ± 2, A has eigenvalues $\pm 2i$ and 0. Therefore the minimum polynomial, by the Cayley-Hamilton theorem,

is $A(A^2+4\mathbb{1})=0$, which means that we can write $A^3=-4A$ and truncate the Taylor expansion at second order:

$$\begin{aligned}
e^{At}=\sum_{n=0}^{\infty}\frac{t^n}{n!}A^n &= \mathbb{1}+tA+\frac{t^2}{2!}A^2+\frac{t^3}{3!}A^3+\frac{t^4}{4!}A^4+\frac{t^5}{5!}A^5+\cdots= \\
&= \mathbb{1}+tA+\frac{t^2}{2!}A^2+\frac{t^3}{3!}(-4)A+\frac{t^4}{4!}(-4)A^2+\frac{t^5}{5!}(-4)^2A+\cdots= \\
&= \mathbb{1}+A\left(t+\frac{t^3}{3!}(-4)+\frac{t^5}{5!}(-4)^2+\ldots\right)+A^2\left(\frac{t^2}{2!}+\frac{t^4}{4!}(-4)A^2+\ldots\right)= \\
&= \mathbb{1}+A\sum_{n=0}^{\infty}(-4)^n\frac{t^{2n+1}}{(2n+1)!}+A^2(-\frac{1}{4})\sum_{n=1}^{\infty}(-4)^n\frac{t^{2n}}{(2n)!}= \\
&= \mathbb{1}+A\frac{1}{2}\sum_{n=0}^{\infty}(-1)^n\frac{(2t)^{2n+1}}{(2n+1)!}-A^2\frac{1}{4}\sum_{n=1}^{\infty}(-1)^n\frac{(2t)^{2n}}{(2n)!}= \\
&= \mathbb{1}+\frac{\sin 2t}{2}A-\frac{\cos 2t-1}{4}A^2
\end{aligned} \tag{39}$$

We now rename $e^{A^{(k,\gamma;l,\delta)}t}=G^{(k,\gamma;l,\delta)}(t)$ explicitly note that this matrix represents a Givens rotation. Indeed, in matrix notation we can write

$$G^{(k,\gamma;l,\delta)}(t)=\begin{bmatrix}
1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & & \vdots & & \vdots \\
0 & \cdots & \cos(2t) & \cdots & \sin(2t) & \cdots & 0 \\
\vdots & & \vdots & \ddots & \vdots & & \vdots \\
0 & \cdots & -\sin(2t) & \cdots & \cos(2t) & \cdots & 0 \\
\vdots & & \vdots & & \vdots & \ddots & \vdots \\
0 & \ldots & 0 & \ldots & 0 & \cdots & 1
\end{bmatrix}, \tag{40}$$

where the diagonal elements are all 1 except from $G^{(k,\gamma;l,\delta)}_{k,\gamma;k,\gamma}=G^{(k,\gamma;l,\delta)}_{l,\delta;l,\delta}=\cos(2t)$ and the off-diagonal elements are zero except from $G^{(k,\gamma;l,\delta)}_{k,\gamma;l,\delta}=-G^{(k,\gamma;l,\delta)}_{l,\delta;k,\gamma}=\sin(2t)$. As explained in the main text, this is a rotation matrix known as Givens rotation.

B Angle formulas for Paardekooper algorithm

Here we write down the angle formulas for the Jacobi annihilator in Paardekooper algorithm, as explained in [31]. Using simplified notation

$$H_{pp}=\begin{bmatrix}0 & \alpha\\ -\alpha & 0\end{bmatrix},\quad H_{pq}=\begin{bmatrix}x & v\\ w & y\end{bmatrix},\quad H_{qq}=\begin{bmatrix}0 & \beta\\ -\beta & 0\end{bmatrix}$$

the angle formulas are for $t_1\in(-\pi/4,\pi/4]$

$$\tan 2t_1=\frac{2(\alpha w-\beta v)}{\alpha^2-\beta^2+v^2-w^2}$$

and for $t_2\in(-\pi/2,\pi/2]$

$$\tan t_2=\begin{cases}-\frac{c_1v-s_1\beta}{c_1\alpha+s_1w}, & c_1\alpha+s_1w\neq 0\\ -\frac{c_1w-s_1\alpha}{c_1\beta+s_1v}, & c_1\beta+s_1v\neq 0\end{cases}$$

Here $c_1 = \cos t_1$ and $s_1 = \sin t_1$. The first two angles have been chosen to annihilate v and w. Note that x and y have not changed. Suppose that H_{pp}, H_{pq} and H_{qq} are transformed by the first two rotations into

$$\widetilde{H}_{pp} = \begin{bmatrix} 0 & \tilde{\alpha} \\ -\tilde{\alpha} & 0 \end{bmatrix}, \quad \widehat{H}_{pq} = \begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix}, \quad \widetilde{H}_{qq} = \begin{bmatrix} 0 & \widetilde{\beta} \\ -\tilde{\beta} & 0 \end{bmatrix}.$$

The angles $t_3 \in (-\pi/4, \pi/4]$ and $t_4 \in (-\pi/2, \pi/2]$ are computed from the requirement that x and y become zero, which yields

$$\tan 2t_3 = \frac{2(\tilde{\alpha}y + \widetilde{\beta}x)}{\widetilde{\alpha}^2 - \widetilde{\beta}^2 + x^2 - y^2}$$
$$\tan t_4 = \begin{cases} -\frac{c_3 x + s_3 \widetilde{\beta}}{c_3 \widetilde{\alpha} + s_3 y}, & c_3\widetilde{\alpha} + s_3 y \neq 0 \\ \frac{c_3 y - s_3 \widetilde{\alpha}}{c_3 \widetilde{\beta} - s_3 x}, & c_3\widetilde{\beta} - s_3 x \neq 0 \end{cases}$$

where $c_3 = \cos t_3$ and $s_3 = \sin t_3$.

C Numerical calculation of the energy

Throughout this work, the numerical calculations were done in the second quantized formalism. Here we explain how we computed the energy of the free fermion Hamiltonian (3) and the SYK Hamiltonian (26). These energies are always taken with respect to the fermionic vacuum $|\Omega\rangle$ with respect to the original Hamiltonian configuration.

C.1 Vacuum energy of free fermions

The energy in the free-fermion case is

$$E = \langle\Omega|\hat{\mathcal{H}}|\Omega\rangle \ , \tag{41}$$

with $\hat{\mathcal{H}}$ given by

$$\hat{\mathcal{H}} = \frac{\mathrm{i}}{2} \sum_{i,j=0}^{n-1} \sum_{\alpha,\beta=0}^{1} H_{i,\alpha;j,\beta} \hat{c}_{i,\alpha} \hat{c}_{j,\beta} \ . \tag{42}$$

By using that a Majorana operator can be expressed as

$$\hat{c}_{i,\alpha} = \mathrm{i}^{\alpha}(\hat{a}_i + (-1)^{\alpha}\hat{a}_i^{\dagger}) \tag{43}$$

in terms of Dirac operators, where i is the imaginary unit, and that Dirac operators act on the number state basis as $a_k|\Omega\rangle = 0$ and $a_k^{\dagger}|\Omega\rangle = |1_k\rangle$, we can write

$$\hat{c}_{l,\alpha}\hat{c}_{k,\beta}|\Omega\rangle = \begin{cases} \mathrm{i}^{\alpha+\beta}(-1)^{\beta}|\Omega\rangle & \text{if } l = k \\ \mathrm{i}^{\alpha+\beta}(-1)^{\alpha+\beta}\hat{a}_l^{\dagger}\hat{a}_k^{\dagger}|\Omega\rangle & \text{if } l \neq k \end{cases} \tag{44}$$

When applying $\langle\Omega|$ from the left, only the terms with $l = k$ survive and we have:

$$E = \langle\Omega|\hat{\mathcal{H}}|\Omega\rangle = \frac{\mathrm{i}}{2} \sum_{l,\alpha;k,\beta} H_{l,\alpha;k,\beta} \langle\Omega|\hat{c}_{l\alpha}\hat{c}_{k\beta}|\Omega\rangle = \frac{\mathrm{i}}{2} \sum_{l,\alpha,\beta} H_{l,\alpha;l,\beta} \cdot \mathrm{i}^{\alpha+\beta}(-1)^{\beta} \tag{45}$$

i.e., sum over the elements of the middle diagonal blocks of H. Using that H is antisymmetric, $H_{l,\alpha;k,\beta} = -H_{k,\beta;l,\alpha}$ so:

$$H_{l,\alpha;l,\beta} = -H_{l,\beta;l,\alpha} \tag{46}$$

Then the diagonal terms with $\alpha = \beta$, $H_{l,0;l,0}$ and $H_{l,1;l,1}$, vanish, and we only have left:

$$E = \frac{1}{2} \sum_{l} (H_{l,0;l,1} - H_{l,1;l,0}) = -\sum_{l} H_{l,1;l,0} \tag{47}$$

which is the sum of the bottom-left terms in each of the diagonal (2×2) blocks of the H matrix.

C.2 Vacuum energy of the SYK

For the SYK Hamiltonian the derivation is analogue to the free fermions:

$$E_0 = \langle\Omega|\hat{\mathcal{H}}_{SYK}|\Omega\rangle = \sum_{\mathbf{i}_1,\mathbf{i}_2,\mathbf{i}_3,\mathbf{i}_4=0}^{2n-1} J_{\mathbf{i}_1,\mathbf{i}_2,\mathbf{i}_3,\mathbf{i}_4}\langle\Omega|\hat{c}_{\mathbf{i}_1}\hat{c}_{\mathbf{i}_2}\hat{c}_{\mathbf{i}_3}\hat{c}_{\mathbf{i}_4}\Omega\rangle, \tag{48}$$

where

$$\begin{aligned}\langle\Omega|\hat{c}_{\mathbf{i}_1}\hat{c}_{\mathbf{i}_2}\hat{c}_{\mathbf{i}_3}\hat{c}_{\mathbf{i}_4}|\Omega\rangle = \\ &= \langle\Omega|\hat{c}_{i_1,\alpha_1}\hat{c}_{i_2,\alpha_2}\hat{c}_{i_3,\alpha_3}\hat{c}_{i_4,\alpha_4}|\Omega\rangle \\ &= \mathrm{i}^{\alpha_1+\alpha_2+\alpha_3+\alpha_4}\left((-1)^{\alpha_3+\alpha_4}(\delta_{i_1,i_4}\delta_{i_2,i_3}-\delta_{i_1,i_4}\delta_{i_2,i_3})+(-1)^{\alpha_2+\alpha_4}\delta_{i_1,i_2}\delta_{i_3,i_4}\right).\end{aligned} \tag{49}$$