Simulation of the Fermi-Hubbard model with shortcuts to adiabaticity

Irene Agudo Zamora*
Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.

Advisor: Axel Pérez-Obiol
Tutor: Bruno Juliá

Abstract: This thesis investigates the efficiency of the adiabatic evolution, in the context of a 1-dimensional Fermi-Hubbard system. Two equivalent methods were used: CNOTs and FSWAPs. As it is easier to write a generalised code for CNOTs, it was the method used in subsequent sections. The sources of error in quantum simulations were studied (the adiabatic and Trotteritzation errors), and it was verified that they behaved as predicted (in particular, the adiabatic error increased with a small N, and the Trotteritzation error increased with a larger δt). The optimal parameters needed to achieve the desired infidelity were also determined. Shortcuts to adiabaticity using the Counter-Diabatic Driving method are also applied, resulting in a faster evolution.

I. INTRODUCTION

Quantum computing has emerged in recent years as a promise to carry out operations that a classical machine could not. Even so, it is still limited for very large systems. This has given rise to algorithms such as the adiabatic algorithm [2], which aims to obtain solutions for complex systems from known solutions for simpler systems. These adiabatic processes are powerful but slow, which has led researchers to look for faster methods, such as shortcuts to adiabaticity [7]. This thesis explores adiabatic evolution and shortcuts to adiabaticity, using the Counter-Diabatic Driving [1] method, for a 1-dimensional Fermi-Hubbard system. Systems to up to 10 sites (Ns=10) were studied. To perform the evolutions, the software used was OpenFermion and Qibo. OpenFermion is a Python library for simulating fermionic systems, which play an important role in chemistry and materials science research. Qibo, on the other hand, is a quantum circuit simulator. The initial ground state of the evolution and the operators need to be converted into quantum gates. In the case of the kinetic term, this is achieved by using two different procedures named here FSWAPs and CNOTs.

The thesis is organized as follows: Sections II and III cover the theoretical foundations: the Fermi-Hubbard model and Jordan-Wigner transformation. Section IV presents the different algorithms used, as well as the gates necessary to perform adiabatic evolution in a quantum computer, using FSWAPs and CNOTs. Section IV D briefly introduces the shortcuts to the adiabaticity (STA) method with Counter-Diabatic Driving (CD). In Section V, the results of the simulations are presented, and the effectiveness of the adiabatic evolution and the STA methods are studied. There is also an analysis of the different sources of error. Finally, in Section VI the conclusions are presented, proposing future work.

II. FERMI-HUBBARD MODEL

The Fermi-Hubbard model describes the behaviour of electrons in a solid. This is done by considering their movement between sites (hopping term) and electron repulsion (interaction term). It is based on the tight-binding electron model and adds a short-range interaction. Electronic systems are described by Hamiltonians consisting of the fermionic operators in the second quantization formalism. These are $a_0^\dagger$, the creation operator, which creates a particle at site 0, and a_0, the annihilation operator, also known as raising and lowering operators. In this formalism, the number operator ($n_j = a_j^\dagger a_j$) is also relevant. Its eigenvalues are the number of fermions in a particular state associated with the lattice site j. The Fermi-Hubbard Hamiltonian is [5]:

$$H = -J \sum_{j=1}^{Ns-1} \sum_{\sigma \in \{\uparrow,\downarrow\}} a_{j,\sigma}^\dagger a_{j+1,\sigma} + h.c. + U \sum_j n_{j\uparrow} n_{j\downarrow}, \tag{1}$$

where σ indicates spin up or spin down, and Ns is the total number of sites. The first term is the hopping term, which describes particles tunnelling between neighbouring sites (h.c stands for hermitian conjugate). The last term is the on-site interaction, which acts between fermions occupying the same site (in the case of electrons, it's the Coulomb interaction). Periodic boundary conditions are assumed.

III. JORDAN-WIGNER TRANSFORMATION

Quantum computers (QC) are made up of qubits, so to be able to work in a QC, one has to transform Hamiltonians (like the one introduced in section II) into the qubit operators $X_j, Y_j, Z_j, ...$ to treat them.

For fermions, one possible mapping was proposed by Jordan and Wigner in 1928. The Jordan-Wigner transformation is particularly significant in the study of quantum many-body systems and quantum simulations. It

[*] Electronic address: ireneagudozamora@gmail.com

provides a way to map fermionic operators to spin operators (qubits), which is crucial for implementing fermionic Hamiltonians on quantum computers. As described in OpenFermion:

$$a_j = \prod_{k=1}^{j-1} Z_k \frac{1}{2}(X_j + iY_j). \tag{2}$$

IV. METHODOLOGY

A. Adiabatic Algorithm

The adiabatic evolution (AE) is an algorithm used to find the ground state of a Hamiltonian that would be hard to find analytically. It evolves from the ground state of an initial Hamiltonian, which is easy to construct, to the ground state of the target Hamiltonian. The evolution time must be long enough to satisfy the conditions of the adiabatic theorem. There can be level crossings as long as some symmetry protects them. The time-dependent Hamiltonian is of the form:

$$H(s) = (1-s)H_i + sH_f, \tag{3}$$

where s is the effective time: $s = t/T \Rightarrow \in [0,1]$, with T the period and t the current state of the evolution $t = j\delta t$ (δt is the time step and j is the number of steps made).

To run a simulation on a quantum computer, it is important to map the sites of the Hamiltonian into qubits. A 1D chain can be mapped by representing the particles with spin down in the odd qubits, and the spin up in the even.

Having the Hamiltonian in equation 3, the evolution consists on evolving the Hamiltonian from U=0 to U=1, this means:

$$H_i = H_{hopping}, \tag{4}$$

$$H_f = H_{inter} + H_{hopping}, \tag{5}$$

and so

$$H(s) = H_{hopping} + s \cdot H_{inter} \ . \tag{6}$$

Notice that the initial Hamiltonian chosen is the kinetic term of the Fermi-Hubbard Hamiltonian. So the initial state is the ground state of $H_{hopping}$, which in the cases studied can be found easily. In a QC, the initial state must be expressed with a circuit. In this thesis, the function prepare_gaussian_state from OpenFermion was used. This function prepares the eigenstate of a quadratic Hamiltonian using the algorithm described in [3].

The solution of the Time-dependent Schrödinger equation is

$$|\psi(T)\rangle = \hat{\mathcal{T}} e^{-i\int_0^T H_{adiab}(t)dt} |\psi(0)\rangle. \tag{7}$$

The exponential of the integral should be transformed into a product to be able to compute it:

$$\psi(T)\rangle \approx \prod_{j=0}^{N} e^{-iH_{adiab}(jt)\delta t} |\psi(0)\rangle. \tag{8}$$

To perform the evolution in a QC, it is key to remember that they only manage up to 2 qubit operators, so we need to decompose the previous operator into a simpler one. To do so, we can use the Trotter-Suzuki formula (valid for small δt):

$$\begin{aligned} e^{-i\delta t H_{adiab}(j\delta t)} &= e^{-i\delta t \frac{j\cdot\delta t}{T} H_C} e^{-i\delta t H_{hopp}} \\ &\quad + O([H_{hopp}, H_C], \delta t^2). \end{aligned} \tag{9}$$

And so the expression for the evolved state turns to:

$$\begin{aligned} |\psi(T)\rangle \approx & \prod_{j=0}^{N} \prod_{i}^{L} e^{-i\delta t \frac{j\delta t}{T} U \hat{n}_{i\uparrow}\hat{n}_{i\downarrow}} \\ & \prod_{i\sigma}^{L} e^{-i\delta t J\left(\hat{a}^\dagger_{i+1\sigma}\hat{a}_{i\sigma} + \hat{a}^\dagger_{i\sigma}\hat{a}_{i+1\sigma}\right)} |\psi(0)\rangle. \end{aligned} \tag{10}$$

Now, it is needed to define the gates used to express H_C and H_{hopp} in the circuit.

For the interaction term $\delta t^2 \frac{j}{T} \cdot U \cdot \hat{n}_{i\uparrow}\hat{n}_{i\downarrow} = \delta t^2 \frac{j}{T} h_C$, where

$$h_C = U \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{11}$$

δt, T and U (corresponding to the final H) are constant during the evolution but j changes. The evolution operator is then:

$$e^{-i\delta t^2 \frac{j}{T}\cdot h_C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-i\delta t^2 \frac{j}{T}\cdot U} \end{pmatrix}, \tag{12}$$

which is implemented as the CU1 gate in Qibo. The Hopping term adds an interaction between non-adjacent qubits, so it can't be built with only one operator. There are two equivalent procedures: FSWAPs and CNOTs.

B. FSWAPs

The expression of the hopping operator is:

$$h_h = -J(\hat{a}^\dagger_{i+1}\hat{a}_i + \hat{a}^\dagger_i\hat{a}_{i+1}) = -J \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{13}$$

where the matrix is in the qubit space as long as i and $i+1$ correspond to adjacent qubits. The evolution operator is then:

$$e^{-i\delta t\cdot h_h} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(J\delta t) & i\sin(J\delta t) & 0 \\ 0 & i\sin(J\delta t) & \cos(J\delta t) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{14}$$

which corresponds to the fSim gate in Qibo, with an angle of $J\delta t$.

In general, the hopping term adds an interaction between non-adjacent qubits, so it is necessary to bring them together. The particles are fermions, so to switch them, the commutation relations need to be applied. The fermionic swap gates have that precise function [4]:

$$f^{p,q}_{swap} = 1 + a^\dagger_p a_q + a^\dagger_q a_p - a^\dagger_p a_p - a^\dagger_q a_q, \tag{15}$$

$$f^{p,q}_{swap} a^\dagger_p (f^{p,q}_{swap})^\dagger = a^\dagger_q \quad f^{p,q}_{swap} a_p (f^{p,q}_{swap})^\dagger = a_q. \tag{16}$$

C. CNOTs

Another method that could be used is to map the exponent to qubit operators and exponentiate each Pauli string with the staircase method:

The operator $e^{i\theta Z}$, can be written in Qibo as $RZ = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$. With several Z gates, for example $e^{i\theta Z_0 Z_1}$, the resulting circuit is shown in Figure 1:

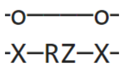

FIG. 1: Circuit for $e^{i\theta Z_0 Z_1}$

If the gates X and Y are included, the result is something similar to $e^{\theta iYZY}$ or $e^{\theta iXZX}$. To use the staircase method, the gates X and Y can be expressed in terms of the Z gate with a change of basis (X=HZH† and Y=RX($\pi/2$) Z RX($\pi/2$)†). The circuit for $e^{\theta Y_1 Z_2 Y_3}$ is shown in Figure 2.

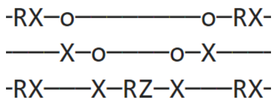

FIG. 2: Circuit for $e^{\theta Y_1 Z_2 Y_3}$

The construction of the circuit of the kinetic term (for any Ns) is done with a single generalized code that creates a circuit for any operator of the form $e^{\theta H}$, where H must be under a Jordan-Wigner transformation. This generalised code contrasts with the FSWAPs procedure, in which the circuit was designed for every value of Ns. For this reason, the further coding (shortcuts to adiabaticity) was coded using CNOTs.

D. Shortcuts to Adiabaticity with Counter-Diabatic Driving

Adiabatic processes can be too slow for real-life applications and might imply circuits with large amounts of gates. That is where Shortcuts to Adiabaticity come into play. STA achieves the same final states as those obtained through adiabatic processes, in a much shorter time. The specific STA method CD, introduces an auxiliary Hamiltonian designed to suppress non-adiabatic transitions, to achieve rapid and controlled evolution of quantum states.

Consider the time-dependent Hamiltonian $H_0(t)$ with instantaneous eigenvalues $E_n(t)$ and eigenstates $|n(t)\rangle$ such that:

$$H_0(t)|n(t)\rangle = E_n(t)|n(t)\rangle. \tag{17}$$

In a truly adiabatic evolution (as long as the evolution is slow), the system remains in an instantaneous eigenstate $|n(t)\rangle$ of $H_0(t)$. However, for small values of T, non-adiabatic transitions can occur, leading to results different from $|n(t)\rangle$. In the CD method, a term is added to the Hamiltonian $H_{CD}(t)$, such that the total Hamiltonian

$$H_{total}(t) = H_0(t) + H_{CD}(t), \tag{18}$$

leads to the desired final state even for fast processes.

The counter-diabatic term $H_{CD}(t)$ is given by:

$$H_{CD} = \dot{s}A_s. \tag{19}$$

This auxiliary term (A_s) cannot be calculated exactly in many practical cases, limiting its applicability in complex systems. Luckily, this term can be obtained with the following relation:

$$A_s(l) = i\sum_{k=1}^{l} \alpha_k(t)[H,[H,[\cdots[H,\partial_s H]]]], \tag{20}$$

where $l \to \infty$. A_s can be approximated by $ia_k[H,\partial_s H] = ia_k(H_h H_c - H_c H_h)$.

The coefficient a_k is the one that minimizes $S = Tr(G^2)$, where $G = \delta_s H - i[H, A_s]$.

V. RESULTS

In this section there will be an analysis of the errors related to T and N, to understand the chosen values of N and T. The result of the study of the adiabatic evolution and the STA method leads to Figure 3, where the evolution (with AE and STA) of the infidelity for a big system (Ns=10) is shown.

In Table II, the second column corresponds to the values for which a normalized infidelity of 0.05 is reached. The normalized infidelity is $I_N = I/I_0$, where I_0 is the

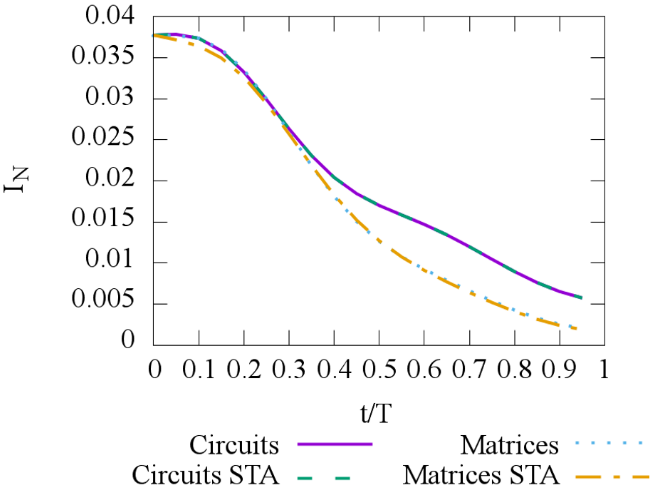

FIG. 3: Comparison of infidelity (I_N) for a single evolution with N=20, T=2 and Ns=10. With matrix calculus and circuits simulations, with and without STA.

$\mathbf{N_s}$	Circuit	N_q	Depth
3	CNOTS	67	94
	FSWAPS	21	10
4	CNOTS	100	134
	FSWAPS	32	11
5	CNOTS	133	174
	FSWAPS	47	15
6	CNOTS	166	214
	FSWAPS	58	15

TABLE I: Number of two-qubit gates N_q and circuit depths for different lattice sizes and the CNOT and FSWAP procedures.

$\mathbf{N_s}$	Best value of I_N			Values for $I_N \approx 0.5$		
	N	T	I_N	N	T	I_N
3	120	2	$2.6 \cdot 10^{-5}$	20	6.8	0.059
4	120	3	$4.36 \cdot 10^{-5}$	20	4.6	0.050
5	120	2.8	$7.8 \cdot 10^{-5}$	20	6.4	0.056
6	120	3.6	$2.7 \cdot 10^{-4}$	20	7	0.057

TABLE II: Study of the values of T and N with CNOTs.

infidelity of the prepared initial state with the final state. For Ns=4, the ground state is degenerated, so the adiabatic evolution gave bad results. To solve this and make the evolution, the ground state of a Hamiltonian with $U \to 0$ can be used. The results from Table II and Figure 4, indicate that the best results are for larger values of T, and small values of $T/N = \delta t$. This is due to two sources of error: the adiabatic error, and the Trotterization error. Adiabatic errors are reduced for larger periods, and the Trotteritzation error is reduced for a smaller δt. As the number of steps N is fixed, increasing T will increase δt leading to a bigger Trotteritzation error. That is why the worst results appear for small N and big T.

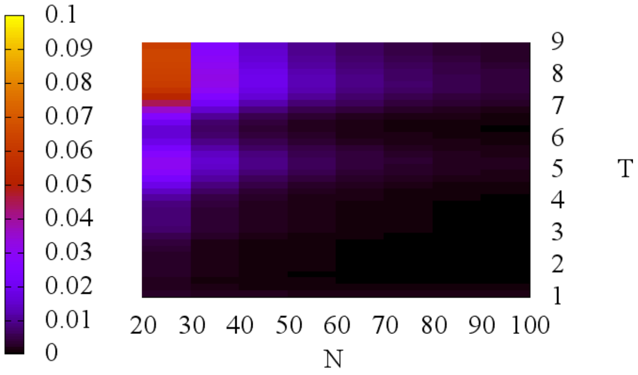

FIG. 4: Density plot of the infidelity (I_N) for Ns=6 with CNOTS for different values of T and N.

With this in mind, STA was coded and simulated for a larger system (compared with the ones studied in the table), Ns=7, using the determined values to reach a small infidelity: $N \geq 20$ and a small T. To study the advantage of the STA algorithm, in Figure 5 the difference in the infidelity can be seen ($I_{adiabatic} - I_{STA}$). For big N and T (but still small δt) the two methods lead to similar results, as expected. On the other hand, STA gives better results for small N and T, where the standard adiabatic evolution is not that efficient. In [6] it is proposed to only use the H_{CD} term added to the Hamiltonian, a method included in Figure 6, where the comparison of all the methods can be seen. It is quite surprising that for small N the results with only the CD term are better than with and without STA. However, those two converge fast so this advantage is soon lost.

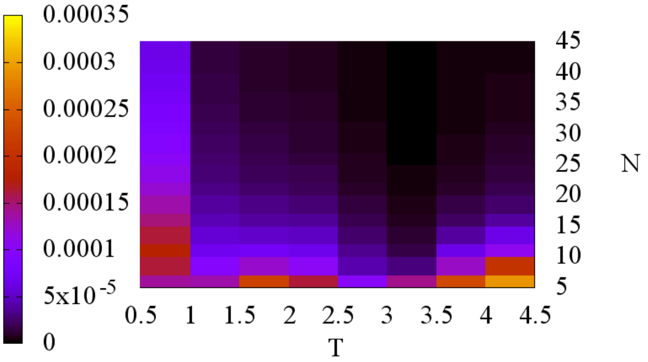

FIG. 5: Difference in the infidelity ($I_{adiabatic} - I_{STA}$) for different values of N and T.

Finally, it would be interesting to perform a single evolution for the largest system possible, to see if the previous conclusions about the parameters and the behaviour of the different methods, allow us to reach an acceptable value of the infidelity. The larger system that could be

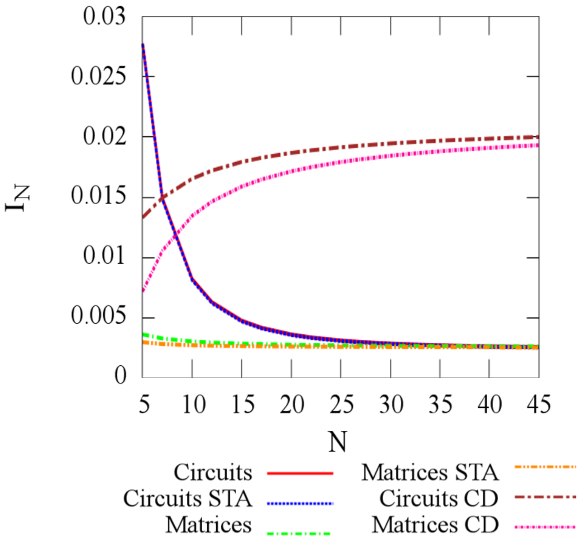

FIG. 6: Infidelity comparison of different methods for Ns=7 and different values of N (with $T = 1$).

computed analytically, to be able to compare results, was Ns=10. From Table II, it is deduced that $N = 20$ and $T = 2$ should be enough to get an infidelity smaller than 0.05. The behaviour for the different methods can be anticipated: for $N = 20$, only the CD term will not give good results (therefore it is not computed), and STA will be very similar for matrices and circuits as due to small δt and large N. The result of the evolution is Figure 3, which gives a final infidelity of 0.005 (better than 0.05, as expected), and the expected behaviours for each method.

VI. CONCLUSIONS

This thesis investigated the adiabatic evolution and shortcuts to adiabaticity, particularly Counter-Diabatic Driving in the context of quantum computing. Key takeaways include: The convergence behaviour of quantum circuits was evaluated, revealing optimal parameters for achieving desired infidelity levels. For instance (considering $N \geq 20$), $1 < T < 6$ is small enough to make $\delta t = T/N$ small, and large enough to reduce the error in the adiabatic evolution. As for the equivalent methods CNOTs and FSWAPs, Table I shows that the circuit depth and the number of two-qubit gates are smaller for FSWAPs. However, this is because for CNOTs the evolution is decomposed in simpler gates than FSWAPs. In a real QC (depending on the native gates used) FSWAPs should be decomposed into simpler gates and the advantadge could be lost. In addition, when it comes to programming, it is easier to make a generalized code for CNOTs, and therefore it was the method used for programming STA. The results obtained for the infidelity for the systems studied (up to 10 sites) are small, so all methods can give good results. STA doesn't show a remarkable improvement; however, this could be due to the simplicity of the system (1-dimensional) and the low number of sites. A study with larger systems and more complex systems could be done to determine the improvement of STA in real-life systems. Notably, evolution with only the CD term appears to yield better results for small N compared to using STA or no STA. There could be further studies of this phenomenon.

Acknowledgments

I would like to express my gratitude to my advisor Axel Pérez-Obiol for his patience and dedication. I would also like to thank Bruno Julià for giving me the opportunity to meet the members of the Barcelona Supercomputing Center Quantum group, especially Joan Triadu for his help.

[1] Shuoming An, Dingshun Lv, Adolfo del Campo, and Kihwan Kim. Shortcuts to adiabaticity by counterdiabatic driving for trapped-ion displacement in phase space. *Nature Communications*, 7(1), September 2016.

[2] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000.

[3] Zhang Jiang, Kevin J. Sung, Kostyantyn Kechedzhi, Vadim N. Smelyanskiy, and Sergio Boixo. Quantum algorithms to simulate many-body physics of correlated fermions. *Physical Review Applied*, 9(4), April 2018.

[4] Ian D. Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Alán Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush. Quantum simulation of electronic structure with linear depth and connectivity. *Physical Review Letters*, 120(11), March 2018.

[5] Mingpu Qin, Thomas Schäfer, Sabine Andergassen, Philippe Corboz, and Emanuel Gull. The hubbard model: A computational perspective. *Annual Review of Condensed Matter Physics*, 13(1):275–302, March 2022.

[6] Jialiang Tang, Ruoqian Xu, Yongcheng Ding, Xusheng Xu, Yue Ban, Manhong Yung, Axel Pérez-Obiol, Gloria Platero, and Xi Chen. Exploring ground states of fermi-hubbard model on honeycomb lattices with counterdiabaticity, 2024.

[7] Erik Torrontegui, Sara Ibáñez, Sofia Martínez-Garaot, Michele Modugno, Adolfo del Campo, David Guéry-Odelin, Andreas Ruschhaupt, Xi Chen, and Juan Gonzalo Muga. *Shortcuts to Adiabaticity*, page 117–169. Elsevier, 2013.