

國立臺灣大學理學院物理學系

碩士論文

Department of Physics

College of Science

National Taiwan University

Master Thesis



張量網路演算法對 Thirring 模型在一維無限長格點之研究  
Tensor Network Studies of Thirring Model on a One-dimensional  
Infinite-size Lattice

洪浩迪

Hao-Ti Hung

指導教授：高英哲博士

Advisor: Ying-Jer Kao, Ph.D.

中華民國 107 年 6 月

June, 2018

# 口試委員會審定書

張量網路演算法對 Thirring 模型在一維無限長格  
點之研究

Tensor Network Studies of Thirring Model on  
one-dimensional Infinite-size Lattice

本論文係洪浩迪君 (R05222034) 在國立臺灣大學物理學研究所  
完成之碩士學位論文，於民國 107 年 6 月 15 日承下列考試委員審  
查通過及口試及格，特此證明

口試委員：

陳柏中

---

高莫如

---

林俊仁

---

---

---

---



## 致謝

碩士班兩年，回想起來真的很快。首先我要感謝我的指導教授高英哲老師兩年來的指導，想當初進來時我說我想要做演算法應用在物理模型上，果然真的讓我做了演算法，很幸運的，雖然一路上碰到一大堆的問題，許許多多問題在我們討論之後都被一一解決了，幾乎可以說是順利的完成演算法。模型的部分，非常感謝林及仁教授在場論模型這方面的指導。實驗室的許多學長姊都非常的熱心，我要感謝林育平學長，在我剛進來時我幾乎全部的問題都問他，他都抽空幫我解決，如果不是他的話我學習應該沒那麼順利。再來是周昀萱學長也常常在程式方面教我許多技巧，讓我可以加速完成我的演算法，李致遠學長、吳愷訢和趙凱文也常常幫我解決許多程式方面的問題，易德學長也在我剛進來的時候給了不少的幫助，以及博士後 Adam 在英文方面的幫助，博士後 Adil 也會給我一些研究方面的建議，還要感謝實驗室許許多多的夥伴包含曾郁欽學長、高文瀚學長、鄭君筑學姊、鍾瑞輝和顏敬哲在平常的閒聊，除了我們的團隊外我也要也感謝譚道璘以及羅中佑學長給我許許多多的幫助，當然還有爸媽一路上的支持。最後一定要感謝我的大學老師孫士傑老師的引薦。很開心可以來到這個研究室做研究，讓我在數值方面的技巧更進一步。



## 中文摘要

我們利用張量網路演算法研究 Thirring 模型。我們將模型離散化後，找出 Thirring 模型哈密頓的自旋算符表示法並用矩陣作用算符表示。

利用均勻矩陣乘積態的變分優化演算法去找出模型的基態解並調查其相圖。然後利用時間相依變分原理來研究 Thirring 模型的動態演化，特別是對於跨相變的動態演化特別有興趣。

關鍵字: 張量網路、矩陣乘積態、均勻矩陣乘積態的變分優化演算法、時間相依變分原理、Thirring 模型、量子演化、動態相變





# Abstract

We use tensor networks to study the Thirring model. We discretize the model onto the lattice, find the spin representation for the Hamiltonian of the Thirring model and use the matrix product operator (MPO) to represent it.

Using the variational optimization algorithms for uniform Matrix Product State (VUMPS), we find the ground state of the model and investigate the phase diagram. Then, we use the time-dependent variational principle algorithm (TDVP) to study the quench dynamics for the Thirring model, especially for what happens when quenching different phases.

**Keywords:** Tensor network (TN), matrix product state (MPS), variational optimization algorithm for uniform matrix product state (VUMPS), time-dependent variational principle (TDVP), Thirring model, quantum quench, dynamical phase transition (DPT)



# Contents

口試委員會審定書	i
致謝	ii
中文摘要	iii
Abstract	iv
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
<b>2 Thirring Model</b>	<b>3</b>
2.1 Spin Representation of the Thirring Model . . . . .	3
2.2 Chiral Condensate . . . . .	5
2.3 Mapping to the Classical 2D XY Model . . . . .	5
<b>3 Tensor Network and Matrix Product state</b>	<b>6</b>
3.1 Tensor Network and Tensor Diagram . . . . .	6
3.2 Matrix Product States . . . . .	7
3.3 Uniform Matrix Product States . . . . .	10
3.4 Expectation Values . . . . .	11
3.5 Gauge Degrees of Freedom, Canonical Form and Symmetric gauge . .	12
3.6 Geometric Series for Transfer Matrix . . . . .	16
3.7 Matrix Product Operator . . . . .	18
<b>4 Variational Optimization Method for uniform Matrix Product State</b>	<b>20</b>
4.1 Effective Hamiltonian . . . . .	20
4.2 VUMPS algorithm . . . . .	24
<b>5 Time-Dependent Variational Principle Applied to Matrix Product State</b>	<b>27</b>
5.1 Tangent Vector Space . . . . .	27
5.2 Gauge Fixing for Tangent Vector . . . . .	28
5.3 Projection Operator . . . . .	29
5.4 TDVP algorithm . . . . .	32
<b>6 Result and Conclusion</b>	<b>34</b>
6.1 Ground State of the Thirring Model . . . . .	34
6.2 TDVP Result . . . . .	40

**7 Summary**

**Bibliography**

**Appendix**

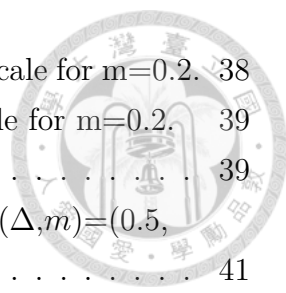
**More Numerical Results**





# List of Figures

3.1	Tensor network diagrams (a) a scalar, (b) a vector, (c) a matrix and (d) a rank-3 tensor. . . . .	7
3.2	Tensor network diagram for Eqs. (3.1) and (3.2). . . . .	7
3.3	(a) Graphical representation of an iterative construction of left-canonical MPS from arbitrary quantum state by SVD. (b) Graphical representation of an iterative construction of right-canonical MPS from arbitrary quantum state by SVD. (c) The property of left canonical form. (d) The property of right canonical form. (e) Tensor network diagram of mixed canonical form. . . . .	9
3.4	Expectation value represented as a tensor network diagram. . . . .	11
3.5	(a) Use SVD to decompose tensor $A$ . (b) $(\Lambda, \Gamma)$ notation for uMPS. . . . .	12
3.6	Check the property of the left canonical form. . . . .	14
4.1	(a) One-site effective Hamiltonian, (b) zero-site effective Hamiltonian (c) acting with tensor $A_C$ on one-site effective Hamiltonian (d) acting with tensor $C$ on zero-site effective Hamiltonian. . . . .	21
4.2	(a) one-site effective Hamiltonian (b) zero-site effective Hamiltonian . . . . .	24
4.3	Flow chart of VUMPS algorithm. . . . .	26
5.1	An illustration of the uMPS manifold and tangent space. The black dot represents the uMPS, $\psi(A)$ , and the tangent vector $\Phi(B; A)$ is a vector line on the tangent plane. . . . .	28
5.2	(a) The properties of $V_L$ (b) The properties of tensor $B$ . . . . .	29
6.1	Energy density of the Thirring model. . . . .	34
6.2	Entanglement entropy of the Thirring model. . . . .	35
6.3	Chiral Condensate of the Thirring model. . . . .	35
6.4	Fermion correlator for the massless case on a linear scale. . . . .	36
6.5	Fermion correlator for the massless case with a semi-log scale. . . . .	37
6.6	Fermion correlator for the massless case on a log-log scale. . . . .	37
6.7	Fermion correlator for the massive case on a linear scale for $m=0.2$ . . . . .	38



6.8 Fermion correlator for the massive case with a semi-log scale for  $m=0.2$ . 38

6.9 Fermion correlator for the massive case on a log-log scale for  $m=0.2$ . 39

6.10  $\Delta(g)$  near the transition. . . . . 39

6.11 The Thirring model evolving from  $(\Delta,m)=(-0.8, 0.2)$  to  $(\Delta,m)=(0.5, 0.2)$ . . . . . 41

6.12 The Thirring model evolving from  $(\Delta,m)=(0.5, 0.2)$  to  $(\Delta,m)=(-0.8, 0.2)$ . . . . . 41

A.1 The Thirring model evolving from  $(\Delta,m)=(0, 0)$  to  $(\Delta,m)=(0, 0.2)$ . . 46

A.2 The Thirring model evolving from  $(\Delta,m)=(0.5, 0)$  to  $(\Delta,m)=(0.5, 0.2)$ . 46

A.3 The Thirring model evolving from  $(\Delta,m)=(0, 0.2)$  to  $(\Delta,m)=(0, 0)$ . . 47

A.4 The Thirring model evolving from  $(\Delta,m)=(0.5, 0.2)$  to  $(\Delta,m)=(0.5, 0)$ . . . . . 47

A.5 The Thirring model evolving from  $(\Delta,m)=(-0.5, 0)$  to  $(\Delta,m)=(-0.5, 0.5)$ . . . . . 48

A.6 The Thirring model evolving from  $(\Delta,m)=(0, 0)$  to  $(\Delta,m)=(0, 0.5)$ . . 48

A.7 The Thirring model evolving from  $(\Delta,m)=(0.5, 0)$  to  $(\Delta,m)=(0.5, 0.5)$ . 48

A.8 The Thirring model evolving from  $(\Delta,m)=(-0.5, 0.5)$  to  $(\Delta,m)=(-0.5, 0)$ . . . . . 49

A.9 The Thirring model evolving from  $(\Delta,m)=(0, 0.5)$  to  $(\Delta,m)=(0, 0)$ . . 49

A.10 The Thirring model evolving from  $(\Delta,m)=(0.5, 0.5)$  to  $(\Delta,m)=(0.5, 0)$ . . . . . 50

A.11 The Thirring model evolving from  $(\Delta,m)=(0.5, 0.5)$  to  $(\Delta,m)=(0.5, 0.1)$ . . . . . 50

A.12 The Thirring model evolving from  $(\Delta,m)=(-0.5, 0)$  to  $(\Delta,m)=(0.5, 0)$ . 50

A.13 The Thirring model evolving from  $(\Delta,m)=(-0.8, 0)$  to  $(\Delta,m)=(0.5, 0)$ . 51

A.14 The Thirring model evolving from  $(\Delta,m)=(0.5, 0.2)$  to  $(\Delta,m)=(0.2, 0.2)$ . . . . . 51

A.15 The Thirring model evolving from  $(\Delta,m)=(0.5, 0.2)$  to  $(\Delta,m)=(0, 0.2)$ . 51

A.16 The Thirring model evolving from  $(\Delta,m)=(0.5, 0.2)$  to  $(\Delta,m)=(-0.5, 0.2)$ . . . . . 52

A.17 The Thirring model evolving from  $(\Delta,m)=(0.2, 0.2)$  to  $(\Delta,m)=(0.5, 0.2)$ . . . . . 52

A.18 The Thirring model evolving from  $(\Delta,m)=(0, 0.2)$  to  $(\Delta,m)=(0.5, 0.2)$ . 52

A.19 The Thirring model evolving from  $(\Delta,m)=(0, 0.2)$  to  $(\Delta,m)=(-0.8, 0.2)$ . . . . . 53

A.20 The Thirring model evolving from  $(\Delta,m)=(-0.8, 0.2)$  to  $(\Delta,m)=(0, 0.2)$ . . . . . 53



# Chapter 1

## Introduction

### 1.1 Overview

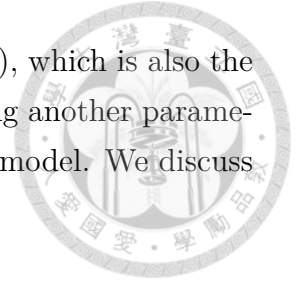
Tensor network methods are taking a central role in modern quantum physics, especially in quantum many-body systems [1, 2]. In these methods, the quantum state is represented as a tensor network, which we call a tensor network state. There are several forms of tensor network states such as the matrix product state (MPS), which is a good representation of quantum states in 1D lattice systems, and the projected entangled pair state (PEPS), which is a representation of quantum states in 2D lattice systems. In this thesis, we will introduce the Thirring model on 1D infinite lattice in Chapter 2, and we use the so-called uniform matrix product state (uMPS) (or infinite MPS (iMPS)), which we will introduce in Chapter 3, to represent our quantum states.

There are several algorithms based on the tensor network methods for finding the ground state of system, such as density matrix renormalization group (DMRG) [3] and infinite time-evolving block decimation (iTEBD) [4]. In this thesis, we use the variational optimization algorithm for uniform matrix product states (VUMPS) [5], which will be discussed in Chapter 4, to find the ground state of the Thirring model.

There are tensor network algorithms for simulating real-time evolution for many-body system on 1D lattices such as iTEBD or the time-dependent variational principle (TDVP) [6]. We extend the original TDVP algorithm using the idea of the infinite boundary conditions for MPS [7], and then we can use the matrix product operator (MPO) to do TDVP simulations (see Chapter 5).

A quantum quench is a protocol in which one prepares an eigenstate of one Hamiltonian,  $H_0$ , and then evolves dynamically in time under a different Hamiltonian,  $H_1$ . We want to ask whether the Thirring model exists the dynamical phase transition [8]. We use the VUMPS algorithm to prepare the ground state of the

Thirring model with a given coupling constant ( $\Delta$ ) and mass ( $m$ ), which is also the eigenstate of Hamiltonian  $H_0$ , and do real-time evolution by using another parameters ( $\Delta', m'$ ) to investigate the quench dynamics of the Thirring model. We discuss our results in Chapter 6.





## Chapter 2

# Thirring Model

### 2.1 Spin Representation of the Thirring Model

We want to discretize the massive Thirring model onto a 1D lattice<sup>1</sup>. The massive Thirring model is a theory of Dirac fields in the (1+1)-dimensional spacetime with a current-current interaction; it is described by the action

$$S(\psi, \bar{\psi}) = \int d^2x \left[ \bar{\psi} i \gamma^\mu \partial_\mu \psi - m \bar{\psi} \psi - \frac{g}{2} (\bar{\psi} \gamma^\mu \psi)^2 \right], \quad (2.1)$$

where  $m$  denotes the mass and  $g$  is the coupling constant of the current-current interaction term  $(\bar{\psi} \gamma^\mu \psi)^2$ . The canonical momentum is

$$\Pi = \frac{\partial \mathcal{L}}{\partial \dot{\psi}} = \bar{\psi} i \gamma_0 = i \psi^\dagger. \quad (2.2)$$

Thus, the Hamiltonian of the Thirring model becomes

$$\begin{aligned} H &= \int dx (\Pi \partial_0 \psi - \mathcal{L}) \\ &= \int dx \left[ \bar{\psi} i \gamma_0 \partial_0 \psi - \bar{\psi} i \gamma^\mu \partial_\mu \psi + m \bar{\psi} \psi + \frac{g}{2} (\bar{\psi} \gamma^\mu \psi)^2 \right] \\ &= \int dx \left[ -\bar{\psi} i \gamma^1 \partial_1 \psi + m \bar{\psi} \psi + \frac{g}{2} (\bar{\psi} \gamma^\mu \psi)^2 \right]. \end{aligned} \quad (2.3)$$

We choose the  $\gamma$  matrices as

$$\gamma_0 = \sigma_3, \quad \gamma_1 = i \sigma_2, \quad (2.4)$$

where  $\sigma_i$  are the Pauli matrices. We can see that the  $\gamma$  matrices will satisfy the Clifford algebra as  $\{\gamma_\mu, \gamma_\nu\} = 2g_{\mu\nu}$ , where  $g_{\mu\nu}$  is the metric of the Minkowski space chosen as (1,-1).

---

<sup>1</sup>The part of the discussion is based on Ref. [9].



In addition, we will use  $\phi_u$  and  $\phi_d$  to denote the up and down components of the field. Namely,

$$\psi = \begin{pmatrix} \phi_u \\ \phi_d \end{pmatrix}. \quad (2.5)$$

Then, Eq. (2.3) becomes

$$H = \int dx [-i(\psi_d^* \partial_1 \psi_u + \psi_u^* \partial_1 \psi_d) + m(\psi_u^* \psi_u - \psi_d^* \psi_d) + 2g(\psi_u^* \psi_u \psi_d^* \psi_d)]. \quad (2.6)$$

We want to discretize the Hamiltonian on a 1D infinite lattice with the lattice constant  $a$  and use ‘‘Kogut-Susskind fermions’’ [10], which put the up component  $\psi_u$  on the even sites and the down component  $\psi_d$  on the odd sites. Hence,

$$\psi_u \rightarrow \frac{1}{\sqrt{a}} c_{2n}, \quad \psi_d \rightarrow \frac{1}{\sqrt{a}} c_{2n+1} \quad (2.7)$$

and the differentials in space dimension are

$$\partial_1 \psi_u(x) = \frac{1}{\sqrt{a}} \frac{(c_{2n+1} - c_{2n-1})}{2a}, \quad \partial_1 \psi_d(x) = \frac{1}{\sqrt{a}} \frac{(c_{2n+2} - c_{2n})}{2a}. \quad (2.8)$$

Then Eq. (2.6) becomes

$$\begin{aligned} H_{\text{lattice}} = & -\frac{i}{2a} \sum_n \left( c_n^\dagger c_{n+1} - c_{n+1}^\dagger c_n \right) + m \sum_n (-1)^n c_n^\dagger c_n \\ & + \frac{2g}{a} \sum_n c_{2n}^\dagger c_{2n} c_{2n+1}^\dagger c_{2n+1}. \end{aligned} \quad (2.9)$$

We use Jordan-Wigner transformation, which maps spin operators onto fermionic creation and annihilation operators, to transform the Hamiltonian of the Thirring model to the spin representation. Jordan-Wigner transformation can be represented as

$$c_n = \exp\left(\pi i \sum_{j=1}^{n-1} S_j^z\right) S_n^-, \quad c_n^\dagger = S_n^+ \exp\left(-\pi i \sum_{j=1}^{n-1} S_j^z\right). \quad (2.10)$$

Substituting Eq. (2.10) into (2.9), then we get

$$\begin{aligned} H_{\text{spin}} = & -\frac{1}{2a} \sum_n (S_n^+ S_{n+1}^- + S_n^- S_{n+1}^+) + m \sum_n (-1)^n \left( S_n^z + \frac{1}{2} \right) \\ & + \frac{2g}{a} \sum_n \left( S_{2n}^z + \frac{1}{2} \right) \left( S_{2n+1}^z + \frac{1}{2} \right). \end{aligned} \quad (2.11)$$

Finally, we have the spin representation of the Thirring model. There is a problem here: this spin representation of the Thirring model breaks the lattice shift symmetry. So, we need to complete the staggered term in the Hamiltonian.

$$\begin{aligned} H_{\text{spin}} = & -\frac{1}{2a} \sum_n (S_n^+ S_{n+1}^- + S_n^- S_{n+1}^+) + m \sum_n (-1)^n \left( S_n^z + \frac{1}{2} \right) \\ & + \frac{\Delta}{a} \sum_n \left( S_n^z + \frac{1}{2} \right) \left( S_{n+1}^z + \frac{1}{2} \right), \end{aligned} \quad (2.12)$$

where  $\Delta(g) = \frac{\cos(\pi - g)}{2}$ ; this mapping is based on Bethe ansatz [11]. In order to find the ground states with the constraint  $\langle S_z \rangle = \langle S_{\text{target}} \rangle$ , we add a penalty term [12] to the spin Hamiltonian:

$$H_{\text{spin}}^{\text{penalty}} = H_{\text{spin}} + \lambda \sum_n (S_n^z - S_{\text{target}})^2 \quad (2.13)$$

In practice, this works very well when  $\lambda > 1$ . For this thesis, we set  $S_{\text{target}} = 0$ .

## 2.2 Chiral Condensate

The chiral condensate can be defined as

$$\left| \int dx \bar{\psi} \psi \right|. \quad (2.14)$$

We can use same technique as previous section. We can use “Kogut-Susskind fermions” and Jordan-Wigner transformation:

$$\begin{aligned} \left| \int dx \bar{\psi} \psi \right| &= \left| \int dx \psi^\dagger \gamma_0 \psi \right| = \left| \int dx \psi_u^* \psi_u - \psi_d^* \psi_d \right| \\ &\rightarrow \lim_{N \rightarrow \infty} \frac{1}{N} \left| \sum_n (-1)^n c_n^* c \right| = \lim_{N \rightarrow \infty} \frac{1}{N} \left| \sum_n (-1)^n \left( S_n^z + \frac{1}{2} \right) \right|. \end{aligned} \quad (2.15)$$

Here  $N$  is the number of sites, which we add by hand. For infinite systems, we cannot measure the entire condensate, only the condensate at each site can be measured.

## 2.3 Mapping to the Classical 2D XY Model

There exists a mapping [9] between the coupling constant ( $g$ ) of the Thirring model and the temperature ( $T$ ) in the 2D classical XY model:

$$H_{XY} = -K \sum_{\langle i,j \rangle} \cos(\theta_i - \theta_j). \quad (2.16)$$

The relation is

$$g = \frac{T}{K} - \pi. \quad (2.17)$$

The classical 2D XY model undergoes a Berezinskii-Kosterlitz-Thouless transition (BKT transition) [13, 14] at  $T = \frac{K\pi}{2}$ , the corresponding  $\Delta$  is  $-\frac{1}{\sqrt{2}} \approx -0.707$ . For  $\Delta < -0.707$ , the quantum state is in the KT phase.



## Chapter 3

# Tensor Network and Matrix Product state

### 3.1 Tensor Network and Tensor Diagram

Tensors can be classified by their rank. For example, a rank-0 tensor is a scalar, a rank-1 tensor is a vector and a rank-2 tensor is a matrix. We can also define tensors for higher ranks. The *index contraction* is the sum over all the possible values of the repeated indices of a set of tensors. A tensor can also be constructed by index contraction. For instance, the matrix product

$$C_{\alpha\gamma} = \sum_{\beta} A_{\alpha\beta} B_{\beta\gamma} \quad (3.1)$$

is the contraction of index  $\beta$ , which amounts to the sum over its possible values. One can also have more complicated contractions, such as :

$$\Theta_{afeg} = \sum_{bcd} A_{bac} B_{dce} H_{bdfg} \quad (3.2)$$

From these examples, we see that we can construct a tensor by contracting combination of other tensors and we call these combinations “tensor networks.” It is too tedious to write every index when the network is very complicated; for convenience, we introduce “tensor network diagrams” which can represent tensor networks more clearly (see Fig. 3.1). Tensor network diagrams allow to handle complicated expressions in a visual way. For instance, the contractions in Eqs. (3.1) and (3.2) can be represented by the diagrams in Fig. 3.2. Hereafter, we will primarily use tensor network diagrams to represent tensors and tensor networks.

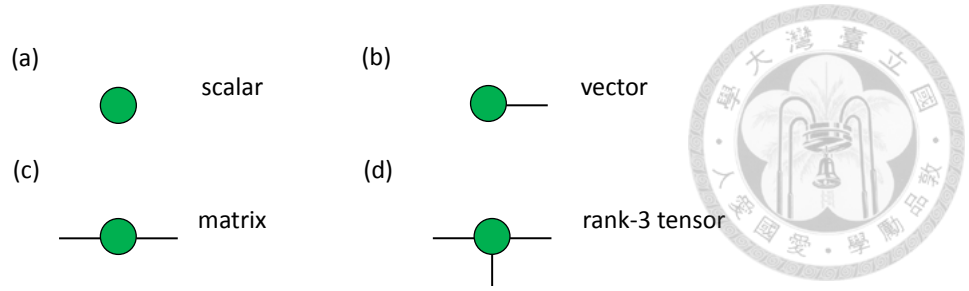


Figure 3.1: Tensor network diagrams (a) a scalar, (b) a vector, (c) a matrix and (d) a rank-3 tensor.

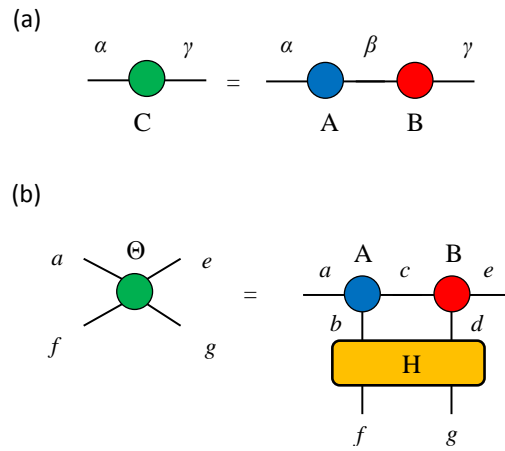


Figure 3.2: Tensor network diagram for Eqs. (3.1) and (3.2).

## 3.2 Matrix Product States

Tensor network states have emerged as a very useful conceptual and numerical framework for studying quantum many-body systems [2]. For example, if we consider a 1D lattice system with  $d$ -dimensional local state space  $\{\sigma_i\}$  on sites  $i = 1, 2, \dots, L$ , we can write the quantum state as:

$$|\phi\rangle = \sum_{\sigma_1, \dots, \sigma_L} c_{\sigma_1, \dots, \sigma_L} |\sigma_1, \dots, \sigma_L\rangle \quad (3.3)$$

For the tensor  $c_{\sigma_1, \dots, \sigma_L}$ , we can use the singular-value decomposition (SVD) to decompose it into

$$\begin{aligned} c_{\sigma_1, \dots, \sigma_L} &= \sum_{a_1}^{r_1} U_{\sigma_1, a_1} S_{a_1, a_1} V_{a_1, \sigma_2, \dots, \sigma_L}^\dagger \\ &= \sum_{a_1}^{r_1} U_{\sigma_1, a_1} c_{a_1, \sigma_2, \dots, \sigma_L} = \sum_{a_1}^{r_1} A_{\sigma_1}^{a_1} c_{a_1, \sigma_2, \dots, \sigma_L}, \end{aligned} \quad (3.4)$$

where  $r_1 \leq d$ ,  $c_{a_1, \sigma_2, \dots, \sigma_L} \equiv S_{a_1, a_1} V_{a_1, \sigma_2, \dots, \sigma_L}^\dagger$  and  $A_{\sigma_1}^{a_1} \equiv U_{\sigma_1, a_1}$ . We can repeat this

process for the tensor  $c_{a_1, \sigma_2, \dots, \sigma_L}$ , then

$$\begin{aligned}
& \sum_{a_1}^{r_1} A_{\sigma_1}^{a_1} c_{a_1, \sigma_2, \dots, \sigma_L} \\
&= \sum_{a_1}^{r_1} \sum_{a_2}^{r_2} A_{\sigma_1}^{a_1} U_{(a_1, \sigma_2), a_2} S_{a_2, a_2} V_{a_2, \sigma_3, \dots, \sigma_L}^\dagger \\
&\equiv \sum_{a_1}^{r_1} \sum_{a_2}^{r_2} A_{\sigma_1}^{a_1} U_{(a_1, \sigma_2), a_2} c_{a_2, (\sigma_3, \dots, \sigma_L)} \\
&\equiv \sum_{a_1}^{r_1} \sum_{a_2}^{r_2} A_{\sigma_1}^{a_1} A_{a_1, a_2}^{\sigma_2} c_{a_2, (\sigma_3, \dots, \sigma_L)}
\end{aligned} \tag{3.5}$$



Here,  $r_2 < r_1 \times d$ . If we perform this process iteratively, we will observe that  $r_n$  increases exponentially, so we set a maximum bond dimension  $D$  such that

$$r_n = \begin{cases} d^n, & \text{if } d^n < D \\ D, & \text{if } d^n \geq D \end{cases} \tag{3.6}$$

and truncate the bond dimension by SVD, which avoids the bond dimension  $D$  exponential growth. Finally we can decompose the original tensor  $c_{\sigma, \dots, \sigma_L}$  as:

$$c_{\sigma_1, \dots, \sigma_L} = \sum_{a_1, \dots, a_{L-1}} A_{a_1}^{\sigma_1} A_{a_1, a_2}^{\sigma_2} \dots A_{a_{L-2}, a_{L-1}}^{\sigma_{L-1}} A_{a_{L-1}}^{\sigma_L}. \tag{3.7}$$

Or more compactly,

$$c_{\sigma_1, \dots, \sigma_L} = A^{\sigma_1} A^{\sigma_2} \dots A^{\sigma_{L-1}} A^{\sigma_L}. \tag{3.8}$$

Here  $A^{\sigma_i}$  is a set of matrices with  $r_{i-1} \times r_i$  elements. We can represent the quantum state using  $A^{\sigma_i}$ :

$$|\psi\rangle = \sum_{\sigma_1, \dots, \sigma_L} A^{\sigma_1} A^{\sigma_2} \dots A^{\sigma_{L-1}} A^{\sigma_L} |\sigma_1, \dots, \sigma_L\rangle \tag{3.9}$$

and it is just a set of matrix product with each other, so we call the quantum state in this form as matrix product state (MPS). Because at each SVD,  $M = USV^\dagger$ ,  $U^\dagger U = I$  holds, we have the following properties of this kind of MPS as we construct above.

$$\sum_{\sigma_l} A^{\sigma_l \dagger} A^{\sigma_l} = I \tag{3.10}$$

We call this kind of MPS as “left canonical form” (See Fig. 3.3). Similarly, we can do the same process from the other side as in Fig. 3.3(b) and we call it “right canonical form”

$$|\psi\rangle = \sum_{\sigma_1, \dots, \sigma_L} B^{\sigma_1} B^{\sigma_2} \dots B^{\sigma_{L-1}} B^{\sigma_L} |\sigma_1, \dots, \sigma_L\rangle, \tag{3.11}$$



which will satisfy:

$$\sum_{\sigma_i} B^{\sigma_i} B^{\sigma_i \dagger} = I. \quad (3.12)$$

And also we can construct mixed canonical form:

$$|\psi\rangle = \sum_{\sigma_1, \dots, \sigma_L} A^{\sigma_1} A^{\sigma_2} \dots \Gamma \dots A^{\sigma_{L-1}} A^{\sigma_L} |\sigma_1, \dots, \sigma_L\rangle, \quad (3.13)$$

where  $\Gamma$  is a diagonal matrix.

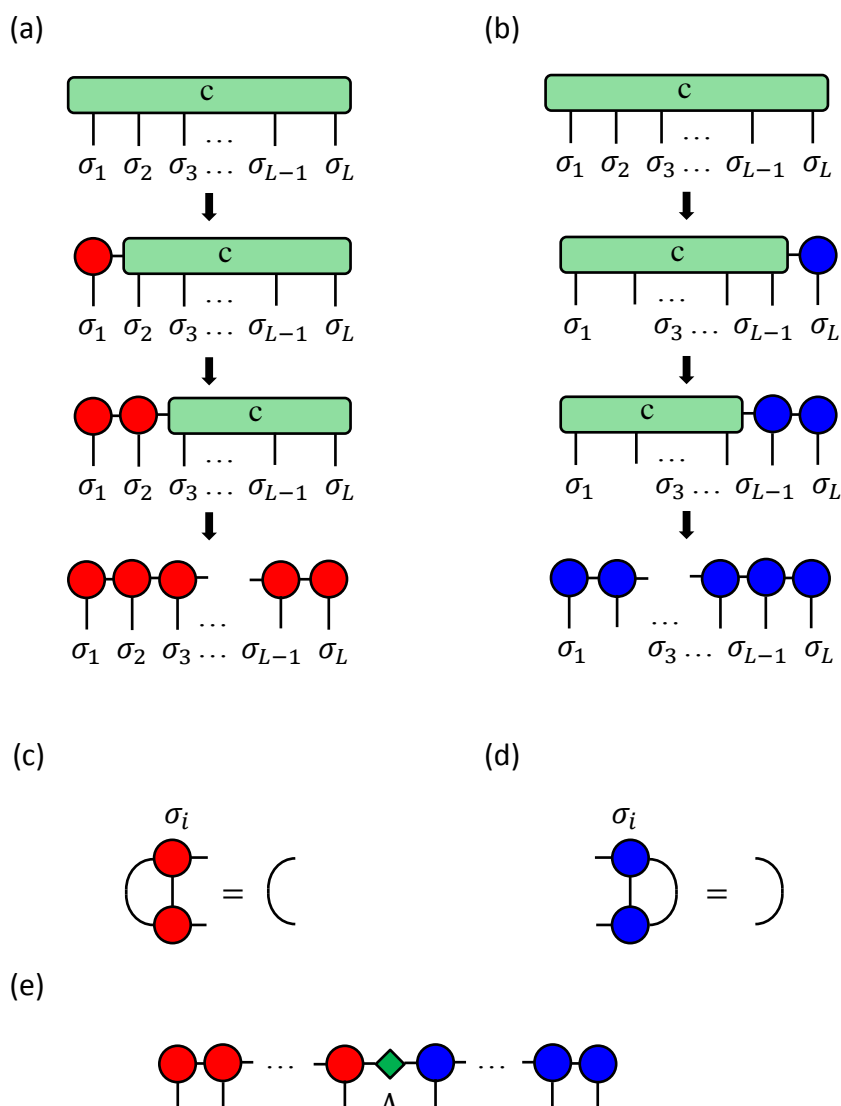


Figure 3.3: (a) Graphical representation of an iterative construction of left-canonical MPS from arbitrary quantum state by SVD. (b) Graphical representation of an iterative construction of right-canonical MPS from arbitrary quantum state by SVD. (c) The property of left canonical form. (d) The property of right canonical form. (e) Tensor network diagram of mixed canonical form.



### 3.3 Uniform Matrix Product States

When we consider 1D quantum lattices in the thermodynamic limit, we need to impose translation symmetry. Hence, we can represent our quantum state in MPS with bond dimension  $D$  as

$$|\psi(A)\rangle = \sum_{\mathbf{s}} \dots A^{s_{n-1}} A^{s_n} A^{s_{n+1}} \dots |\mathbf{s}\rangle, \quad (3.14)$$

where  $A^s \in \mathbb{C}^{D \times D}$  and  $s = 1, \dots, d$  and can be represented diagrammatically as

$$\dots \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \dots \quad (3.15)$$

The MPS in Eq. 3.14 is called “uniform matrix product states (uMPS).” For a given uMPS  $|\psi(A)\rangle$ , we can define the transfer matrix  $E = \sum_s A^s \otimes \bar{A}^s$ , or graphically,

$$\begin{array}{c} \text{---} \boxed{A} \text{---} \\ | \\ \text{---} \boxed{\bar{A}} \text{---} \end{array} , \quad (3.16)$$

which is an operator acting on the space of  $D \times D$  matrices. This kind of matrix has the property that the leading eigenvalue  $\eta$  (the eigenvalue with maximum norm) will be positive, and can be scaled to 1 by rescaling the uMPS tensor as  $A \rightarrow \frac{A}{\sqrt{\eta}}$ . We denote the corresponding left and right eigenvectors as  $l$  and  $r$  and they satisfy the eigenvalue equations:

$$\begin{array}{c} \text{---} \boxed{A} \text{---} \\ | \\ \text{---} \boxed{A} \text{---} \end{array} \begin{array}{c} \text{---} l \text{---} \\ \text{---} l \text{---} \end{array} = \begin{array}{c} \text{---} l \text{---} \\ \text{---} l \text{---} \end{array} , \quad \begin{array}{c} \text{---} \boxed{A} \text{---} \\ | \\ \text{---} \boxed{A} \text{---} \end{array} \begin{array}{c} \text{---} r \text{---} \\ \text{---} r \text{---} \end{array} = \begin{array}{c} \text{---} r \text{---} \\ \text{---} r \text{---} \end{array} . \quad (3.17)$$

And we can normalize the leading eigenvectors  $l$  and  $r$  such that  $\text{Tr}(rl) = 1$ , or diagrammatically,

$$\begin{array}{c} \text{---} \text{---} \\ | \quad | \\ \text{---} l \text{---} \quad \text{---} r \text{---} \\ | \quad | \\ \text{---} \text{---} \end{array} = 1. \quad (3.18)$$

### 3.4 Expectation Values



Suppose we want to compute the expectation value  $\langle O \rangle$  with operator

$$\langle O \rangle = \sum_{n \in \mathbb{Z}} \frac{1}{\mathbb{Z}} O_n, \tag{3.19}$$

where  $\mathbb{Z}$  represents the number of sites. Since translation invariance, so  $\langle O \rangle$  can be represented as in Fig. 3.4.

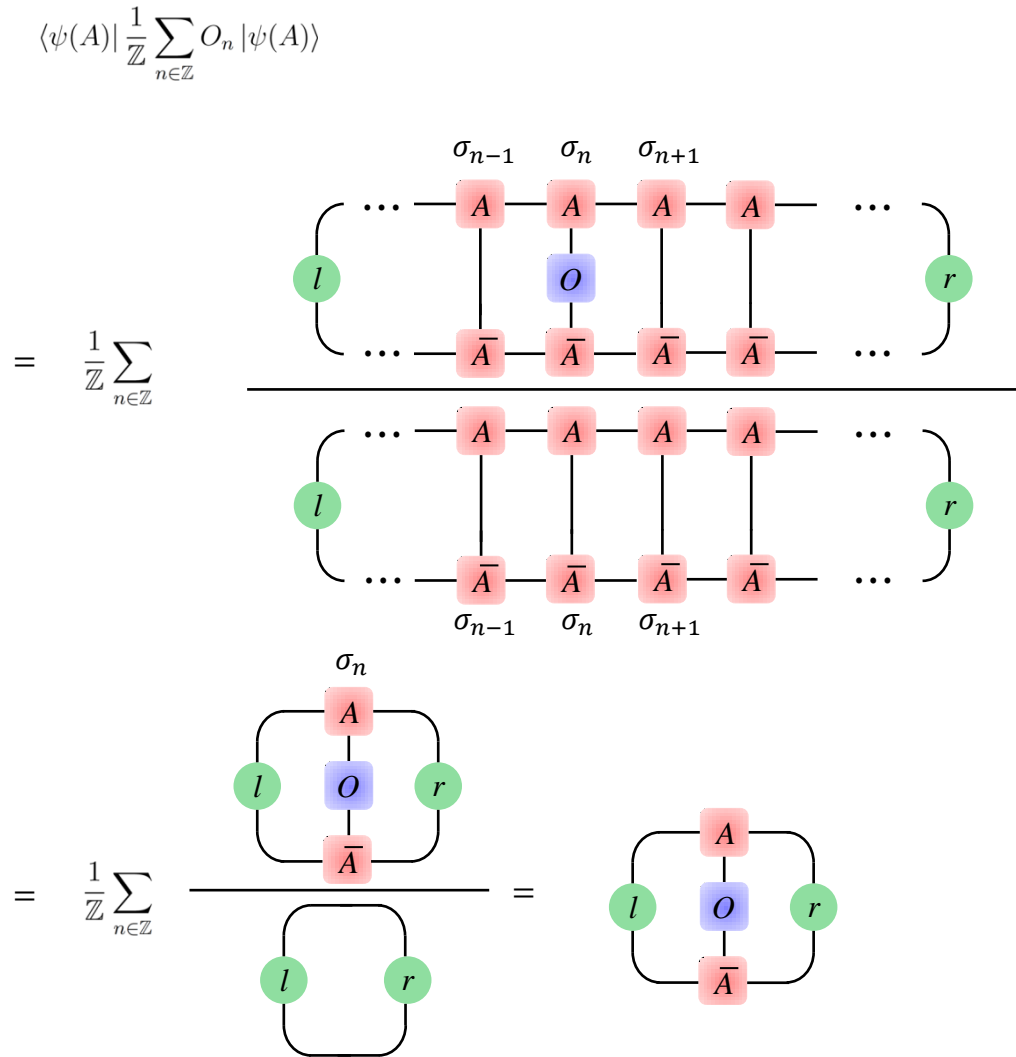


Figure 3.4: Expectation value represented as a tensor network diagram.



### 3.5 Gauge Degrees of Freedom, Canonical Form and Symmetric gauge



There is no unique way to write down a uMPS to represent a given quantum state on a 1D infinite lattice. For a given uMPS  $|\psi(A)\rangle$ , we can use a local gauge transformation  $A^s \rightarrow XA^sX^{-1}$ , with invertible  $X \in \mathbb{C}^{D \times D}$  to represent the same quantum state. Namely,  $|\psi(A)\rangle = |\psi(XAX^{-1})\rangle$ , since

$$\begin{aligned}
 & |\psi(XAX^{-1})\rangle \\
 = & \dots \underbrace{\text{---} \boxed{X} \text{---}}_I \underbrace{\boxed{A} \text{---}}_I \underbrace{\text{---} \boxed{X^{-1}} \text{---}}_I \underbrace{\boxed{X} \text{---}}_I \underbrace{\boxed{A} \text{---}}_I \underbrace{\text{---} \boxed{X^{-1}} \text{---}}_I \underbrace{\boxed{X} \text{---}}_I \underbrace{\boxed{A} \text{---}}_I \underbrace{\text{---} \boxed{X^{-1}} \text{---}}_I \dots \\
 = & \dots \boxed{A} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \dots = |\psi(A)\rangle. \tag{3.20}
 \end{aligned}$$

We introduce  $\Gamma, \Lambda$  notation to represent uMPS. For a given uMPS  $|\psi(A)\rangle$ , we can use SVD to decompose the uMPS as in Fig. 3.5, denoting the diagonal term as  $\Lambda$  and the other term as  $\Gamma$ . In other words, we can use  $(\Lambda, \Gamma)$  to represent uMPS.

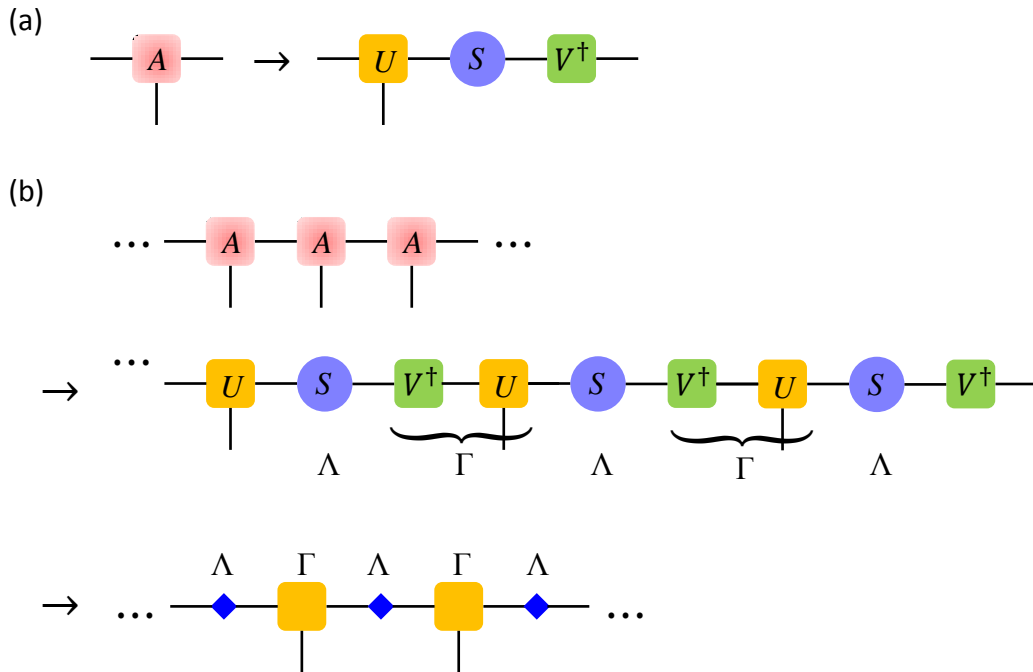


Figure 3.5: (a) Use SVD to decompose tensor  $A$ . (b)  $(\Lambda, \Gamma)$  notation for uMPS.

Furthermore, we say  $(\Lambda, \Gamma)$  is canonical if

$$\left( \begin{array}{c} \Lambda \\ \diamond \\ \Gamma \\ \diamond \\ \bar{\Gamma} \\ \Lambda \end{array} \right) = \left( \begin{array}{c} \Gamma \\ \diamond \\ \bar{\Gamma} \\ \diamond \\ \Lambda \end{array} \right) \quad (3.21)$$

An arbitrary uMPS  $(\Lambda, \Gamma)$  is not generically canonical, but it can be changed to the canonical form using a gauge transformation [15]. We will show the step by step process to change an arbitrary uMPS  $(\Lambda, \Gamma)$  to the canonical form  $(\Lambda', \Gamma')$ . First, we need to find the eigentensors  $l$  and  $r$  by solving the following eigenvalue equations,

$$\left( \begin{array}{c} \Lambda \\ \diamond \\ \Gamma \\ \diamond \\ \bar{\Gamma} \\ \Lambda \end{array} \right) l = l, \quad \left( \begin{array}{c} \Gamma \\ \diamond \\ \bar{\Gamma} \\ \diamond \\ \Lambda \end{array} \right) r = r \quad (3.22)$$

Notice that tensor  $l$  and  $r$  are Hermitian after simple resized. Then we find the matrices  $L$  and  $R$  such that  $L^\dagger L = l$  and  $RR^\dagger = r$ , so we can choose  $L = l^{\frac{1}{2}}$  and  $R = r^{\frac{1}{2}}$ . Finally, we insert these matrices and do SVD as following step:

$$\begin{aligned} & \dots \text{---} \Lambda \text{---} \Gamma \text{---} \Lambda \text{---} \dots \\ &= \dots \text{---} L^{-1} \text{---} L \text{---} \diamond \text{---} R \text{---} R^{-1} \text{---} \Gamma \text{---} L^{-1} \text{---} L \text{---} \diamond \text{---} R^{-1} \text{---} R \text{---} \dots \\ & \quad \downarrow \text{SVD} \quad \downarrow \text{SVD} \\ & \dots \text{---} U \text{---} \Lambda' \text{---} V^\dagger \text{---} \Gamma \text{---} L^{-1} \text{---} U \text{---} \Lambda' \text{---} V^\dagger \text{---} R \text{---} \dots \\ &= \dots \text{---} \Lambda' \text{---} \Gamma' \text{---} \Lambda' \text{---} \dots \end{aligned} \quad (3.23)$$

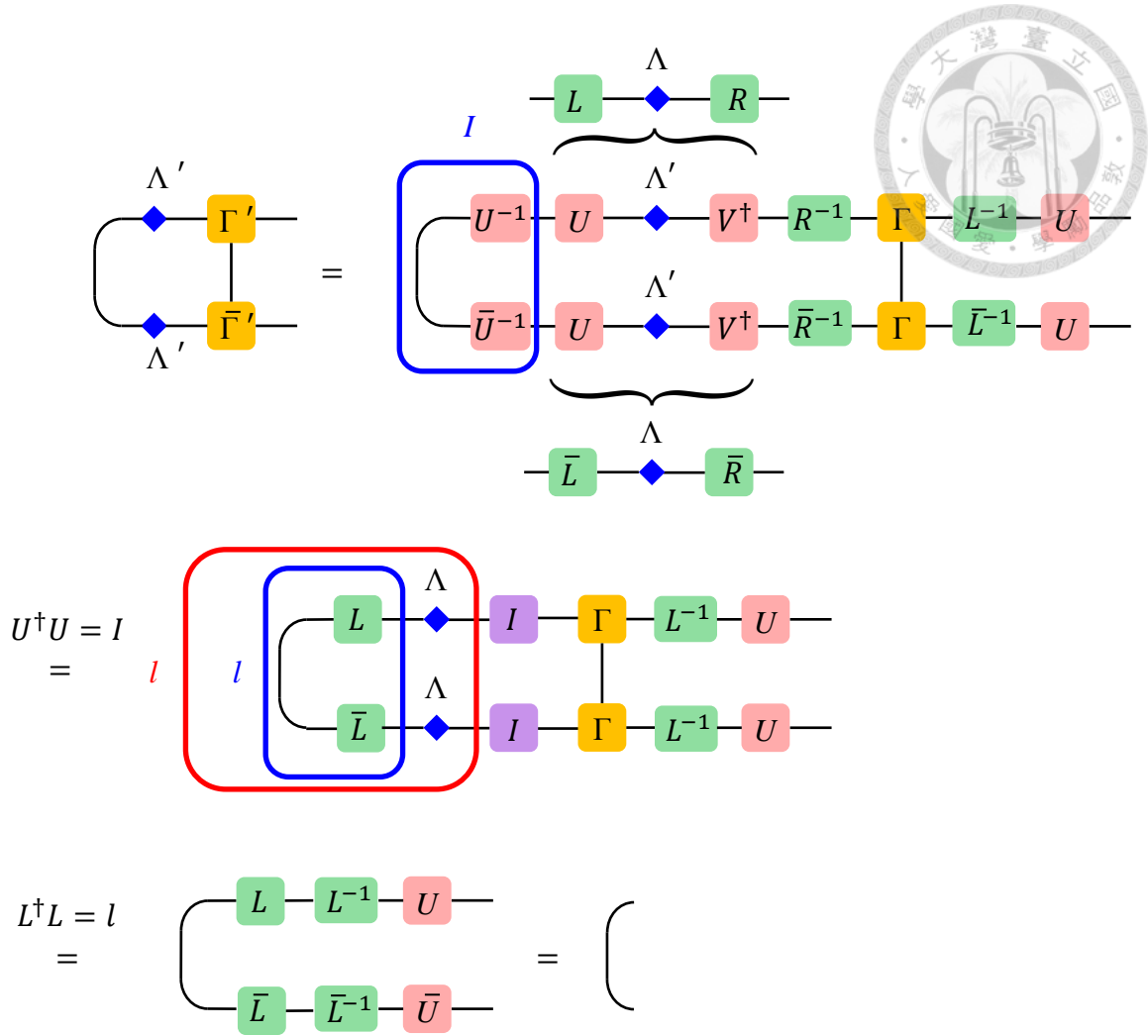


Figure 3.6: Check the property of the left canonical form.

Then we get the canonical form  $(\Lambda', \Gamma')$  and we can easily check the left-canonical form property (as in Fig. 3.6) and also for the right canonical form property. Now we can construct a unit tensor again by using canonical  $(\Lambda, \Gamma)$  as

$$\begin{array}{cc}
 \text{(a) left canonical form} & \text{(b) right canonical form} \\
 \begin{array}{c} \square A \\ | \end{array} = \begin{array}{c} \Lambda \\ \diamond \\ \Gamma \end{array} & \begin{array}{c} \square A \\ | \end{array} = \begin{array}{c} \Gamma \\ \diamond \\ \Lambda \end{array} \quad (3.24)
 \end{array}$$

Another special gauge we shall introduce here is symmetric gauge, which can be constructed by canonical  $(\Lambda, \Gamma)$  as

$$\begin{array}{c} \square A \\ | \end{array} = \begin{array}{c} \Lambda^{\frac{1}{2}} \\ \diamond \\ \Gamma \\ \diamond \\ \Lambda^{\frac{1}{2}} \end{array} \quad (3.25)$$

We can observe that the left and right eigenvectors of this kind of transfer matrix are  $\Lambda$ , since

$$\Lambda \begin{array}{c} \diamond \\ \Lambda/2 \\ \Gamma \\ \Lambda/2 \\ \Gamma\bar{ } \\ \Lambda/2 \\ \diamond \end{array} = \underbrace{\begin{array}{c} \Lambda \\ \Gamma \\ \Lambda \\ \Gamma\bar{ } \\ \Lambda/2 \\ \diamond \end{array}}_I \Lambda \quad (3.26)$$

$$\begin{array}{c} \Lambda/2 \\ \Gamma \\ \Lambda/2 \\ \Gamma\bar{ } \\ \Lambda/2 \\ \diamond \end{array} \Lambda = \begin{array}{c} \Lambda/2 \\ \Gamma \\ \Lambda \\ \Gamma\bar{ } \\ \Lambda \\ \Lambda/2 \\ \diamond \end{array} \Lambda \quad (3.27)$$

Finally, we will introduce the mixed canonical form for uMPS. For a given canonical  $(\Lambda, \Gamma)$ , we can construct mixed-canonical form:

$$\dots \text{---} \begin{array}{c} A_L \\ | \\ A_L \\ | \\ A_L \\ | \\ C \\ | \\ A_R \\ | \\ A_R \end{array} \text{---} \dots = \dots \text{---} \begin{array}{c} A_L \\ | \\ A_L \\ | \\ A_C \\ | \\ A_R \\ | \\ A_R \end{array} \text{---} \dots \quad (3.28)$$

Here,  $A_L$  is left-canonical tensor and  $A_R$  is right-canonical tensor that can be constructed as above, and we define new tensors  $A_C$  and  $C$  which can be constructed by

$$\begin{array}{c} A_C \\ | \end{array} = \begin{array}{c} \Lambda \\ \diamond \\ \Gamma \\ \Lambda \\ \diamond \end{array}, \quad \begin{array}{c} C \end{array} = \begin{array}{c} \Lambda \\ \diamond \end{array} \quad (3.29)$$

The tensors  $A_L, A_R, A_C, C$  in mixed-canonical form need to satisfy the condition

$$A_L^s C = A_C = C A_R^s, \quad (3.30)$$

or diagrammatically,

$$\text{---} \square_{A_L} \text{---} \bigcirc_C \text{---} = \text{---} \square_{A_C} \text{---} = \text{---} \bigcirc_C \text{---} \square_{A_L} \text{---} \quad (3.31)$$

### 3.6 Geometric Series for Transfer Matrix

At Chapter 4 and 5, we will introduce two algorithms (time-dependent variational principle and variational optimization method for uMPS) and both of them are needed to simulate the geometric series of the transfer matrix. We know that for a given uMPS  $|\psi(A)\rangle$ , which the corresponding unit tensor  $A$  has been normalized, we can define the transfer matrix  $E = \sum_s A^s \otimes \bar{A}^s$  and its leading eigenvalue is one. If we want to calculate the geometric series of  $E$ , we can use the formula:

$$\sum_{i=0}^{\infty} E^i = I + E + E^2 + E^3 + \dots = (I - E)^{-1}. \quad (3.32)$$

First, we diagonalize the transfer matrix

$$D = P^{-1}EP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix}, \quad (3.33)$$

where  $1 > \lambda_1 > \lambda_2 > \dots$  are the eigenvalues of the transfer matrix  $E$ . Then,

$$\begin{aligned} P^{-1}(I - E)P = I - P^{-1}EP &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 - \lambda_1 & 0 & 0 \\ 0 & 0 & 1 - \lambda_2 & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \tilde{\lambda}_1 & 0 & 0 \\ 0 & 0 & \tilde{\lambda}_2 & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix}, \end{aligned} \quad (3.34)$$

where  $\tilde{\lambda}_i = 1 - \lambda_i$ , for all  $i = 1, 2, 3, \dots$ . Define  $\tilde{D} = P^{-1}(I - E)P = I - P^{-1}EP$ , then  $I - E = P\tilde{D}P^{-1}$ . We can clearly see that the determinant of matrix  $I - E$  is 0, which implies the matrix  $I - E$  is not invertible. So we define the *pseudo-inverse*<sup>1</sup>

$$(I - E)^P = P \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \tilde{\lambda}_1^{-1} & 0 & 0 \\ 0 & 0 & \tilde{\lambda}_2^{-1} & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix} P^{-1}. \quad (3.35)$$

<sup>1</sup>We can see the idea of pseudo-inverse from Ref. [6].

If  $\{\tilde{l}_i\}$  and  $\{\tilde{r}_i\}$  are the set of eigenvectors of matrix  $I - E$ , and  $\langle l|$  and  $|r\rangle$  are the leading eigenvectors of transfer matrix  $E$ . Then,

$$(I - E) = \tilde{\lambda}_0|\tilde{r}_0\rangle\langle\tilde{l}_0| + \tilde{\lambda}_1|\tilde{r}_1\rangle\langle\tilde{l}_1| + \tilde{\lambda}_2|\tilde{r}_2\rangle\langle\tilde{l}_2| + \cdots + \tilde{\lambda}_{n-1}|\tilde{r}_{n-1}\rangle\langle\tilde{l}_{n-1}|, \quad (3.36)$$

where  $n = D \times D$  and  $\lambda_0 = 0$ . We can rewrite the pseudo-inverse as

$$(I - E)^P = \tilde{\lambda}_1^{-1}|\tilde{r}_1\rangle\langle\tilde{l}_1| + \tilde{\lambda}_2|\tilde{r}_2\rangle\langle\tilde{l}_2| + \cdots + \tilde{\lambda}_{n-1}^{-1}|\tilde{r}_{n-1}\rangle\langle\tilde{l}_{n-1}| \quad (3.37)$$

and Eq. (3.36) implies that

$$I - (E - |r\rangle\langle l|) = |r\rangle\langle l| + \tilde{\lambda}_1|\tilde{r}_1\rangle\langle\tilde{l}_1| + \tilde{\lambda}_2|\tilde{r}_2\rangle\langle\tilde{l}_2| + \cdots + \tilde{\lambda}_{n-1}|\tilde{r}_{n-1}\rangle\langle\tilde{l}_{n-1}| \quad (3.38)$$

If we inverse the Eq. (3.38), then we get

$$\begin{aligned} & [I - (E - |r\rangle\langle l|)]^{-1} \\ &= |r\rangle\langle l| + \tilde{\lambda}_1^{-1}|\tilde{r}_1\rangle\langle\tilde{l}_1| + \tilde{\lambda}_2^{-1}|\tilde{r}_2\rangle\langle\tilde{l}_2| + \cdots + \tilde{\lambda}_{n-1}^{-1}|\tilde{r}_{n-1}\rangle\langle\tilde{l}_{n-1}|. \end{aligned} \quad (3.39)$$

Compare the Eq. (3.37) and Eq. (3.39), we can rewrite the pseudo-inverse of  $I - E$  as

$$(I - E)^P = [I - (E - |r\rangle\langle l|)]^{-1} - |r\rangle\langle l|. \quad (3.40)$$

In many cases, we need to operate a given left tensor or right tensor with  $(I - E)^P$ . For instance, given a tensor  $h_l$ , we want to calculate tensor  $K$ :

$$\text{Diagram: } \text{Circle}(h_l) \text{ --- } \text{Box}(I-E)^P \text{ --- } \text{Circle}(K) = \text{Circle}(K) \quad (3.41)$$

and we can denote it as

$$\langle K| = \langle h_l|(I - E)^P. \quad (3.42)$$

According to Eq. (3.40), we have:

$$\langle K| = \langle h_l|[I - (E - |r\rangle\langle l|)]^{-1} - \langle h_l|r\rangle\langle l| \quad (3.43)$$

Then,

$$\begin{aligned} & [\langle K| + \langle h_l|r\rangle\langle l|][I - (E - |r\rangle\langle l|)] = \langle h_l| \\ \Rightarrow & \langle K|[I - (E - |r\rangle\langle l|)] + \langle h_l|r\rangle\langle l| - \langle h_l|r\rangle\langle l|E + \langle h_l|r\rangle\langle l|r\rangle\langle l| = \langle h_l| \end{aligned} \quad (3.44)$$

Since  $\langle l|E = \langle l|$  and  $\langle l|r\rangle = 1$ , the Eq. (3.44) becomes

$$\langle K|[I - (E - |r\rangle\langle l|)] + \langle h_l|r\rangle\langle l| - \langle h_l|r\rangle\langle l| + \langle h_l|r\rangle\langle l| = \langle h_l|, \quad (3.45)$$

which implies

$$\langle K | [I - (E - |r\rangle\langle l|)] = \langle h_l | - \langle h_l | r \rangle \langle l|. \quad (3.46)$$

Tensor  $h_l$  is given, so we can solve the linear equation to obtain the tensor  $K$ . Rather than invert the matrix  $I - E + |r\rangle\langle l|$  directly, we can use *biconjugate gradient stabilized method* (BiCGSTAB) [16], which is a Krylov subspace method, to solve the linear equation. Its computational complexity is just  $O(D^3)$ . And we can follow the same procedure to contract right tensor with  $(I - E)^P$ .

### 3.7 Matrix Product Operator

In many cases, we can write the Hamiltonian in matrix product operator (MPO) form; for instance, the Hamiltonian of Heisenberg model is:

$$H = \frac{J}{2} \sum_n (S_n^+ S_{n+1}^- + S_n^- S_{n+1}^+) + J_z S_n^z \quad (3.47)$$

If we define an operator

$$\hat{O}^{[i]} = \begin{pmatrix} \hat{I} & 0 & 0 & 0 & 0 \\ \hat{S}^+ & 0 & 0 & 0 & 0 \\ \hat{S}^- & 0 & 0 & 0 & 0 \\ \hat{S}^z & 0 & 0 & 0 & 0 \\ 0 & (J/2)\hat{S}^- & (J/2)\hat{S}^+ & J^z \hat{S}^z & \hat{I} \end{pmatrix} = \text{---} \boxed{O} \text{---} \quad (3.48)$$

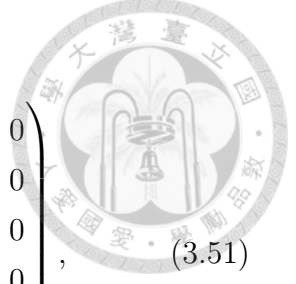
Then we can make the Heisenberg model as a product of these operators

$$H = \prod_n O^{[i]} \quad (3.49)$$

or graphically as:

$$\dots \text{---} \boxed{O} \text{---} \boxed{O} \text{---} \boxed{O} \text{---} \boxed{O} \text{---} \dots \quad (3.50)$$

For the Thirring model [Eqs. (2.12) and (2.13)] (with  $S_{\text{target}} = 0$ ), we can write



down the MPO as

$$O^{[n]} = \begin{pmatrix} I & 0 & 0 & 0 & 0 & 0 \\ S^- & 0 & 0 & 0 & 0 & 0 \\ S^+ & 0 & 0 & 0 & 0 & 0 \\ S^z & 0 & 0 & I & 0 & 0 \\ S^z & 0 & 0 & 0 & 0 & 0 \\ \beta_n S^z + \gamma I & -\frac{1}{2a} S^+ & -\frac{1}{2a} S^- & 2\lambda S^z & \frac{\Delta}{a} S^z & I \end{pmatrix}, \quad (3.51)$$

where

$$\begin{aligned} \beta_n &= \frac{\Delta}{a} + (-1)^n m \\ \gamma &= \frac{\lambda}{4} + \frac{\Delta}{4a}. \end{aligned}$$

Since this MPO acts on local pairs of sites can be represented as a tensor network diagram

$$\dots \text{---} \boxed{O_1} \text{---} \boxed{O_2} \text{---} \boxed{O_1} \text{---} \boxed{O_2} \text{---} \dots \quad (3.52)$$

This kind of MPO form is not invariant under single-site translation, so we merge two MPO to one by:

$$\begin{array}{c} \text{---} \boxed{O_1} \text{---} \boxed{O_2} \text{---} \\ | \quad | \\ 2 \quad 2 \end{array} = \begin{array}{c} \text{---} \boxed{O} \text{---} \\ | \\ 4 \end{array} \quad (3.53)$$

In this thesis, we choose physical bond dimension  $d = 4$  for uMPS.

We have already found the MPO form for the Thirring model, so we can use tensor network method to find the ground state of the Thirring model and do real-time evolution for it. We are going to introduce the time-dependent variational principle (TDVP) algorithm and variational uMPS algorithm (VUMPS) at next two chapter.





# Chapter 4

## Variational Optimization Method for uniform Matrix Product State

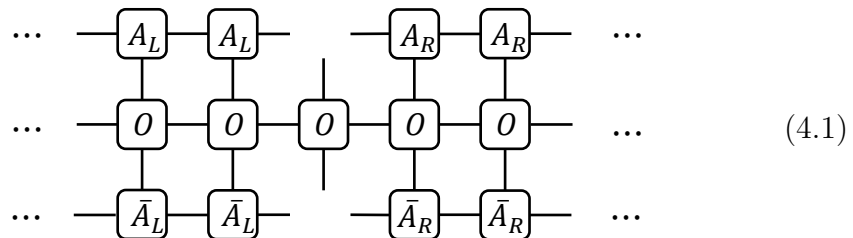
The variational optimization method for uniform matrix product state (VUMPS) [5] algorithm is a variational tensor network algorithm for determining the ground state of many-body systems on 1D lattices.

### 4.1 Effective Hamiltonian

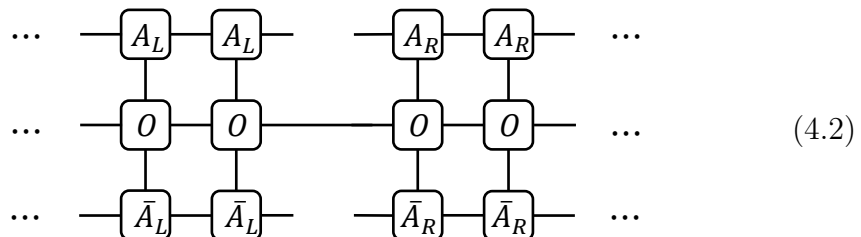
In most many-body systems, it is impossible to directly determine the ground state, even for simple nearest neighbour Hamiltonians. Instead, we will introduce an effective Hamiltonian. For a given mixed canonical uMPS  $(A_L, A_R, C)$ , we can define the one-site effective Hamiltonian and zero-site effective Hamiltonian as in Fig. 4.1. It is much easier to find the ground state of the effective Hamiltonian.

If we use MPO to represent the effective Hamiltonian, it can be represented as:

(a) single-site effective hamiltonian:



(b) zero-site effective hamiltonian:



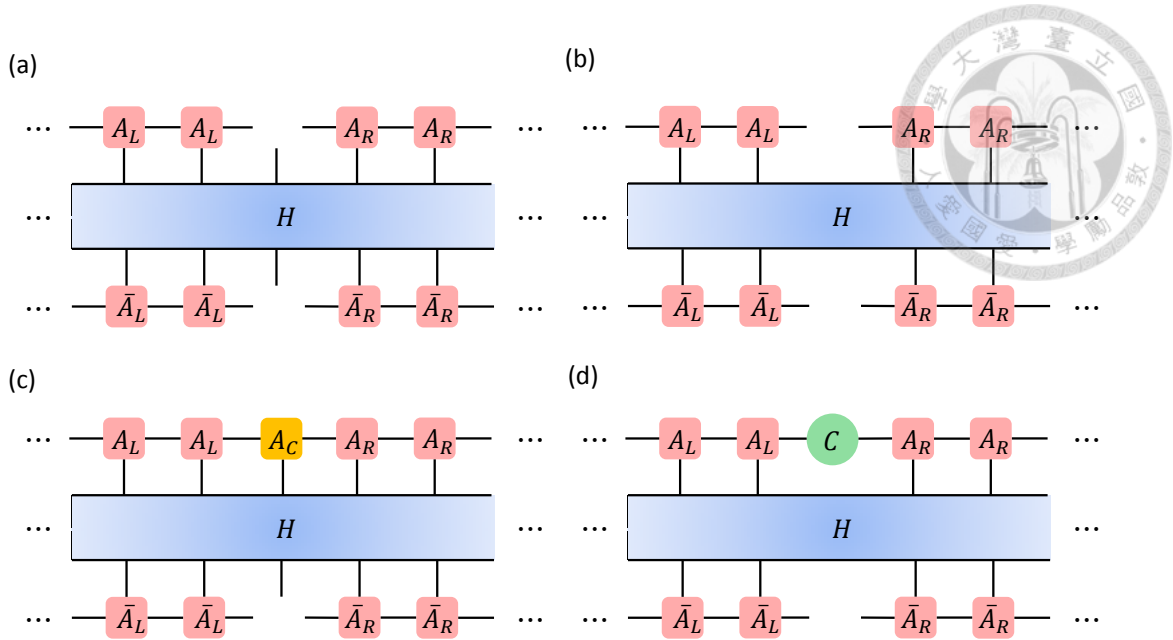


Figure 4.1: (a) One-site effective Hamiltonian, (b) zero-site effective Hamiltonian (c) acting with tensor  $A_C$  on one-site effective Hamiltonian (d) acting with tensor  $C$  on zero-site effective Hamiltonian.

In practice, we need to find a block tensor  $L$  such that

$$\begin{array}{c}
 \boxed{L} \\
 \text{---} \\
 \text{---} \\
 \text{---}
 \end{array}
 =
 \begin{array}{c}
 \dots \\
 \dots \\
 \dots
 \end{array}
 \begin{array}{cc}
 \boxed{A_L} & \boxed{A_L} \\
 \boxed{O} & \boxed{O} \\
 \boxed{\bar{A}_L} & \boxed{\bar{A}_L}
 \end{array}
 \begin{array}{c}
 \text{---} \\
 \text{---} \\
 \text{---}
 \end{array}
 ,
 \quad
 \begin{array}{c}
 \boxed{R} \\
 \text{---} \\
 \text{---} \\
 \text{---}
 \end{array}
 =
 \begin{array}{cc}
 \boxed{A_R} & \boxed{A_R} \\
 \boxed{O} & \boxed{O} \\
 \boxed{\bar{A}_R} & \boxed{\bar{A}_R}
 \end{array}
 \begin{array}{c}
 \dots \\
 \dots \\
 \dots
 \end{array}
 \quad (4.3)$$

For instance, the MPO of the Thirring model with penalty term can be represented by the following form

$$\begin{array}{c}
 \alpha \\
 | \\
 \boxed{O} \\
 | \\
 \beta
 \end{array}
 =
 \begin{array}{c}
 \alpha \\
 \beta
 \end{array}
 \begin{array}{cccccc}
 \beta & 1 & 2 & 3 & 4 & 5 & 6 \\
 \begin{pmatrix}
 I & 0 & 0 & 0 & 0 & 0 \\
 Z_2 & 0 & 0 & 0 & 0 & 0 \\
 Z_3 & 0 & 0 & 0 & 0 & 0 \\
 Z_4 & 0 & 0 & \mathbb{I} & 0 & 0 \\
 Z_5 & 0 & 0 & 0 & 0 & 0 \\
 X & Y_2 & Y_3 & Y_4 & Y_5 & I
 \end{pmatrix}
 \end{array}
 \quad (4.4)$$

And we can see that tensor  $L$  will satisfy



$$L \text{ (with } \beta \text{ lines)} = L \text{ (with } \alpha \text{ lines)} \begin{matrix} A_L \\ O \\ \bar{A}_L \end{matrix} \text{ (with } \beta \text{ lines)}. \quad (4.5)$$

We can represent the tensor  $L$  as a set of tensors:

$$L \text{ (with } \beta \text{ lines)} = \{ L_1, L_2, L_3, L_4, L_5, L_6 \}. \quad (4.6)$$

Then, construct the tensor  $L$  order by order. For  $\beta = 6$ ,

$$L_6 \text{ (with } \beta \text{ lines)} = L_6 \text{ (with } \alpha \text{ lines)} \begin{matrix} A_L \\ I \\ \bar{A}_L \end{matrix} \Rightarrow L_6 \text{ (with } \beta \text{ lines)} = \left[ \dots \right]. \quad (4.7)$$

For  $\beta = 2, 3, 5$ ,

$$L_i \text{ (with } \beta \text{ lines)} = L_6 \text{ (with } \alpha \text{ lines)} \begin{matrix} A_L \\ Y_i \\ \bar{A}_L \end{matrix} = \left[ \dots \right], \quad i = 2, 3, 5. \quad (4.8)$$

For  $\beta = 4$ ,

$$L_4 \text{ (with } \beta \text{ lines)} = L_1 \text{ (with } \alpha \text{ lines)} \begin{matrix} A_L \\ Y_4 \\ \bar{A}_L \end{matrix} + L_4 \text{ (with } \alpha \text{ lines)} \begin{matrix} A_L \\ I \\ \bar{A}_L \end{matrix}. \quad (4.9)$$

Or we can denote it as

$$L_4(n+1) = T_L(L_4(n)) + C_1, \quad (4.10)$$

where  $T_L(L_4)$  means that the transfer matrix constructed by  $A_L$  operates on the left tensor  $L_4$ , and tensor  $C_1$  is defined

$$C_1 \equiv L_1 \begin{matrix} A_L \\ Y_4 \\ \bar{A}_L \end{matrix} = \begin{matrix} A_L \\ Y_4 \\ \bar{A}_L \end{matrix} \cdot \quad (4.11)$$

If we assume  $L_4(0) = 0$ , from above recurrence relation Eq.(4.1), we can infer that

$$L_4(n) = \sum_{i=0}^{n-1} T_L^i(C_1). \quad (4.12)$$

If we want to find  $L_4(n \rightarrow \infty)$ , it just an infinite geometric series and it can be found as we discussed at Chapter 3 and should be satisfy Eq. (3.46) and we can replace  $A \rightarrow A_L$  and  $h_l \rightarrow I$ . For  $\beta = 1$ ,

$$L_1 = L_1 \begin{matrix} A_L \\ I \\ \bar{A}_L \end{matrix} + C_2 \cdot \quad (4.13)$$

Here, tensor  $C_2$  is defined

$$C_2 = L_2 \begin{matrix} A_L \\ Z_2 \\ \bar{A}_L \end{matrix} + L_3 \begin{matrix} A_L \\ Z_3 \\ \bar{A}_L \end{matrix} + L_4 \begin{matrix} A_L \\ Z_4 \\ \bar{A}_L \end{matrix} \\ + L_5 \begin{matrix} A_L \\ Z_5 \\ \bar{A}_L \end{matrix} + L_6 \begin{matrix} A_L \\ X \\ \bar{A}_L \end{matrix} \cdot \quad (4.14)$$

Note that we have already construct tensor  $L_2, L_3, L_4, L_5, L_6$ , so we can make tensor  $C_2$  directly. Then we also can observe that

$$L_1 = \sum_{i=0}^{\infty} T_L^i(C_2). \quad (4.15)$$

Then we can construct the tensor  $L_1$  by Eq. (3.46). Thus, we complete the block tensor  $L$  and can use the same process to construct tensor  $R$ . Finally, the effective Hamiltonian can be represented as in Fig. 4.2.

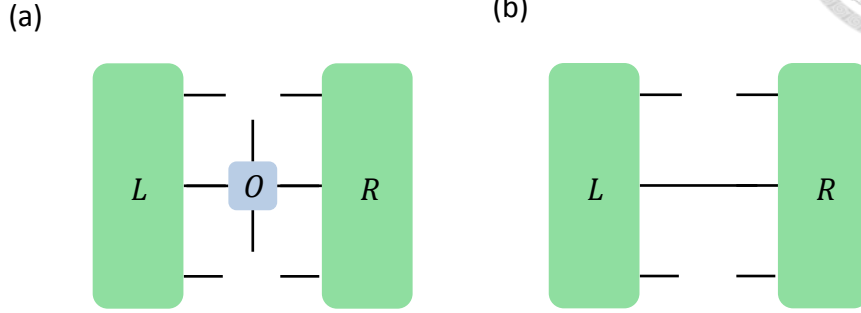


Figure 4.2: (a) one-site effective Hamiltonian (b) zero-site effective Hamiltonian

## 4.2 VUMPS algorithm

We are now ready to formulate our variational uniform MPS (VUMPS) algorithm. First, we start from a random uMPS  $(A_L, A_R, C)$ , and construct both the single-site and zero-site effective Hamiltonians as described in the previous section. Then, we can use Lanczos method to find the tensors  $A_C$  and  $C$ , which are the *eigntensor* with lowest eigenvalues  $E_{A_C}$  and  $E_C$ . Or we can denote it

$$H_{A_C} A_C = E_{A_C} A_C \quad (4.16a)$$

$$H_C C = E_C C. \quad (4.16b)$$

In general, tensors  $A_C$  and  $C$  which satisfy Eq. (4.16) will not satisfy Eq. (3.30), which is the condition for mixed-canonical form for the uMPS. We cannot find any  $A_L \in \{\tilde{A}_L | \tilde{A}_L^\dagger \tilde{A}_L = I\}$  and  $A_R \in \{\tilde{A}_R | \tilde{A}_R \tilde{A}_R^\dagger = I\}$ , such that  $A_L^s C - A_C = C A_R^s - A_C = 0$ . The second best way is to choose the tensor  $A_L$  and  $A_R$  which will update the effective Hamiltonians by solving the following equations

$$\epsilon_L = \min_{A_L^\dagger A_L = I} \|A_C^s - A_L^s C\|_2 \quad (4.17a)$$

$$\epsilon_R = \min_{A_R A_R^\dagger = I} \|A_C^s - C A_R^s\|_2 \quad (4.17b)$$

Fortunately, the solution of this kind of minimization problem has been proven that  $A_L$  will be the isometry in the polar decomposition of  $A_C^s C^\dagger$ ; more concretely, if

$$\boxed{A_C} \text{---} \textcircled{C^\dagger} = \boxed{U} \text{---} \textcircled{P} \quad \text{then} \quad \boxed{A_L} = \boxed{U} .$$

Here we decompose the tensor  $A_C^s C^\dagger$  by polar decomposition,  $U$  is unitary and  $P$  is positive semi-definite. Then the solution of Eq. (4.17a) is  $U$ . We can do the same process to solve for the tensor  $A_R$  [17]. In practice, we can do polar decomposition by SVD as following. For a given matrix  $M = W\Sigma V^\dagger$ . We can set

$$U = WV^\dagger \quad P = V\Sigma V^\dagger. \quad (4.18)$$

We can observe that the matrix  $U$  is unitary and  $P$  is positive semi-definite. And we have

$$M = W\Sigma V^\dagger = WV^\dagger V\Sigma V = UP. \quad (4.19)$$

This implies that we can use SVD to do polar decomposition. After we update  $A_L$  and  $A_R$ , we can update the effective Hamiltonians. And then, resolve the ground state of both effective Hamiltonians to find the new  $A_C$  and  $C$ . And then update  $A_L$  and  $A_R$ . We repeat this process until the Eq. (3.46) is satisfied. Below is a detailed description of the VUMPS algorithm (see also Fig. 4.3).

1. Start from random tensor  $(A_L, A_R, C)$ . Set  $\epsilon_p > \epsilon$ , where we call  $\epsilon_p$  “current precision” and  $\epsilon$  “final precision”.
2. If  $\epsilon_p > \epsilon$ , update tensors  $A_C$  and  $C$  as the ground state of the single-site and zero-site Hamiltonians. If not, return tensor  $(A_L, A_R, C)$  and end the algorithm.
3. Update tensor  $A_L$  and  $A_R$  with respect to tensors  $A_C$  and  $C$  at step 2 by Eq. (4.17), and calculate the corresponding  $\epsilon_L$  and  $\epsilon_R$ .
4. Set  $\epsilon_p = \max\{\epsilon_L, \epsilon_R\}$  and return to step 2.

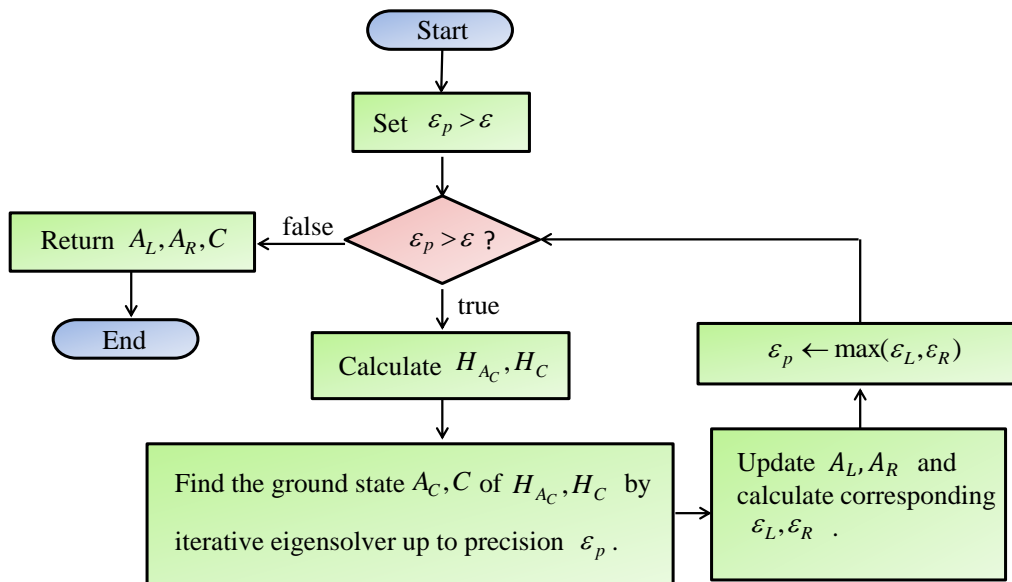


Figure 4.3: Flow chart of VUMPS algorithm.



# Chapter 5

## Time-Dependent Variational Principle Applied to Matrix Product State

Time-dependent variational principle algorithm (TDVP) is a tensor network algorithm for simulating real-time evolution for many-body system on 1D lattices. In this chapter, we are going to introduce the TDVP algorithm on infinite 1D lattice.

### 5.1 Tangent Vector Space

For a given uMPS,  $\psi(A)$ , we can use tensor diagram to represent its tangent vector as

$$\begin{aligned}
 \Phi(\{B_i\}; A) = & \\
 & \dots \text{---} \boxed{A} \text{---} \boxed{B_{n-1}} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \dots \\
 & \quad \quad \quad \vdots \quad \quad \quad s_{n-1} \quad \quad \quad s_n \quad \quad \quad s_{n+1} \quad \quad \quad \vdots \\
 + & \dots \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{B_n} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \dots \\
 & \quad \quad \quad \vdots \quad \quad \quad s_{n-1} \quad \quad \quad s_n \quad \quad \quad s_{n+1} \quad \quad \quad \vdots \\
 + & \dots \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{B_{n+1}} \text{---} \boxed{A} \text{---} \dots \\
 & \quad \quad \quad \vdots \quad \quad \quad s_{n-1} \quad \quad \quad s_n \quad \quad \quad s_{n+1} \quad \quad \quad \vdots \\
 = & \sum_n \dots \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{B_n} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \dots,
 \end{aligned} \tag{5.1}$$



where the set of tensors  $\{B_i\}$  decide the “direction” of the tangent vector. Since there is translation-invariance, we can assume  $B_1 = B_2 = B_3 = \dots = B$ , so the tangent vector can be represented as

$$\Phi(B; A) = \sum_n \dots \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{B} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \dots \quad (5.2)$$

$\vdots$                        $s_{n-1}$                        $s_n$                        $s_{n+1}$                        $\vdots$

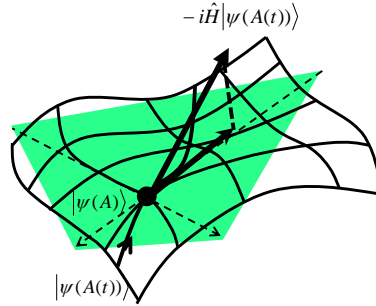


Figure 5.1: An illustration of the uMPS manifold and tangent space. The black dot represents the uMPS,  $\psi(A)$ , and the tangent vector  $\Phi(B; A)$  is a vector line on the tangent plane.

## 5.2 Gauge Fixing for Tangent Vector

It will be very convenient for us if  $B$  in tangent vector  $\Phi(B; A)$  is gauge-fixed such that

$$\sum_s A^{s\dagger} l B^s = 0, \quad \sum_s B^{s\dagger} l A^s = 0. \quad (5.3)$$

In order to ensure this condition, we parametrize  $B$  as

$$\text{---} \boxed{l^{-\frac{1}{2}}} \text{---} \boxed{V_L} \text{---} \boxed{X} \text{---} \boxed{r^{-\frac{1}{2}}} \text{---} \quad (5.4)$$

where  $l$  and  $r$  are the leading left and right eigentensors of the transfer matrix and  $V_L$  is the orthonormal basis for the null space (see Fig. 5.2(a)) of the tensor

$$\quad (5.5)$$

which we find using SVD. Fig. 5.2(b) shows that this parametrization will satisfy the condition at Eq. (5.3).

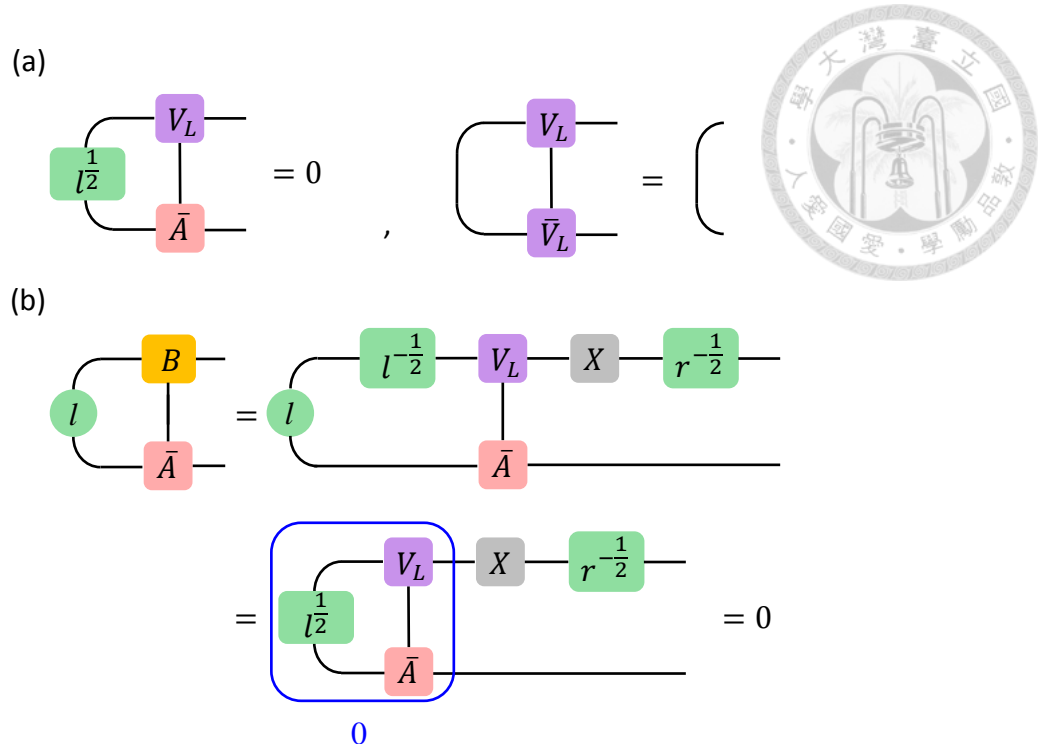


Figure 5.2: (a) The properties of  $V_L$  (b) The properties of tensor  $B$

### 5.3 Projection Operator

As Fig. 5.1 shows, we need to find the projection operator  $P_{|\psi(A)\rangle}$  such that  $P_{|\psi(A)\rangle}|\Theta\rangle = |\Phi(B; A)\rangle$ . This operator projects an arbitrary quantum state  $|\Theta\rangle$  onto the tangent plane of the manifold. In other words, we want to find the tensor  $B$ , which is a function of  $X$ , that minimizes

$$\| |\Theta\rangle - |\Phi(B(X); A)\rangle \|_2. \quad (5.6)$$

So the derivative with respect to  $\bar{X}$  should be zero:

$$\partial_{\bar{X}} \| |\Theta\rangle - |\Phi(B; A)\rangle \|_2 = 0, \quad (5.7)$$

which implies

$$\begin{aligned} \partial_{\bar{X}} [ \langle \Theta | \Theta \rangle - \langle \Theta | \Phi(B(X); A) \rangle - \langle \Phi(B(\bar{X}); A) | \Theta \rangle \\ + \langle \Phi(B(\bar{X}); A) | \Phi(B(X); A) \rangle ] = 0. \end{aligned} \quad (5.8)$$

Finally, we get

$$\partial_{\bar{X}} \langle \Phi(B(\bar{X}); A) | \Phi(B(X); A) \rangle = \partial_{\bar{X}} \langle \Phi(B(\bar{X}); A) | \Theta \rangle. \quad (5.9)$$

Since we must satisfy Eq. (5.3), the inner product of tangent vector state becomes



$$\begin{aligned}
 & \langle \Phi(B; A) | \Phi(B; A) \rangle \\
 &= \sum_{i,j} \langle l | \dots \text{---} \begin{array}{c} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{B} \text{---} \boxed{A} \text{---} \dots \\ | \quad | \quad | \quad | \quad | \\ \boxed{\bar{A}} \text{---} \boxed{\bar{B}} \text{---} \boxed{\bar{A}} \text{---} \boxed{\bar{A}} \text{---} \boxed{\bar{A}} \text{---} \dots \end{array} \dots | r \rangle \\
 &= \mathbb{Z} \langle l | \dots \text{---} \begin{array}{c} \boxed{A} \text{---} \boxed{A} \text{---} \boxed{B} \text{---} \boxed{A} \text{---} \boxed{A} \text{---} \dots \\ | \quad | \quad | \quad | \quad | \\ \boxed{\bar{A}} \text{---} \boxed{\bar{A}} \text{---} \boxed{\bar{B}} \text{---} \boxed{\bar{A}} \text{---} \boxed{\bar{A}} \text{---} \dots \end{array} \dots | r \rangle \\
 &= \mathbb{Z} \langle l | \begin{array}{c} \boxed{B} \\ | \\ \boxed{\bar{B}} \end{array} | r \rangle = \mathbb{Z} \langle l | \begin{array}{c} \boxed{l^{-\frac{1}{2}}} \text{---} \boxed{V_L} \text{---} \boxed{X} \text{---} \boxed{r^{-\frac{1}{2}}} \\ | \quad | \quad | \\ \boxed{l^{-\frac{1}{2}}} \text{---} \boxed{\bar{V}_L} \text{---} \boxed{\bar{X}} \text{---} \boxed{r^{-\frac{1}{2}}} \end{array} | r \rangle \\
 &= \mathbb{Z} \langle \begin{array}{c} \boxed{V_L} \text{---} \boxed{X} \\ | \quad | \\ \boxed{\bar{V}_L} \text{---} \boxed{\bar{X}} \end{array} \rangle = \mathbb{Z} \langle \boxed{X} \rangle . \tag{5.10}
 \end{aligned}$$

This implies that  $\partial_{\bar{X}} \langle \Phi(B(\bar{X}); A) | \Phi(B(X); A) \rangle = \mathbb{Z}X$ , hence  $\mathbb{Z}X = \partial_{\bar{X}} \langle \Phi(B(\bar{X}); A) | \Theta \rangle$ . We can describe this graphically:

$$\langle \boxed{X} \rangle = \dots \text{---} \begin{array}{c} \boxed{\bar{A}} \text{---} \boxed{\bar{A}} \text{---} \boxed{l^{-\frac{1}{2}}} \text{---} \boxed{\bar{V}_L} \\ | \quad | \quad | \quad | \\ \Theta \end{array} \text{---} \begin{array}{c} \boxed{r^{-\frac{1}{2}}} \text{---} \boxed{\bar{A}} \end{array} \dots \tag{5.11}$$

Then, the tangent vector becomes

$$\sum_n \dots \begin{array}{c} \boxed{\bar{A}} \text{---} \boxed{\bar{A}} \text{---} \boxed{l^{-\frac{1}{2}}} \text{---} \boxed{\bar{V}_L} \\ | \quad | \quad | \quad | \\ \Theta \end{array} \text{---} \begin{array}{c} \boxed{r^{-\frac{1}{2}}} \text{---} \boxed{\bar{A}} \\ | \quad | \\ \boxed{r^{-\frac{1}{2}}} \text{---} \boxed{A} \end{array} \dots \tag{5.12}$$

Since  $P_{|\psi(A)\rangle} |\Theta\rangle = |\Phi(B; A)\rangle$ , we can define the projection operator as

$$P_{|\psi(A)\rangle} = \sum_n \dots \quad (5.13)$$

Now, we can start to construct the time-dependent variational principle (TDVP) algorithm. Starting from Schrödinger equation:

$$i \frac{d}{dt} |\Psi(A(t))\rangle = \hat{H} |\Psi(A(t))\rangle, \quad (5.14)$$

we cannot represent exact quantum state as a uMPS with finite bond dimension, so we need to project the quantum state to tangent plane. Hence, the TDVP equation is:

$$i \frac{d}{dt} |\Psi(A(t))\rangle = P_{|\Psi(A)\rangle} \hat{H} |\Psi(A(t))\rangle \quad (5.15)$$

If we use a MPO to represent the Hamiltonian, the right hand side of above equation can be graphically represented as

$$P_{|\psi(A)\rangle} \hat{H} |\psi(A)\rangle = \sum_n \dots \quad (5.16)$$

By the chain rule, the left hand side of the Eq. (5.15) can be represented as

$$i \frac{d}{dt} |\psi(A)\rangle = \sum_n \dots \quad (5.17)$$

Comparing the LHS and RHS of Eq. (5.15), we will obtain

$$\dot{A}(t) = -i B, \quad (5.18)$$

where

(5.19)

Finally, we can solve the differential equation using the Runge-Kutta method. We can also use the technique in Chapter 4.1 to deal with the infinite boundary term by replacing tensor  $A_L$  and  $A_R$  with tensor  $A$ ; then  $L_6$  becomes  $l$ , which is the left eigenvector of the transfer matrix constructed by the tensor  $A$ , and  $R_6$  becomes  $r$ . Then  $B$  can be represented as

(5.20)

We have already found the TDVP equation represented by the tensor network diagram. So at next section, we are going to show the TDVP algorithm more concretely.

## 5.4 TDVP algorithm

Theoretically, we can use any gauge to simulate the tangent vector, but numerically, it is not the case. For instance, if we use left-canonical form for a normalized uMPS  $\Psi(A)$ , then the right eigenvector  $r$  of the transfer matrix will contain the square of Schmidt coefficients since

(5.21)

It is therefore ill-conditioned and the many operations with  $r^{-\frac{1}{2}}$  can produce large numerical errors. Similarly, if we choose right-canonical form,  $l^{-\frac{1}{2}}$  will become very large. Hence, we should use symmetric gauge for tensor  $A$  to avoid this problem. If we want to use TDVP algorithm to perform imaginary time evolution, we just need to use the simple Euler method to solve the differential equation; namely,

$$A(\tau + d\tau) = A(\tau) - Bd\tau \quad (5.22)$$

If we define the function  $f$ , which will return the tensor  $B$  for a given tensor  $A$ , ( $B = f(A)$ ). Then we can use the simple Euler method to find the ground state as following:

1. Change uMPS  $\Psi(A)$  to symmetric gauge.
2. Find tensor  $B = f(A)$  as we show in this chapter.
3. Set  $A(\tau + d\tau) = A(\tau) - Bd\tau$ . Then return to step 1.

Repeat the above procedure until the length of tangent vector  $\|B\|_2$  is small enough. If we want to do real-time evolution, Runge-Kutta method (RK4) is needed, whose error is  $O(dt^4)$ . This algorithm works as follows:

1. Change uMPS  $\Psi(A)$  to the symmetric gauge.
2. Set tensor  $B_1 = f(A)$ ,  $A_1 = A - i\frac{1}{2}B_1dt$
3. Set tensor  $B_2 = f(A_1)$ ,  $A_2 = A - i\frac{1}{2}B_2dt$
4. Set tensor  $B_3 = f(A_2)$ ,  $A_3 = A - iB_3dt$
5. Set tensor  $B_4 = f(A_3)$ ,
6.  $A(t + dt) = A(t) - i\frac{1}{6}(B_1 + 2B_2 + 2B_3 + B_4)dt$
7. Return to step 1.

Note that we cannot change the gauge for uMPS until whole RK4 process is done.



# Chapter 6

## Result and Conclusion

### 6.1 Ground State of the Thirring Model

We use VUMPS algorithm to simulate the ground state of the Thirring model on infinite 1D lattice with the constraint  $\langle S_z \rangle = 0$ . We show our results (with bond dimension  $D=25$ ) in Figs.(6.1~6.3). The Thirring model maps to the classical 2D XY model as we discuss in section 2.3. We can see that (Fig. 6.2) the entanglement entropy of quantum state in KT phase is very large.

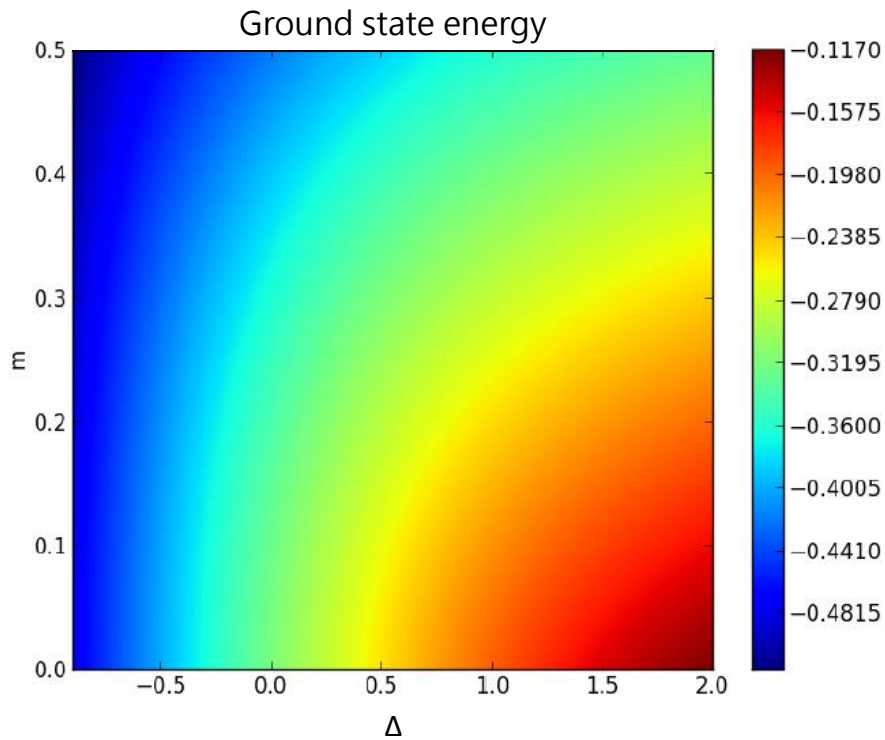


Figure 6.1: Energy density of the Thirring model.

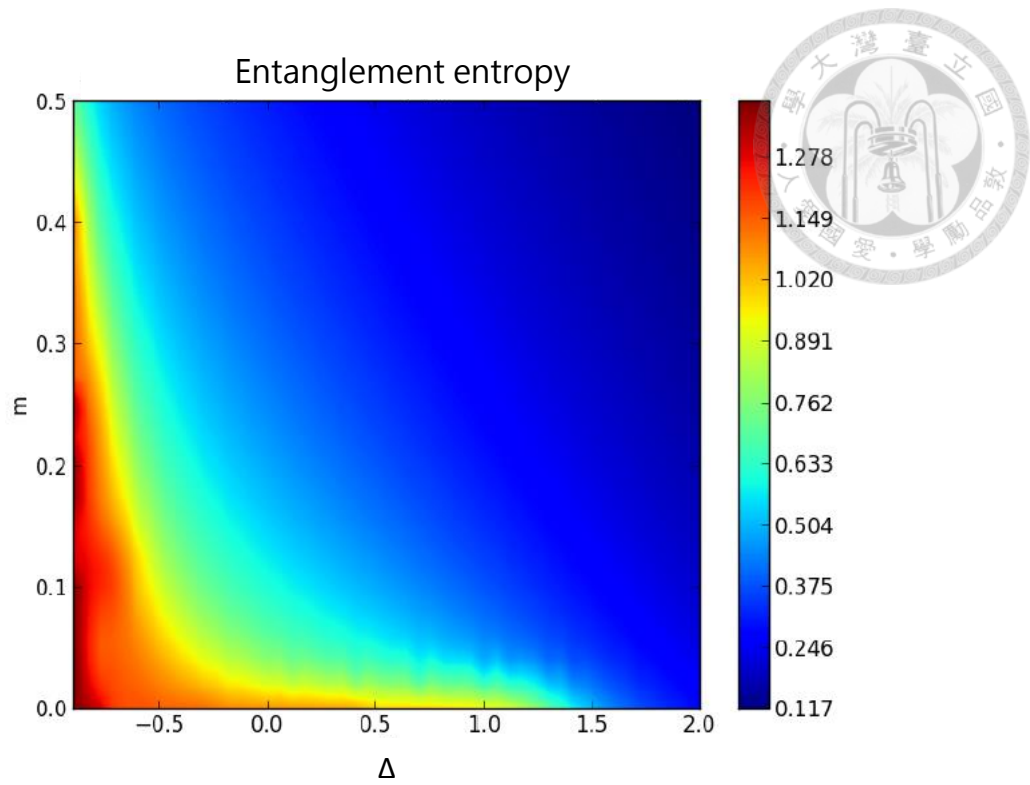


Figure 6.2: Entanglement entropy of the Thirring model.

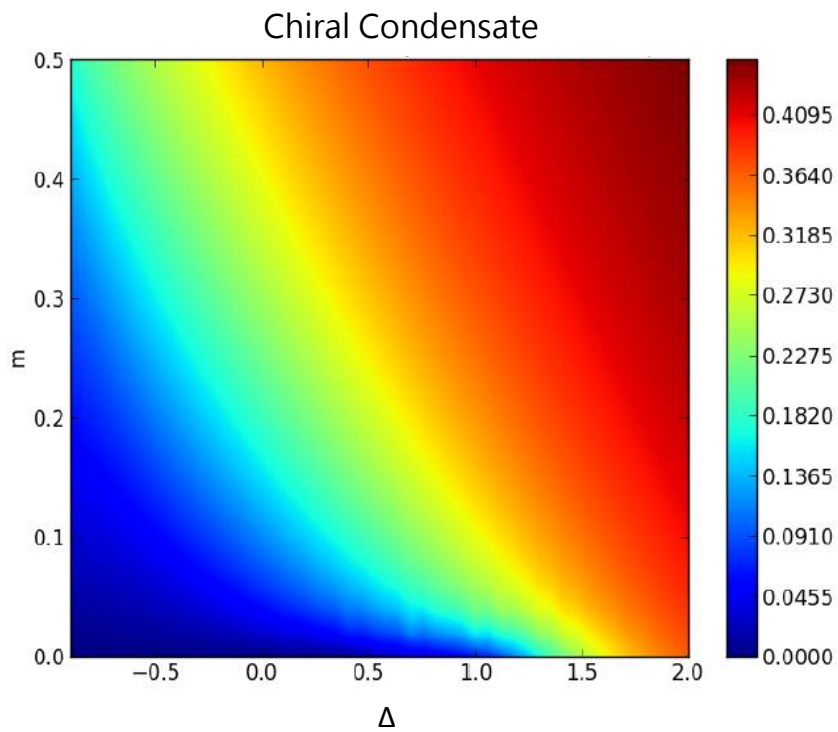


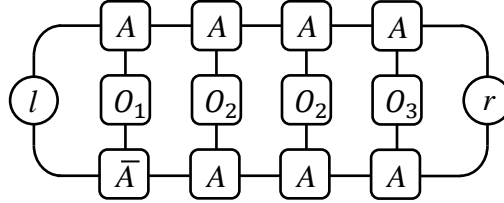
Figure 6.3: Chiral Condensate of the Thirring model.



The fermion correlator can be represented in spin representation as

$$G(r) = S_n^+ e^{(i\pi \sum_{j=n+1}^{n+r-1} S_j^z)} S_{n+r}^i \quad (6.1)$$

For example, if  $r = 8$ , then  $G(r = 3)$  can be represented in tensor network diagram as:



where  $O_1 = S^+ \otimes e^{i\pi S^z}$ ,  $O_2 = e^{i\pi S^z} \otimes e^{i\pi S^z}$ ,  $O_3 = e^{i\pi S^z} \otimes S^-$ . Figs. (6.4~6.10) show the result for fermion correlator with bond dimension  $D=100$ . We can observe that for the massive case ( $m = 0.2$ ), the fermion correlator exhibits power-law decay in the KT phase and exponential decay in the other phase. For the massless case ( $m = 0$ ), the fermion correlator exhibits power-law decay for any  $\Delta$ .

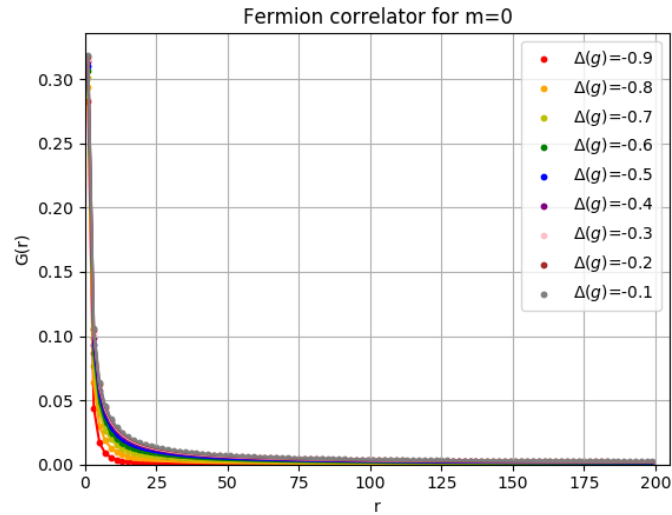


Figure 6.4: Fermion correlator for the massless case on a linear scale.

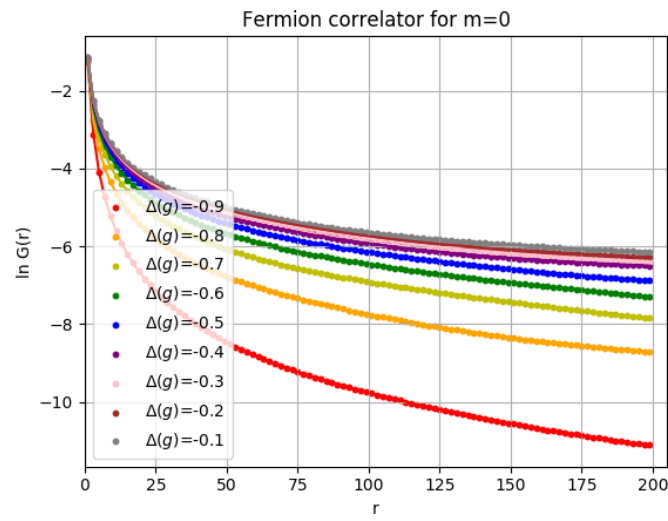


Figure 6.5: Fermion correlator for the massless case with a semi-log scale.

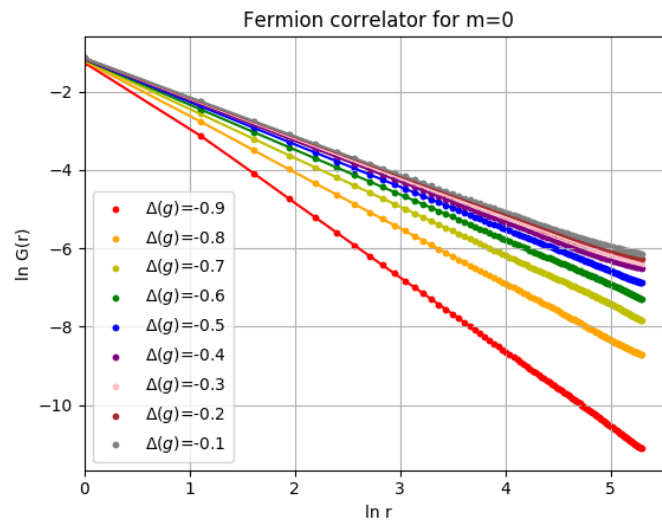


Figure 6.6: Fermion correlator for the massless case on a log-log scale.

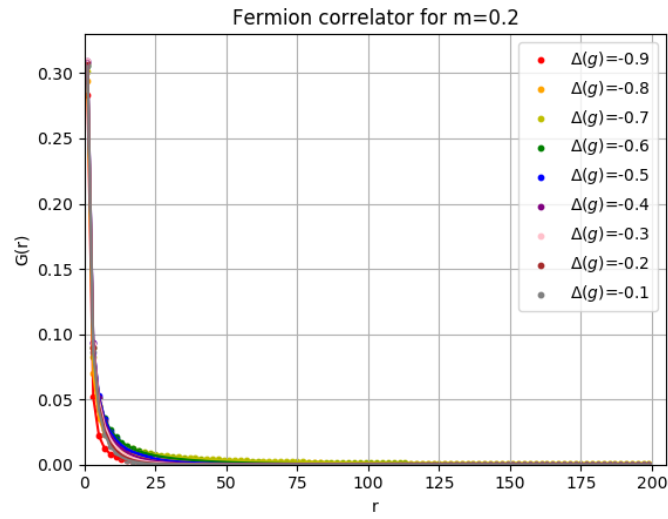


Figure 6.7: Fermion correlator for the massive case on a linear scale for  $m=0.2$ .

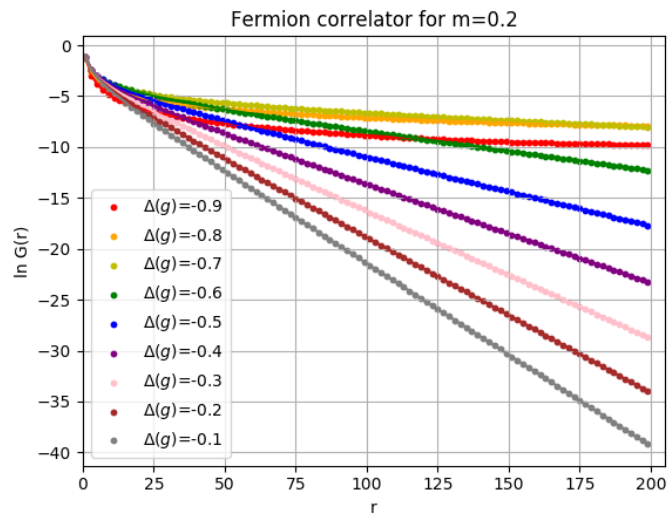


Figure 6.8: Fermion correlator for the massive case with a semi-log scale for  $m=0.2$ .

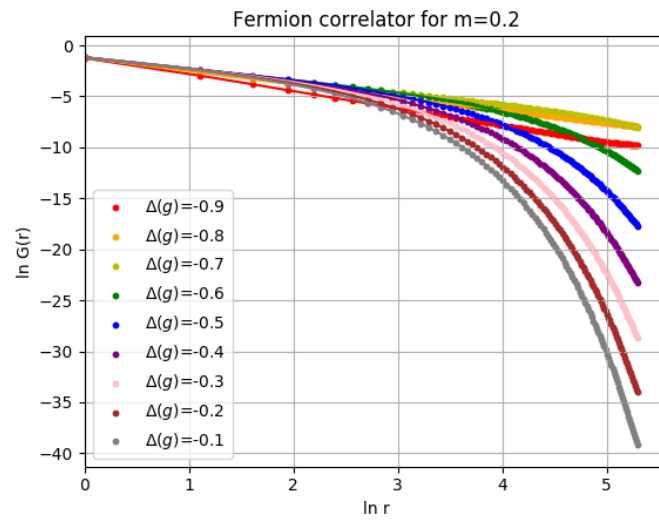


Figure 6.9: Fermion correlator for the massive case on a log-log scale for  $m=0.2$ .

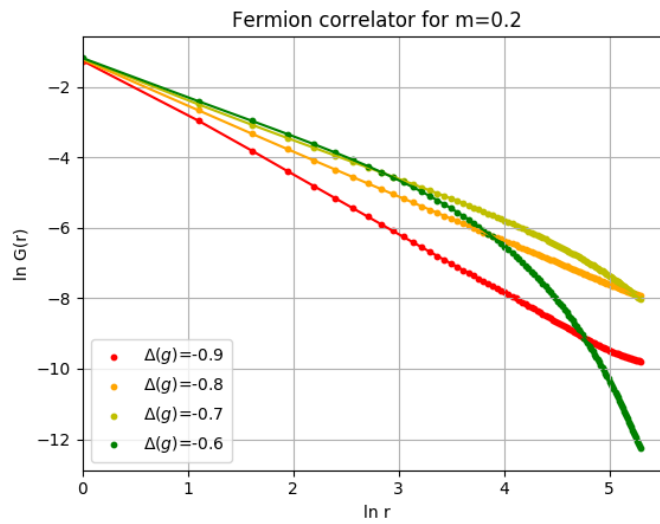
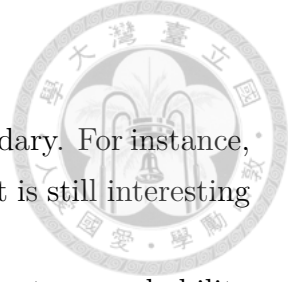


Figure 6.10:  $\Delta(g)$  near the transition.



## 6.2 TDVP Result

The most interesting subject is a quench across the phase boundary. For instance, from the KT phase to the other phase. Even in the same phase, it is still interesting to investigate the dynamics of the Thirring model.

The most important observable for the real time evolution is return probability (Loschmidt echo) [18] defined by

$$L(t) = |\langle \psi(0) | \psi(t) \rangle|^2 = |\langle \psi(0) | e^{-iHt} | \psi(0) \rangle|^2. \quad (6.2)$$

However, for an infinite 1D system, we cannot calculate this observable exactly since it will always be zero for  $t > 0$ . Instead, we use the norm square of the dominant eigenvalue of the transfer matrix,  $E(t) = \sum_s A(t)^s \otimes \bar{A}^s(0)$ , arising from the overlap between the initial state and the time-evolved state at time  $t$  to represent the return probability (denoted as  $P(t)$ ). Another important observable is the return rate function defined by:

$$g(t) = - \lim_{N \rightarrow \infty} \frac{1}{N} \ln L(t), \quad (6.3)$$

which is well-defined even in the thermodynamic limit, and

$$\begin{aligned} L(t) &\approx P^N(t) \\ \Rightarrow g(t) &= - \lim_{N \rightarrow \infty} \frac{1}{N} N \ln P(t) = - \ln P(t). \end{aligned} \quad (6.4)$$

We can represent the return-rate function with the negative logarithm of the dominant eigenvalue of the transfer matrix.

In this thesis, we investigate the dynamics of the Thirring model by TDVP algorithm with bond dimension  $D=100$  without a penalty term. Note that  $\langle S_z \rangle$  and  $\langle H \rangle$  are conserved quantities since  $[H, H] = [S_z, H] = 0$ . We can see that  $\langle S_z \rangle$  is conserved in our simulation (Fig. 6.11) even though we turned off the penalty term. We found that the simulation is very unstable if we turn on the penalty term.

Now let us examine the case starting from the ground state with the parameters  $(\Delta, m) = (-0.8, 0.2)$  and use the Hamiltonian with parameters  $(\Delta, m) = (0.5, 0.2)$  to evolve the state. In Fig. 6.11, we observe that the entanglement entropy saturates after  $t \approx 15$  since the quantum state has evolved to a state far away from the uMPS manifold. So we can only trust the result before  $t = 15$ .

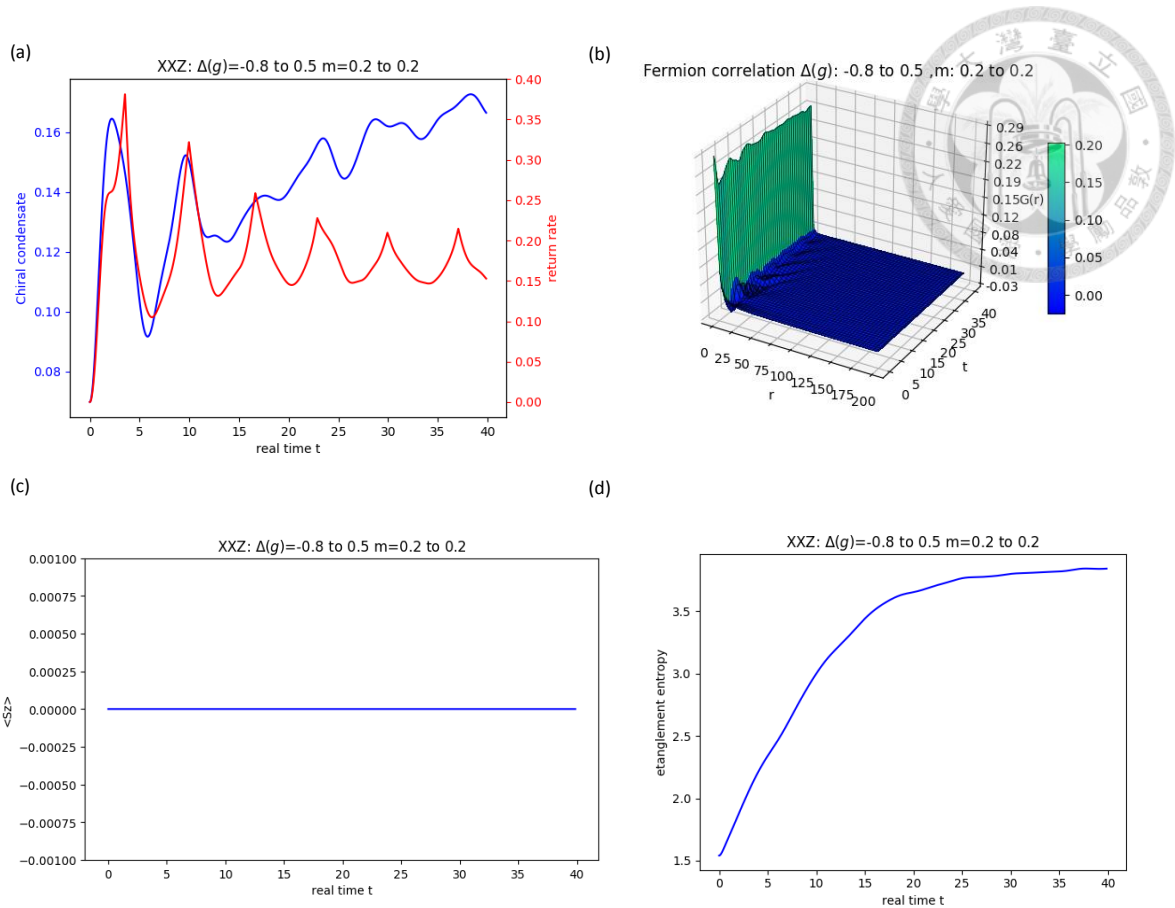


Figure 6.11: The Thirring model evolving from  $(\Delta, m) = (-0.8, 0.2)$  to  $(\Delta, m) = (0.5, 0.2)$ .

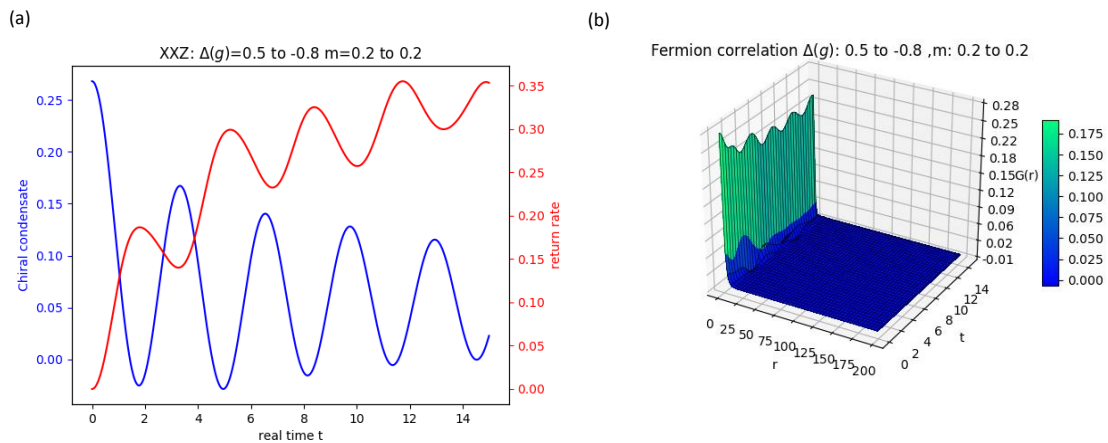


Figure 6.12: The Thirring model evolving from  $(\Delta, m) = (0.5, 0.2)$  to  $(\Delta, m) = (-0.8, 0.2)$ .

We can present results of a quench on  $\Delta - m$  plane and we show the dynamics of the chiral condensate, the return probability and the fermion correlator. More

numerical results are shown in the Appendix. We observe that when quenching from the KT phase with the same mass ( $m \neq 0$ ), there exist several non-analytic cusps (see Fig. 6.11) of the return-rate function which indicates that there may be a dynamical phase transition. Otherwise, the return-rate function evolving with the same mass will be very smooth (see Fig. 6.12). We obtain numerical results for the quench dynamics of the Thirring model. It may exist dynamical phase transition for some cases. Further exploration is necessary.



# Chapter 7

## Summary

Tensor network methods are a powerful tool for studying many-body systems and there are several efficient algorithms available for 1D systems. After discretizing the Thirring model on a 1D infinite lattice, we represent it as a spin-1/2 representation. We use a tensor network method to find its ground state and to study time evolution.

The time-dependent variational principle algorithm (TDVP) and the variational optimization methods for uniform matrix product state (VUMPS) are very efficient and accurate. In this thesis, we use the VUMPS algorithm to find the ground state of the Thirring model and characterize the phase diagram. We then extend the TDVP algorithm in MPO form such that we can deal with the Thirring model problem and investigate the quench dynamics. We can see that TDVP algorithm preserves conserved quantities very well.

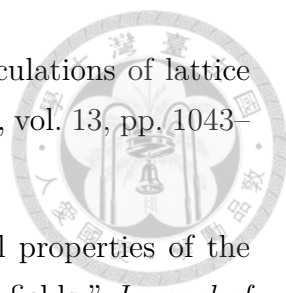
We want to ask whether the Thirring model exists the dynamical phase transition. So we use TDVP algorithm to do real-time evolution and see the return-rate function. And we found the existence of non-analytic cusps in the return-rate function which suggest the existence of the dynamical phase transition. We still wonder what is the physical meaning of the dynamics of the fermion correlator for the Thirring model and it is worthwhile to research it in the future.





# Bibliography

- [1] R. Orús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states,” *Annals of Physics*, vol. 349, pp. 117 – 158, 2014.
- [2] U. Schollwöck, “The density-matrix renormalization group in the age of matrix product states,” *Annals of Physics*, vol. 326, no. 1, pp. 96 – 192, 2011. January 2011 Special Issue.
- [3] S. R. White, “Density matrix formulation for quantum renormalization groups,” *Phys. Rev. Lett.*, vol. 69, pp. 2863–2866, Nov 1992.
- [4] G. Vidal, “Classical Simulation of Infinite-Size Quantum Lattice Systems in One Spatial Dimension,” *Phys. Rev. Lett.*, vol. 98, p. 070201, Feb 2007.
- [5] V. Zauner-Stauber, L. Vanderstraeten, M. T. Fishman, F. Verstraete, and J. Haegeman, “Variational optimization algorithms for uniform matrix product states,” *Phys. Rev. B*, vol. 97, p. 045145, Jan 2018.
- [6] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pižorn, H. Verschelde, and F. Verstraete, “Time-Dependent Variational Principle for Quantum Lattices,” *Phys. Rev. Lett.*, vol. 107, p. 070601, Aug 2011.
- [7] H. N. Phien, G. Vidal, and I. P. McCulloch, “Infinite boundary conditions for matrix product state calculations,” *Phys. Rev. B*, vol. 86, p. 245107, Dec 2012.
- [8] M. Heyl, “Dynamical quantum phase transitions: a review,” *Reports on Progress in Physics*, vol. 81, no. 5, p. 054001, 2018.
- [9] Mari Carmen Bañuls, Krzysztof Cichy , Ying-Jer Kao, C.-J. David Lin, Yu-Ping Lin, and David Tao-Lin Tan , “Tensor Network study of the (1+1)-dimensional Thirring Model,” *EPJ Web Conf.*, vol. 175, p. 11017, 2018.

- 
- [10] T. Banks, L. Susskind, and J. Kogut, “Strong-coupling calculations of lattice gauge theories:  $(1 + 1)$ -dimensional exercises,” *Phys. Rev. D*, vol. 13, pp. 1043–1053, Feb 1976.
- [11] F. C. Alcaraz and A. L. Malvezzi, “Critical and off-critical properties of the XXZ chain in external homogeneous and staggered magnetic fields,” *Journal of Physics A: Mathematical and General*, vol. 28, no. 6, p. 1521, 1995.
- [12] M. Bañuls, K. Cichy, J. Cirac, and K. Jansen, “The mass spectrum of the Schwinger model with matrix product states,” *Journal of High Energy Physics*, vol. 2013, p. 158, Nov 2013.
- [13] J. M. Kosterlitz and D. J. Thouless, “Ordering, metastability and phase transitions in two-dimensional systems,” *Journal of Physics C: Solid State Physics*, vol. 6, no. 7, p. 1181, 1973.
- [14] V. L. Berezinskiĭ, “Destruction of Long-range Order in One-dimensional and Two-dimensional Systems having a Continuous Symmetry Group I. Classical Systems,” *Soviet Journal of Experimental and Theoretical Physics*, vol. 32, p. 493, 1971.
- [15] R. Orús and G. Vidal, “Infinite time-evolving block decimation algorithm beyond unitary evolution,” *Phys. Rev. B*, vol. 78, p. 155117, Oct 2008.
- [16] H. van der Vorst, “Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems,” *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 2, pp. 631–644, 1992.
- [17] R. Bhatia, *Matrix Analysis*. Springer-Verlag, 1997. Thm. IX.7.2,R.
- [18] J. C. Halimeh and V. Zauner-Stauber, “Dynamical phase diagram of quantum spin chains with long-range interactions,” *Phys. Rev. B*, vol. 96, p. 134427, Oct 2017.



# Appendix

## More Numerical Results

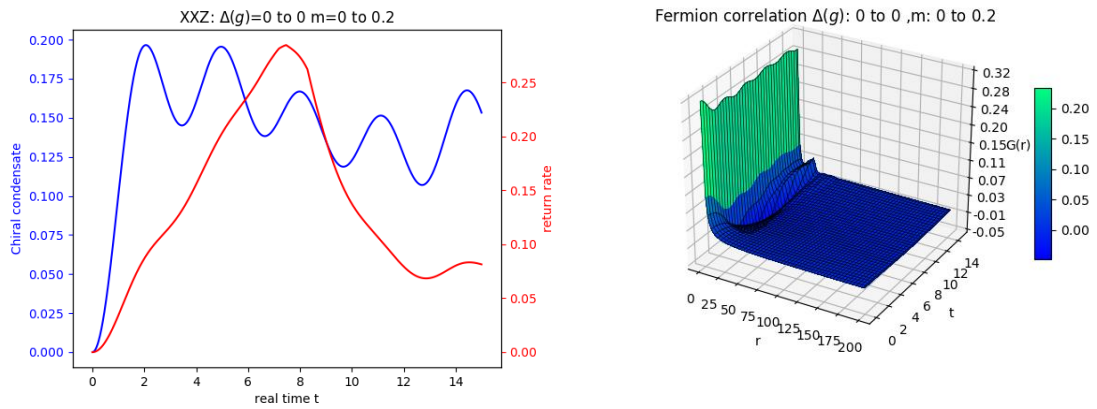


Figure A.1: The Thirring model evolving from  $(\Delta, m) = (0, 0)$  to  $(\Delta, m) = (0, 0.2)$ .

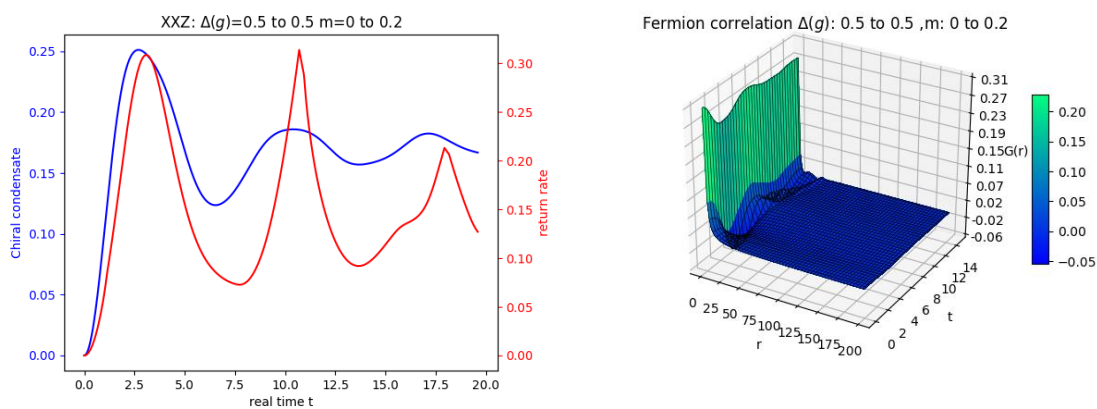


Figure A.2: The Thirring model evolving from  $(\Delta, m) = (0.5, 0)$  to  $(\Delta, m) = (0.5, 0.2)$ .

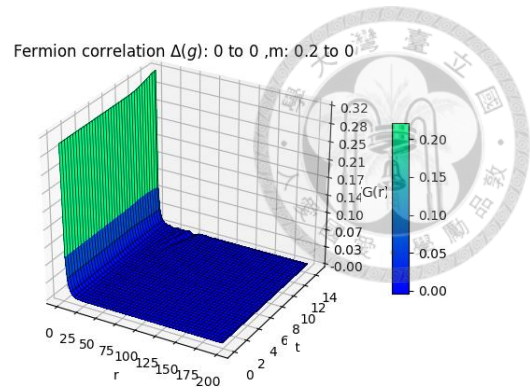
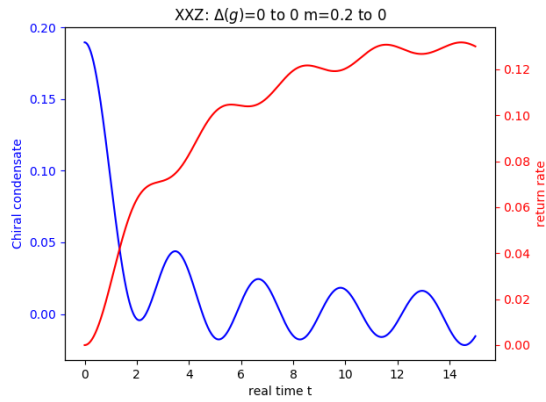


Figure A.3: The Thirring model evolving from  $(\Delta, m)=(0, 0.2)$  to  $(\Delta, m)=(0, 0)$ .

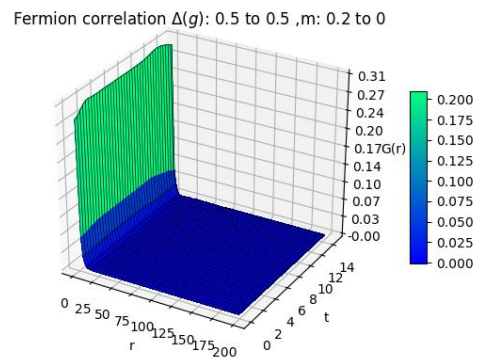
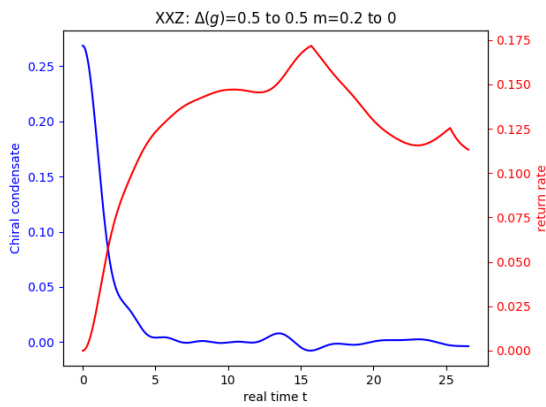


Figure A.4: The Thirring model evolving from  $(\Delta, m)=(0.5, 0.2)$  to  $(\Delta, m)=(0.5, 0)$ .

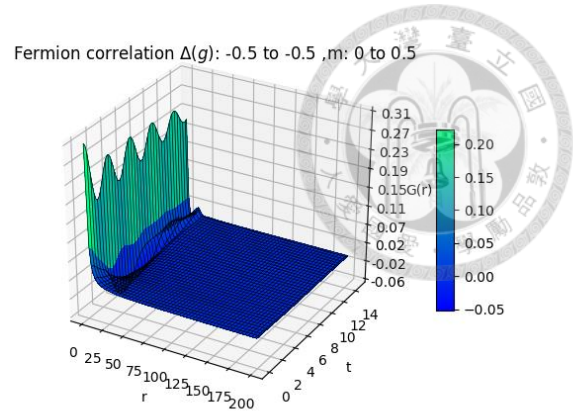
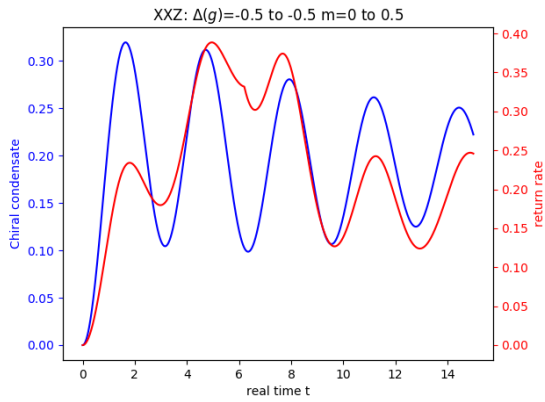


Figure A.5: The Thirring model evolving from  $(\Delta, m)=(-0.5, 0)$  to  $(\Delta, m)=(-0.5, 0.5)$ .

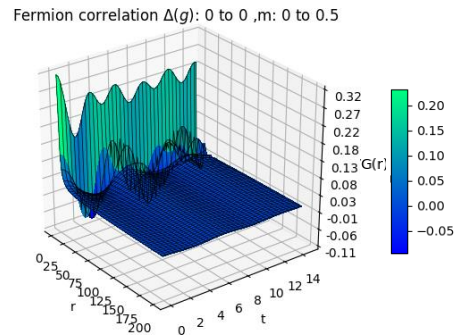
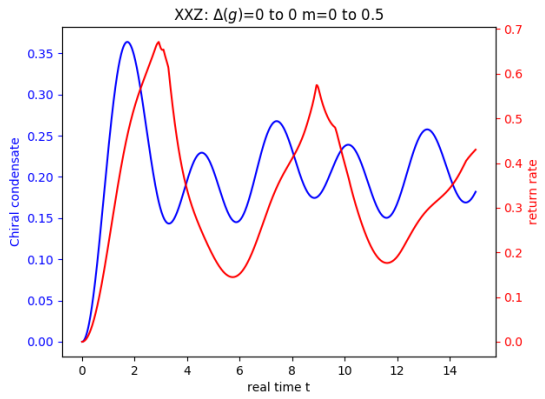


Figure A.6: The Thirring model evolving from  $(\Delta, m)=(0, 0)$  to  $(\Delta, m)=(0, 0.5)$ .

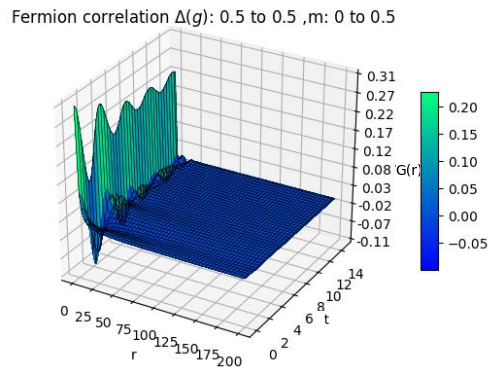
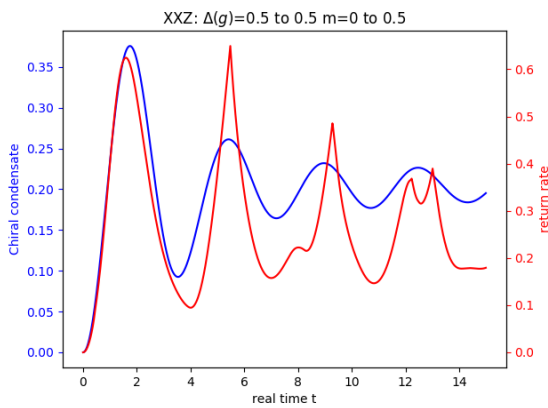


Figure A.7: The Thirring model evolving from  $(\Delta, m)=(0.5, 0)$  to  $(\Delta, m)=(0.5, 0.5)$ .

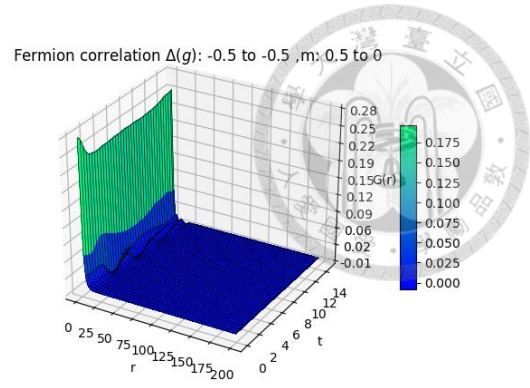
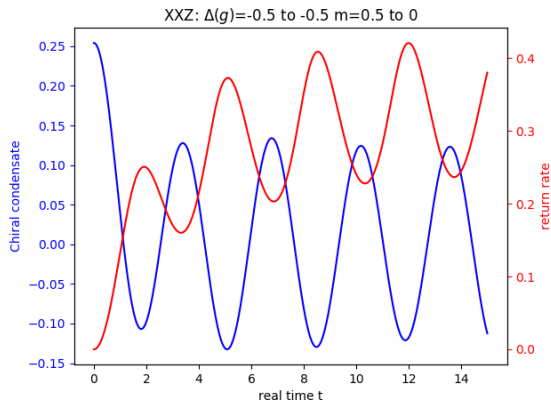


Figure A.8: The Thirring model evolving from  $(\Delta, m) = (-0.5, 0.5)$  to  $(\Delta, m) = (-0.5, 0)$ .

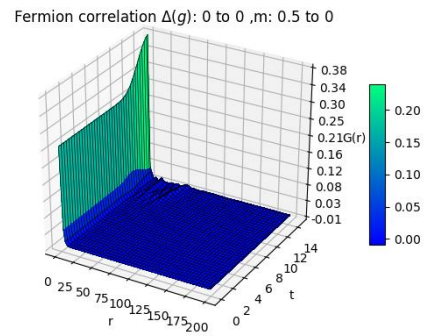
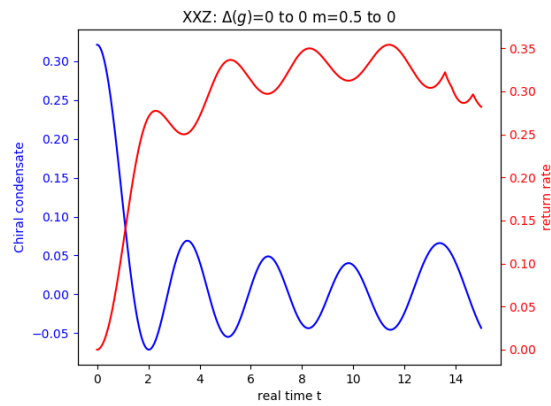


Figure A.9: The Thirring model evolving from  $(\Delta, m) = (0, 0.5)$  to  $(\Delta, m) = (0, 0)$ .

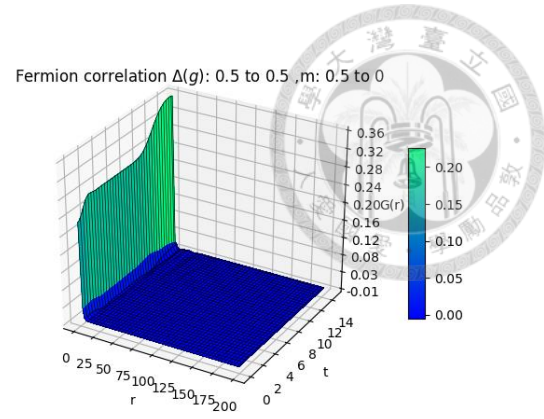
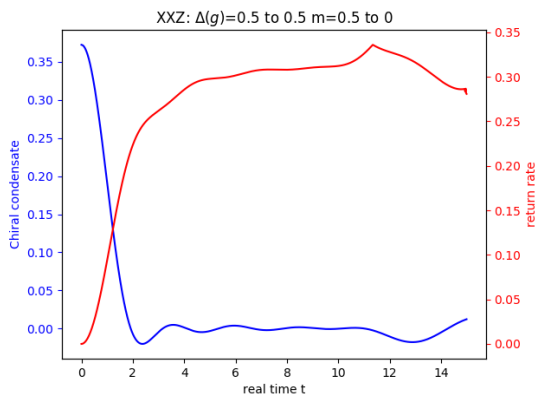


Figure A.10: The Thirring model evolving from  $(\Delta, m)=(0.5, 0.5)$  to  $(\Delta, m)=(0.5, 0)$ .

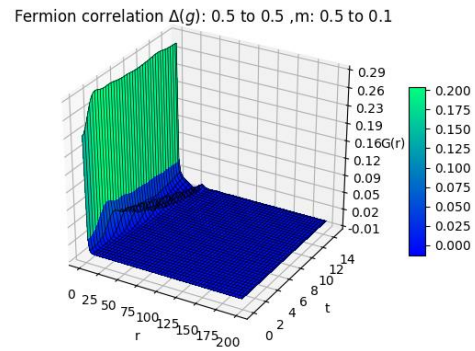
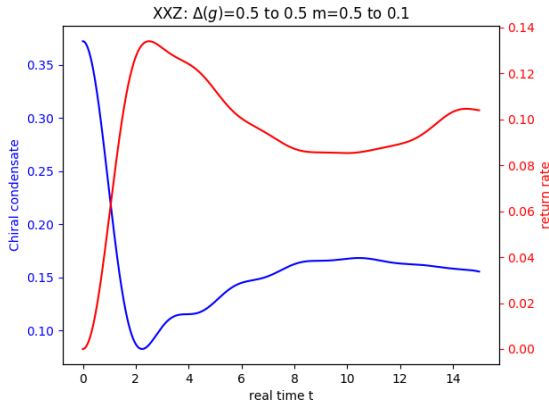


Figure A.11: The Thirring model evolving from  $(\Delta, m)=(0.5, 0.5)$  to  $(\Delta, m)=(0.5, 0.1)$ .

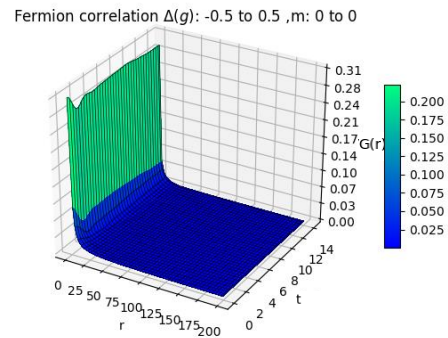
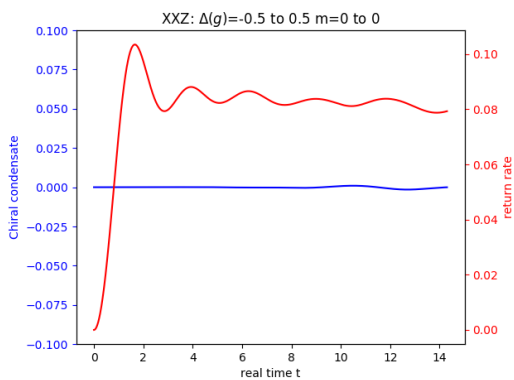


Figure A.12: The Thirring model evolving from  $(\Delta, m)=(-0.5, 0)$  to  $(\Delta, m)=(0.5, 0)$ .



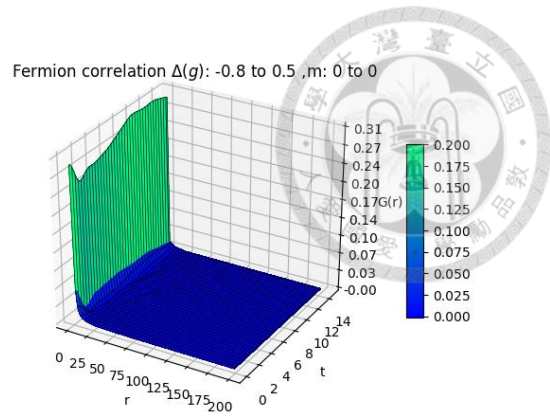
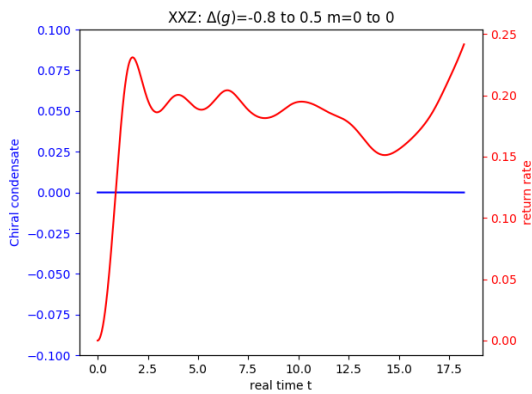


Figure A.13: The Thirring model evolving from  $(\Delta, m) = (-0.8, 0)$  to  $(\Delta, m) = (0.5, 0)$ .

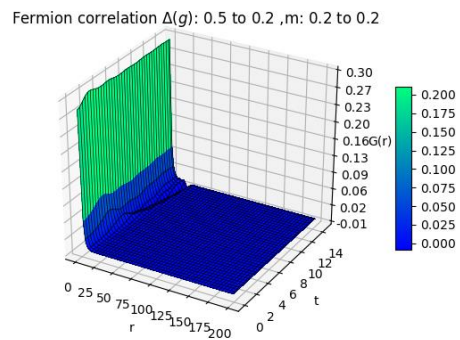
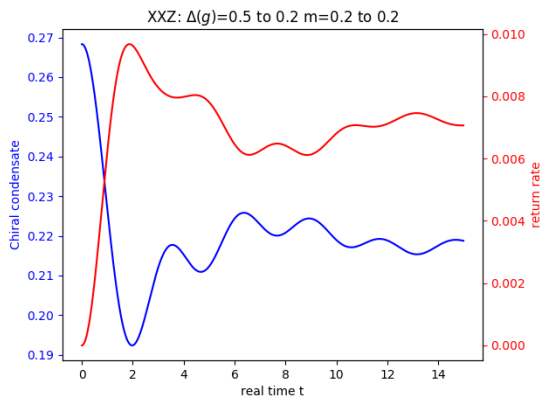


Figure A.14: The Thirring model evolving from  $(\Delta, m) = (0.5, 0.2)$  to  $(\Delta, m) = (0.2, 0.2)$ .

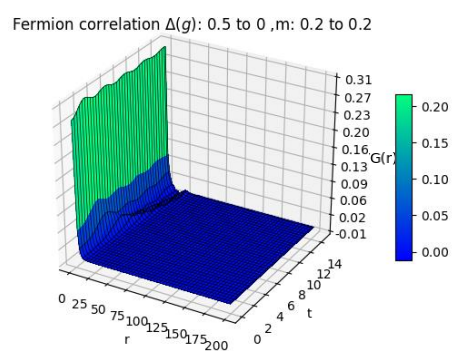
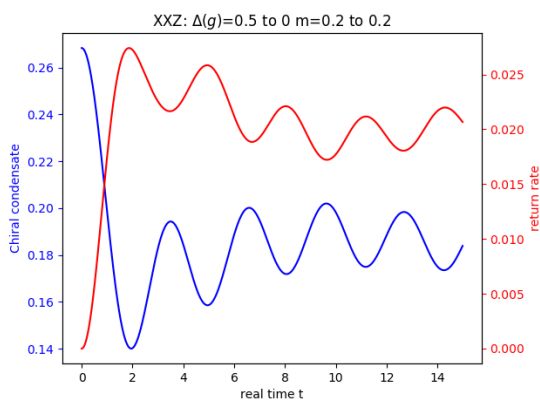


Figure A.15: The Thirring model evolving from  $(\Delta, m) = (0.5, 0.2)$  to  $(\Delta, m) = (0, 0.2)$ .



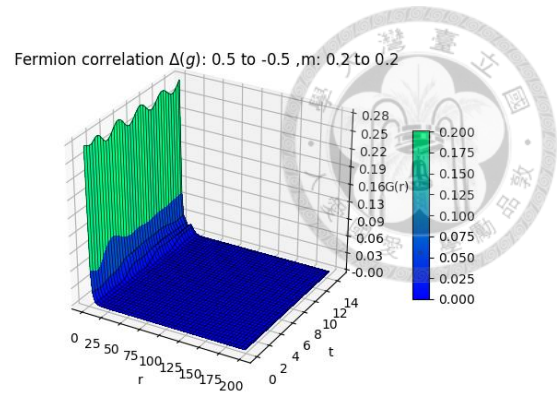
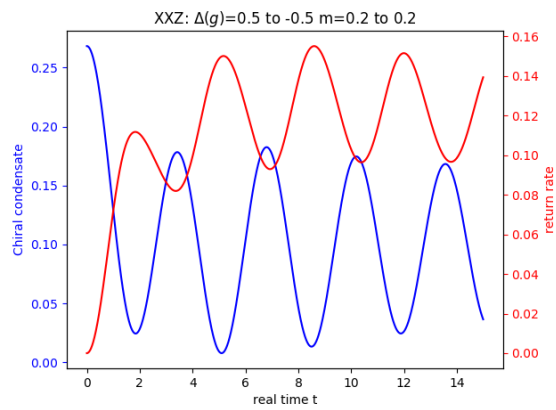


Figure A.16: The Thirring model evolving from  $(\Delta, m)=(0.5, 0.2)$  to  $(\Delta, m)=(-0.5, 0.2)$ .

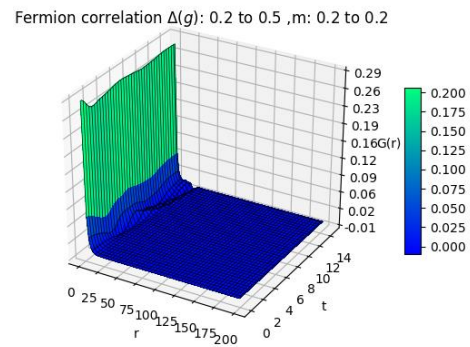
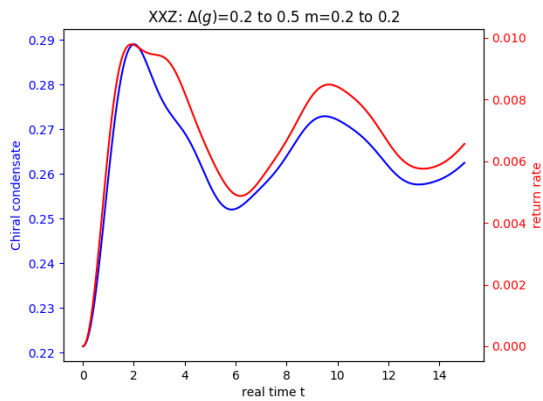


Figure A.17: The Thirring model evolving from  $(\Delta, m)=(0.2, 0.2)$  to  $(\Delta, m)=(0.5, 0.2)$ .

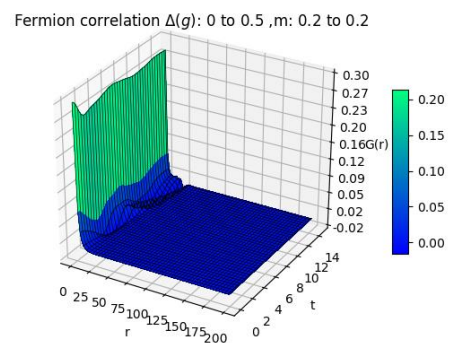
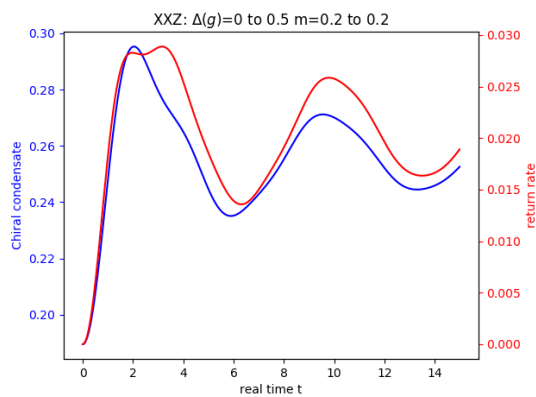


Figure A.18: The Thirring model evolving from  $(\Delta, m)=(0, 0.2)$  to  $(\Delta, m)=(0.5, 0.2)$ .

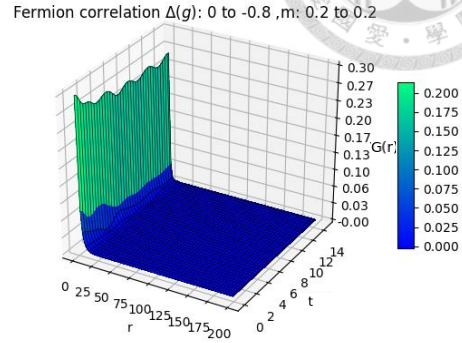
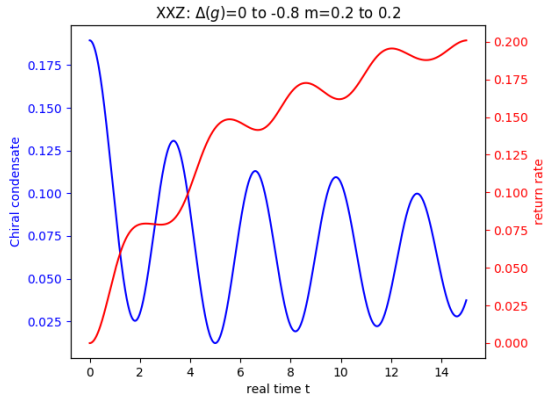
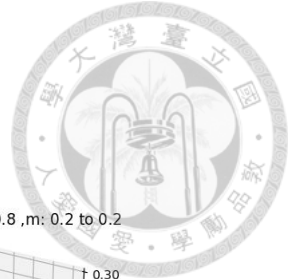


Figure A.19: The Thirring model evolving from  $(\Delta, m)=(0, 0.2)$  to  $(\Delta, m)=(-0.8, 0.2)$ .

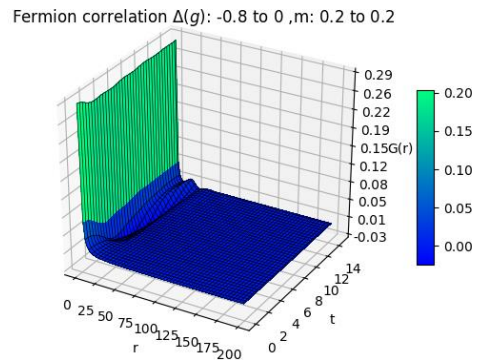
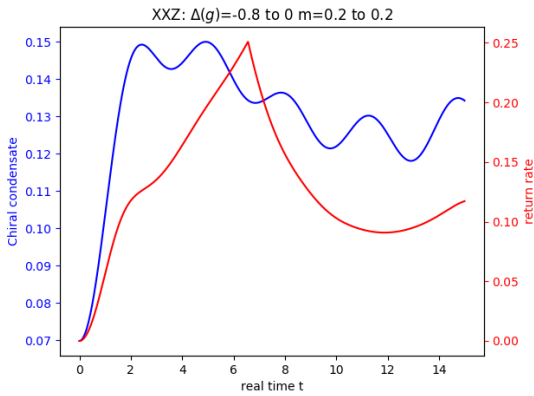


Figure A.20: The Thirring model evolving from  $(\Delta, m)=(-0.8, 0.2)$  to  $(\Delta, m)=(0, 0.2)$ .