Topologically Protected Transport in One-Dimensional Lattices of the SSH Type

National University of Singapore

Kurniawan Tjandra
Supervisor: A/P Dimitris Angelakis
Co-supervisor: Prof.Gong Jiangbin

5th April 2019

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in this thesis. This thesis has also not been submitted for any degree in any university previously.

Kurniawan Tjandra
April 2019

Acknowledgements

I thank A/P Dimitris Angelakis for the opportunity to do this project and for his guidance and ideas which is the basis of this project.

I thank Marc-Antoine Lemonde for his generous help, meaningful discussions, and guidance throughout the entirety of this project.

I thank Isabela Desita for lending me her computer's computing power for running codes that took overnight to complete, for being very supportive while I write this thesis, and for her prayers.

Abstract

Topological physics is a topic that gathered immense interests in the quantum physics community since the first observation of the Quantum Hall effect in 1980 and still sparkles active research as the number of different platforms simulating topological phases grows. One of the most impactful consequences of topology is the possibility to perform quantum transport protected against local imperfections in the system.

In this report, I consider the simplest topological model for spinless fermionic excitations along a 1D chain, known as the SSH model. This model exhibits a topological phase where protected states localise at the edges of the finite chain. In this context, I investigate the adiabatic transfer of such single-excitation topologically protected edge states throughout the entire chain. My goal is to answer the crucial question: how robust to disorder along the chain the adiabatic state transfer is? Thus, we analyse the overall robustness to random set of disorders. Despite the different architectures proposed to simulate the SSH model and the adiabatic transport, fabrication imperfections can never be completely avoided. Having a good understanding of the effects of disorders in the SSH model and for the adiabatic transfer protocol is thus of primary interest for implementations.

In particular, we show that imperfections that locally break the crucial symmetry of the system or that locally break the topological phase have fundamentally different measurable impacts. To my knowledge this is a new point of view and it could help distinguish which type of defect is predominant in a physical implementation of the transport scheme.

Contents

Chapter 1

Introduction

One important achievement of quantum mechanics was the ability of band theory to explain electrical conductivity in most conductors and insulators. Band theory successfully explained why some crystalline materials are insulators even though electrons can hop between the different sites of the crystal lattice. The basic concept is that the allowed energy of the different electronic eigenstates $\omega(\vec{k})$, characterised by a quasimomentum ,$\vec{k}$, can form regions in frequencies with a high density of states, called bands, while leaving finite frequency ranges where no states are allowed, referred to as gaps. A consequence of the Pauli exclusion principle is that by adding (removing) electrons to the system fills up (depletes) the existing bands. A material is conducting when there is a partially filled band or when there is no energy gap between a fully filled band and an unoccupied band. On the other hand, a material is insulating when there is a fully filled band which is separated by a large energy gap from an unoccupied band.

After the quantum Hall effect (QHE) was first observed in 1980 by von Klitzing *et al.* [1], there was evidence that the band structures of materials alone was not sufficient to capture all of the physics. QHE is observed when a strong perpendicular magnetic field is applied to a two-dimensional electron gas. The magnetic field causes the electrons to be confined to localized states in the middle region of the material referred to as the bulk and thus produces a gapped band structure. While the gapped band structure suggests that the electron gas should behave as an insulator, precisely quantized conductivity was observed. It was quickly understood that electrons were remarkably allowed to move along the edges of the two dimensional sample, but not within the bulk as predicted by the band theory.

In 1982 D. J. Thouless *et al.* recognized the topological character of QHE [9]. The conducting channels along the edges in QHE arise from the non-trivial topology encoded in the eigenstates of the occupied bands of the electron gas. Since these eigenstates are delocalised over the whole system, the robustness of their topology, and thus of the quantised transport, against local disorders was pointed out. More precisely, it was shown that topological protection was effective against any defects that would conserve the crucial symmetry of the systems. The topology of eigenstates is characterized by a topological invariant, which in the case of QHE is known as the Chern number [3]. A very interesting relation that came out of those results is known as the bulk-boundary correspondence. That relation highlights the fact that while the topological invariant is a bulk property, i.e. calculated by only referring to the band eigenstates (formally defined in infinite systems), it encodes information about the edge states of the finite system. In other words a parameter defined purely from the bulk gives us information about states that exist only at the boundary or edges.

Since the discovery of QHE, intense efforts have led to a very active field in itself from which multiple theoretical models contributing to a deep understanding of topological physics have emerged. During this time, many topological insulating materials with peculiar edge states have been proposed and experimentally observed [3]. Even more remarkably, extensive works have been recently done for realising topological insulators using optical modes [13, 10, 11, 4].

In this project one of the simplest model of a topological insulator is studied: a one dimensional tight-binding model of non-interacting spinless fermions with only two different (alterned) hopping amplitudes, the Su-Schrieffer-Heeger (SSH) model[7, 8]. The SSH model was first proposed by W.P. Su, J. R. Schrieffer, and A. J. Heeger to explain electrons in polyacetylene, the simplest linear conjugated polymer. In contrast to the two dimensional QHE in the one-dimensional SSH, the edge states are not conducting but expected to be robust against symmetry-preserving defects.

In this study the edge states of the SSH are utilized to transport a quantum state from one end to the other end of the lattice. The final goal of this project is to answer the question of how robust or how protected the transport scheme is against random defects, and therefore what degree of precision is required for a physical implementation of such a transport scheme.

The report is organized as follow: In chapter 2, I present the SSH hamiltonian, its symmetry, band structure, topological phases, edge states, and the

transport scheme. In chapter 3, the robustness of the edge states of the static SSH Hamiltonian is studied. In chapter 4, the robustness of the transport scheme is studied against specific and random defects. Lastly, in chapter 5, I propose a physical implementation of the transport scheme. Chapter 2 are mostly literature review with results reproduced using my own codes (shown in the appendices), while chapter 3 onwards are new contributions.

Chapter 2

Edge States and Transport in the SSH Hamiltonian

This chapter serves as a foundation for the rest of the report as it covers the background of the project. An overview of the SSH Hamiltonian, highlighting its special topological character and crucial symmetry is presented and at the end of this chapter, a section is dedicated to the adiabatic transport that is being studied.

The SSH model was developed to descibe the dynamics of electrons in a polyacetylene molecule, which has alternating bond lengths.[7] In the tight-binding approximation, it reads,

$$\hat{H}_{SSH} = v \sum_{m=1}^{N} |m, B\rangle \langle m, A| + w \sum_{m=1}^{N-1} |m+1, A\rangle \langle m, B| + h.c. \tag{2.1}$$

Where v and w denotes the intracell and intercell hopping amplitudes respectively, $|m, A\rangle$ ($|m, B\rangle$) represents an electron being in the unit cell m and sublattice site A (B), and N denotes the number of unit cells in the lattice. Here, *h.c.* refers to the hermitian conjugate.

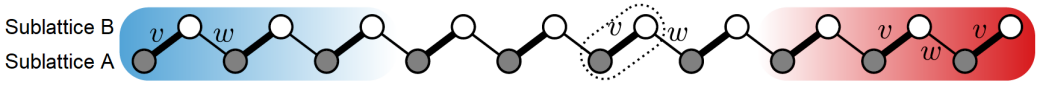


Figure 2.1: SSH lattice with 10 unit cells. *Picture taken from ref. [8]*

Fig. 2.1 shows a representation of an SSH lattice with 20 sites, i.e. 10 unit cells. Each site belongs to either one of the two sublattices, A (grey-filled

circles) or B (white-filled circles). As such, one unit cell contains two sites, one in sublattice A and one in sublattice B. The left (right) edge regions of the lattice is highlighted by the blue (red) background, while the bulk region is the region far from the edges with the white background.

2.1 Chiral Symmetry

The SSH Hamiltonian (2.1) has a crucial symmetry, which is known as chiral symmetry. Whereas the standard symmetry of a Hamiltonian under a unitary transformation, U, is often defined as

$$UHU^\dagger = H \tag{2.2}$$

chiral symmetry under a transformation, Γ, is defined as [8]

$$\Gamma H \Gamma^\dagger = -H \tag{2.3}$$

The chiral symmetry of the SSH Hamiltonian is also called the sublattice symmetry as the transformation under which the Hamiltonian is chiral is the subtractive combination of the projectors on sublattice A, P_A, and sublattice B, P_B,

$$\Gamma = P_A - P_B \tag{2.4}$$

where

$$P_A = \sum_{m=1}^{N} |m, A\rangle \langle m, A| \qquad \text{and} \qquad P_B = \sum_{m=1}^{N} |m, B\rangle \langle m, B| \tag{2.5}$$

Hence,

$$\Gamma = \sum_{m=1}^{N} |m, A\rangle \langle m, A| - \sum_{m=1}^{N} |m, B\rangle \langle m, B| \tag{2.6}$$

There are two important consequences of this chiral symmetry for the eigenstates and energies of the Hamiltonian. First, the energy spectrum of the SSH is always symmetric. For every eigenstate, $|\psi\rangle$, with energy E, there is always another eigenstate, its chiral partner, $\Gamma |\psi\rangle$ with energy $-E$. Second, states with zero energy can be decomposed such as it only occupy a single sublattice (as it is its own chiral partner, $|\psi_0\rangle = \Gamma |\psi_0\rangle$). Chiral symmetry is broken when an onsite potential is introduced to the Hamiltonian. This can be done globally, e.g. by introducing $u\Gamma$ (where u refers to an parameter), or locally, e.g. by introducing $u |i\rangle \langle i|$ (where $|i\rangle$ represents a state on the i^{th} site).

2.2 Energy Spectra

In this section let us turn to the spectra of the SSH Hamiltonian. In particular, we show that depending on the values of the intracell and intercell hopping amplitudes we get two distinct spectra. We also observe that the spectra are symmetric which was predicted by the chiral symmetry of the SSH Hamiltonian discussed in the section 2.1. In the one case ($v > w$), we only observe two energy bands, commonly referred to as the valence and the conduction bands. In the other case ($v < w$), we observe the apparition of zero-energy modes. Both of these spectra show insulating band structures in the sense that there is a finite energy gap between the two regions with high density of states (see fig. 2.2).

To show the energy spectra, the Hamiltonian for $N = 10$ is diagonalized and the eigenenergies are plotted against the ratio v/w in fig. 2.2 (this and all subsequent spectral plots are produced using the Matlab code presented in Appendix A). In this figure, we clearly see the zero-energy modes in the middle of the gap of the two energy bands for $v < w$, and the their disappearance when $v > w$.

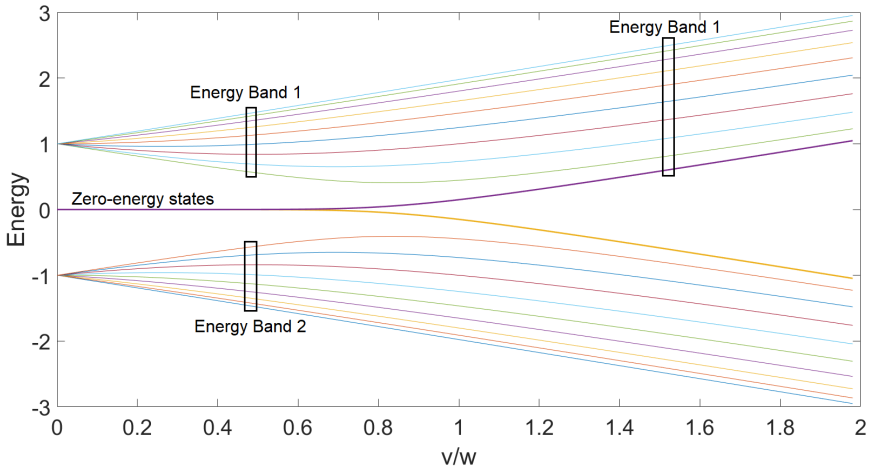


Figure 2.2: Energy spectra of the SSH Hamiltonian (for 10 unit cells) against v/w

When N is increased, we observe the same features but with an increase in the number of lines in the energy bands and a sharper transition between

the two distinct spectra becomes sharper. This is seen in fig.2.3 which shows the eigenenergies against v/w for an increased lattice size, $N = 40$.

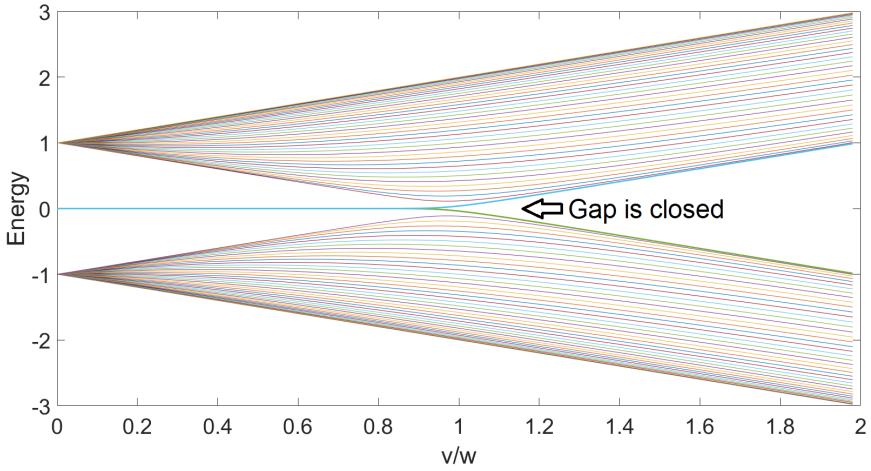


Figure 2.3: Energy spectra of the SSH Hamiltonian (for 40 unit cells) against intracell hopping amplitude, v, when $w = 1$

In the infinite size limit, $N \to \infty$, the two energy bands form a continuous band and the transition of the spectra from $v < w$ to $> w$ is sudden. Furthermore, when the transition happens at $v = w$, the energy gap between the two bands closes, leading to a band structure of a conductor.

2.3 Topological Invariant: Winding Number

The two cases $v < w$ and $v > w$ discussed in section 2.1 gives us two different phases of the SSH, the topological and the trivial phases respectively. These phases are parameterized by a topological invariant, which captures the topology encoded in the eigenstates. In contrast to the Chern number in the case of QHE mentioned in the introduction, the topological invariant of the SSH is known as the winding number, ν.

Since topological invariants are a bulk property, the definition of the winding number is formally expressed by either taking the infinite limit, $N \to \infty$, or adding a periodic boundary condition, such that hopping between the right edge to left edge is possible. If we do the latter, the finite SSH

Hamiltonian defined in equation (2.1) is changed to the bulk Hamiltonian given by,

$$H_{bulk} = v\sum_{m=1}^{N} |m,B\rangle\langle m,A| + w\sum_{m=1}^{N-1} |m+1,A\rangle\langle m,B| + h.c. \\ + w\,|1,A\rangle\langle N,B| + h.c. \quad (2.7)$$

where $w\,|1,A\rangle\langle N,B|$ represents the hopping from the right edge to the. The bulk Hamiltonian is invariant under discrete translation of a unit cell, thus allowing the use of the Bloch theorem [8] by decomposing the eigenstates over a basis of well-defined quasi-momentum, defined as

$$|k,u_n(k)\rangle = |k\rangle \otimes |u_n(k)\rangle \quad (2.8)$$

where,

$$|k\rangle = \frac{1}{\sqrt{N}}\sum_{m=1}^{N} e^{imk}\,|m\rangle \quad (2.9)$$

are the plane wave basis states and

$$|u_n(k)\rangle = a_n(k)\,|A\rangle + b_n(k)\,|B\rangle \quad (2.10)$$

are the eigenstates of the Fourier space Hamiltonian,

$$H(k) = \sum_{\alpha,\beta\in A,B} \langle k,\alpha|\,H_{bulk}\,|k,\beta\rangle\cdot|\alpha\rangle\langle\beta| \quad (2.11)$$

$H(k)$ now has a dimensionality of two and can be expressed as a dot product between a vector $\vec{d}(k)$ and Pauli vector $\vec{\sigma}$,

$$H(k) = \vec{d}(k)\cdot\vec{\sigma} \quad (2.12)$$

Finally, the winding number, ν, is defined from $d(\vec{k})$, as such,

$$\nu = \frac{1}{2\pi i}\int_0^{2\pi}\left(d(\vec{k})\times\frac{d}{dk}d(\vec{k})\right)_z dk \quad (2.13)$$

where z refers to the z component of the vector in the brackets. As evident in the definition of ν it is defined purely by considering the bulk of the SSH

model, i.e. by neglecting the edges. The winding number is an integer, either 1 (when $v < w$, topological phase) or 0 (when $v > w$, trivial phase). To change the winding number, we need to either close the gap between the energy band (see fig. 2.3) or break the chiral symmetry of the system, which is the crucial symmetry of the Hamiltonian

This topological invariant parameterizes the phases of the SSH, $\nu = 1$ and $\nu = 0$ corresponds to the topological phase and the trivial phase of the SSH respectively. Recalling from section 2.2, we now observe the correspondence between the winding number and the apparition of the zero-energy states. The zero-energy states appear only when $\nu = 1$.

2.4 Edge States

The correspondence between the zero-energy states and the winding number noted in section 2.3, like the correspondence between the Chern number and the number of conducting channels in the quantum Hall effect, is in fact a bulk-boundary correspondence since the winding number a bulk property and the zero-energy states are edge states of the SSH. We observe these edge states in the this chapter. The wavefunctions of the zero-energy and a finite-energy eigenstates over the 1D lattice of the SSH is plotted (using Matlab code presented in Appendix B) and compared in fig. 2.4.

We observe that both of the zero-energy eigenstates are localised to the edges of the lattice, unlike the finite-energy state which is spread over the bulk of the lattice. The zero-energy states can be resolved as such: $|+\rangle = 1/\sqrt{2}(|L\rangle + |R\rangle)$ and $|-\rangle = 1/\sqrt{2}(|L\rangle - |R\rangle)$. And so, the $|L\rangle$ ($|R\rangle$) state is localised about the left (right) edge and only occupy the A (B) sublattice. This matches what was predicted from the chiral symmetry as noted in section 2.1, zero-energy states can be resolved such that they occupy only one of the two sublattices. The $|L\rangle$ and $|R\rangle$ states are known as the left and the right edge states respectively. Whereas the finite-energy states are known as the bulk states. The edge states extend to the bulk with an exponentially decaying amplitude (of the wavefunction). More precisely, the extension of the edge states increases with increasing value of v/w. In the limit of $v/w = 0$, an edge state only occupy one single edge site. All of the observations noted in this section match the analytical treatment given in ref. [8].

In the case of a finite chain, the left (only occupying the A sublaticce)

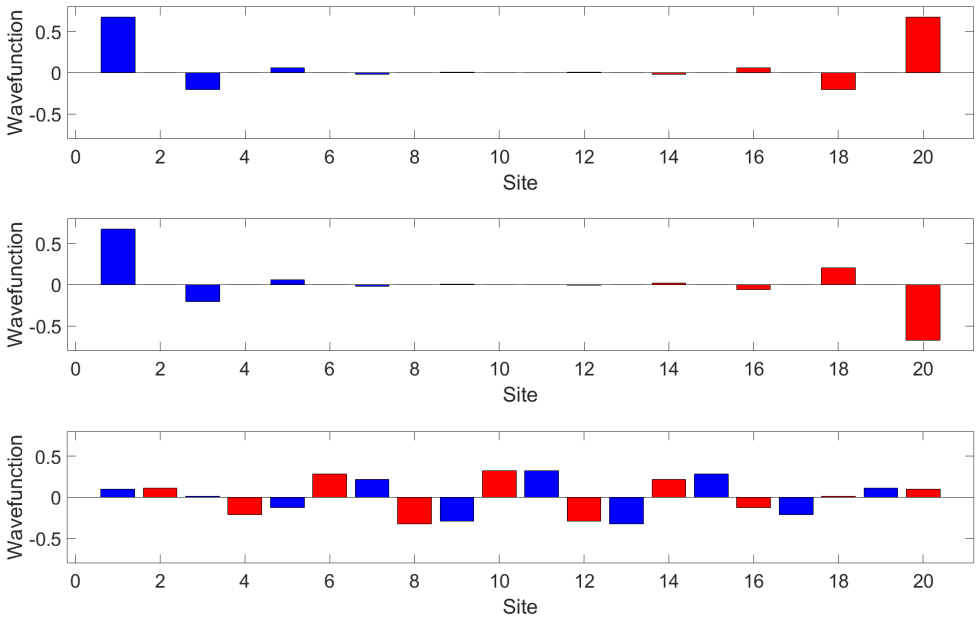


Figure 2.4: Wavefunctions of the zero-energy states $|+\rangle$ (a) $|-\rangle$ (b), and a finite-energy state

hybridizes with the right edge state (only occupying the B sublattice). This is due to the two edge states having the same energy and a small overlap in the bulk which allows them to couple via the hopping.

A remarkable feature is the persistence of the zero-energy states as long as the system remains in the topological phase, when $v < w$ or winding number is 1. Hence, the edge states are robust given a change in the v and w globally as long as $v < w$.

2.5 Adiabatic Transport

The goal of the transport scheme studied in this work is to adiabatically move a single excitation from one edge state, e.g. localised on the left boundary, to the opposite side of the chain. In this case, adiabatic means that the system follows the same eigenstate that smoothly evolves in time. For example, if we were to adiabatically vary the ratio v/w, the system would follow a single line in Fig. 2.2. In analogy with the well-known Thouless pumping

[9], a robust topological pumping described for insulating SSH systems, i.e. when all the valence band is filled with electrons, we consider a protocol that implies tuning the system in and out of its topological phase without closing the energy gap (i.e. $v \neq w$ at all time). As mentioned in section 2.3, the only other route available is to break the chiral symmetry. This is performed by allowing an additional onsite potential energy. Doing so, we consider the final time-dependant Hamiltonian:

$$\hat{H}(t) = v(t)\sum_{m=1}^{N}(|m,B\rangle\langle m,A| + h.c.) + w(t)\sum_{m=1}^{N-1}(|m+1,A\rangle\langle m,B| + h.c.) + u(t)\sum_{m=1}^{N}(|m,A\rangle\langle m,A| - |m,B\rangle\langle m,B|) \quad (2.14)$$

where u denotes the strength of onsite potential staggering. As mentioned in section 2.1, having a finite u is enough to break the chiral symmetry, thus allowing us to change topological phases while $v \neq w$.

More specifically, choosing

$$u(t) = -sin(\omega t) \quad (2.15)$$

$$v(t) = -cos(\omega t) + 1 \quad (2.16)$$

$$w(t) = 1 \quad (2.17)$$

where ω is the angular frequency of the drive, we drive the system along a closed loop in the parameter regime that is shown in fig. 2.5. In that case, the transport scheme is represented by a loop circular around the origin, starting and ending at a point on the negative $v - w$ axis. It is important to note that any set of parameters that enclose the origin without crossing it could lead to a successful transport.

2.5.1 Adiabatic Condition

To understand the condition for adiabaticity of the transport scheme, in this section a derivation for the adiabatic approximation is shown. Given a time-dependent Hamiltonian, $H(t)$, an arbitrary state $|\Psi(t)\rangle$ evolves according to the Schrodinger equation,

$$i\hbar\partial_t|\Psi(t)\rangle = H(t)|\Psi(t)\rangle \quad (2.18)$$

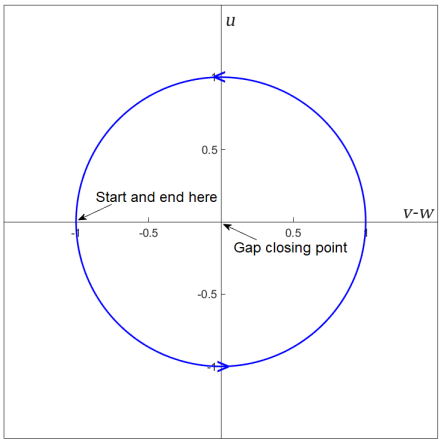


Figure 2.5: Path followed in the parameter space

Expressing the state and Hamiltonian in the the instantaneous eigenstates basis, $\{|\psi_n(t)\rangle\}$, leads to

$$|\Psi(t)\rangle = \sum_n C_n(t)\,|\psi_n(t)\rangle \tag{2.19}$$

$$H(t) = \sum_n E_n(t)\,|\psi_n(t)\rangle\,\langle\psi_n(t)| \tag{2.20}$$

where $\{E_n\}$ are the eigenenergies of the Hamiltonian, and C_n represents the projection of the arbitrary state on the instantaneous eigenstate $|\phi_n\rangle$, i.e. $C_n = \langle\psi_n(t)|\Psi(t)\rangle$. After substituting (3.4) and (3.5) into (3.3), we arrive at,

$$\dot{C}_n = -C_n(t)\,\langle\psi_n(t)|\partial_t\psi_n(t)\rangle - \sum_{m\neq n} C_n(t)e^{i(\theta_m-\theta_n)}\frac{\langle\psi_n(t)|\,\partial_t H\,|\psi_m(t)\rangle}{E_m(t)-E_n(t)} \tag{2.21}$$

The adiabatic approximation is valid when the second term is ignored, i.e.

$$\dot{C}_n \approx -C_n(t)\,\langle\psi_n(t)|d_t\psi_n(t)\rangle \tag{2.22}$$

Which means that if the initial state is an eigenstate, all the other C_ns are zero and at all time. A transition to other eigenstates will not be induced. The validity condition thus formally reads

$$C_n(t)\,\langle\psi_n(t)|\partial_t\psi_n(t)\rangle \gg \sum_{m\neq n} C_n(t)e^{i(\theta_m-\theta_n)}\frac{\langle\psi_n(t)|\,\partial_t H\,|\psi_m(t)\rangle}{E_m(t)-E_n(t)} \tag{2.23}$$

which reduces to

$$1 \gg \frac{\dot{\lambda}}{E_m - E_n} \tag{2.24}$$

Where λ are the parameters of the time-dependent Hamiltonian. The inequality (2.24) means that for a process to be adiabatic, the rate of change of the parameter must be small compared to the smallest energy difference between the eigenstates,i.e. the energy gap.

2.5.2 Simulated Adiabatic Transport

The transport process following the parameter scheme described in section 2.4 is numerically simulated (see Appendix C for Matlab code). A single excitation on the left edge site is evolved under the time-dependent Hamiltonian (2.14) with parameters following equations (2.15)-(2.17) (fig 2.6) for one period of change in parameters, $2\pi/\omega$. Fig. 3.2. shows how the energy of the eigenstates and the amplitudes of the wavefunction over the sites of the lattice change with time. The angular frequency, $\omega/w = 2\pi \times 10^{-4}$, is chosen so the protocol remains deep in the adiabatic regime.

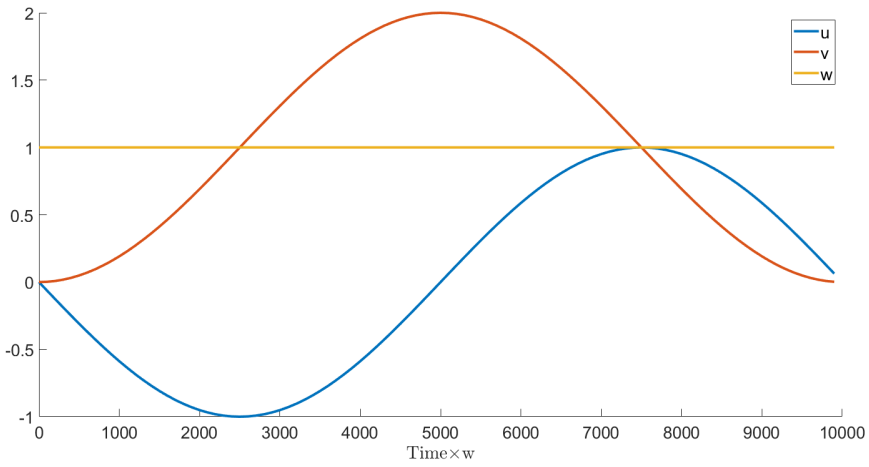


Figure 2.6: Parameters against time

The energy of the transported state is also calculated and was found to follow a single energy line as shown in fig. 2.7(a). As expected, we see from fig. 2.7(b) that the initial zero-energy state is the left edge state. During the

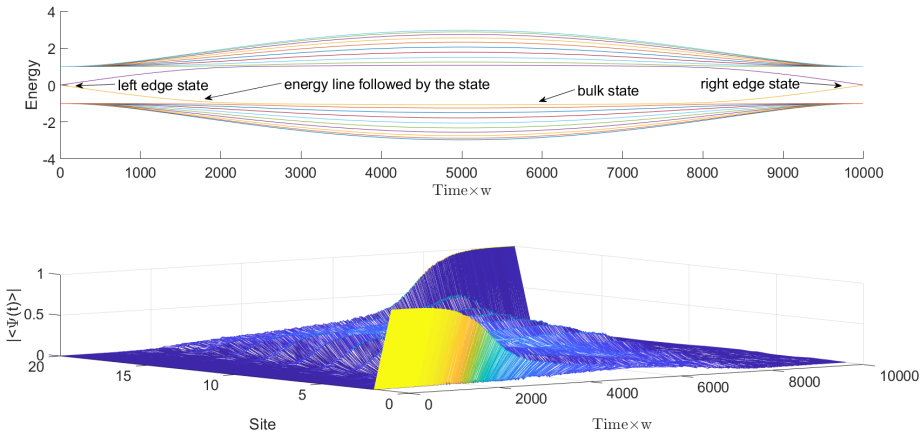


Figure 2.7: Energies of eigenstates (above) and wavefunction (below) against time.

transport, this eigenstate adiabatically evolved into a bulk state, as shown by the delocalisation of the wavefunction. Finally, the system ends in the final zero-energy edge state localised on the right boundary, as predicted.

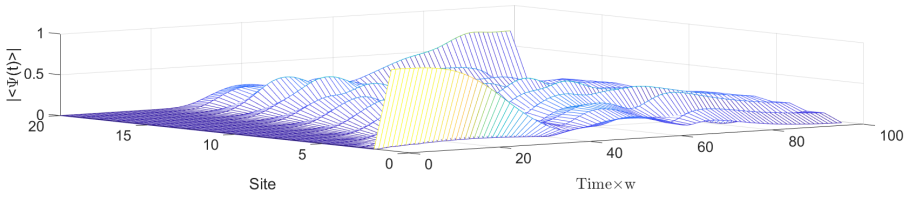


Figure 2.8: Wavefunction over time for a failed transport

In contrast, we show in fig. 2.8 a case where the parameters are changed fast enough to break the adiabatic condition, which leads in a final state that remains delocalised in the bulk.

Chapter 3

Edge State Robustness to Disorder

In this chapter, we present a systematic study of the edge states in presence of local changes of the onsite potential or hopping amplitudes. Such analysis allows us to develop a good sense of the fundamentally different effects of the different type of imperfections. We stress that this is in contrast to the robustness of the zero-energy states to global change of the parameter v/w within the topological phases, as shown in fig. 2.2.

3.1 Case I: Imperfection in the Onsite Potential

We first investigate the consequences of having a local imperfection in the onsite energy u (cf. equation (2.14)). To do so, I explicitly consider the Hamiltonian

$$H' = H_{SSH} + \epsilon \left|i\right\rangle \left\langle i\right| \tag{3.1}$$

where ϵ and i represent the strength and the site position of the defect respectively. Similar to the onsite energy included in equation (2.14), this imperfection breaks the chiral symmetry of our model, albeit locally this time.

In fig. 3.1, the energy spectrum of the Hamiltonian (3.1) is plotted as a function of the defect strength for positions i = 1,2 and 5. A consequence of breaking of the chiral symmetry, even only locally, is that the energy

spectrum is not symmetric anymore, i.e. for each eigenstate with energy E its chiral symmetric partner of energy $-E$ is not necessarily present. For the imperfection at $i = 1$, only the left edge state is considerably touched, while many more bulk states are affected for $i = 5$.

As expected, the edge state are exponentially more sensitive to defects closer to the edge (e.g $i = 1$ versus $i = 5$) except if the defect is positioned on a B site (e.g. $i = 2$) where no amplitude of the left edge state is initially there. An interesting observation is the capability of the defect to pull some bulk state into the energy gap, as shown in fig. 3.1 (c). As we show in the next chapter, such state can play an important role during the transport protocol. Identical results apply to the right edge state.

We now investigate the form of some of the important associated eigenstates. We show in fig. 3.2 (a) and (b) the left edge state for a disorder on site $i = 1$ and strength $\epsilon/w = -0.5$ and 0.5 respectively. Again, given the chiral symmetry breaking, the edge states are no longer only populating the A sites, but are now mixed between the two basis. The same apply for the site $i = 5$ (not shown) but with exponentially weaker population of the B sites.

The most important observation comes by looking at the eigenstate associated with the energy line pulled into the gap by the presence of a defect at $i = 5$. In fig. 3.2 (c) we show its state probability for $\epsilon/w = 2$ (indicated by a circle in fig. 3.1 (c)). We observe the apparition of a new localised state, but this time, near $i = 5$. We also compare it to the zero-energy state for the same parameter (cf. fig 3.2 (d)). We conclude that large enough imperfections can generate dislocation in the chain, thus allowing additional localised states to appear.

We finally show in fig. 3.3 the population on the B site associated with the zero-energy state localised on the left boundary for different positions as a function of ϵ. We use this metric as an indication of how the edge state is perturbed. Again, it shows the exponential sensitivity to the distance of the defect.

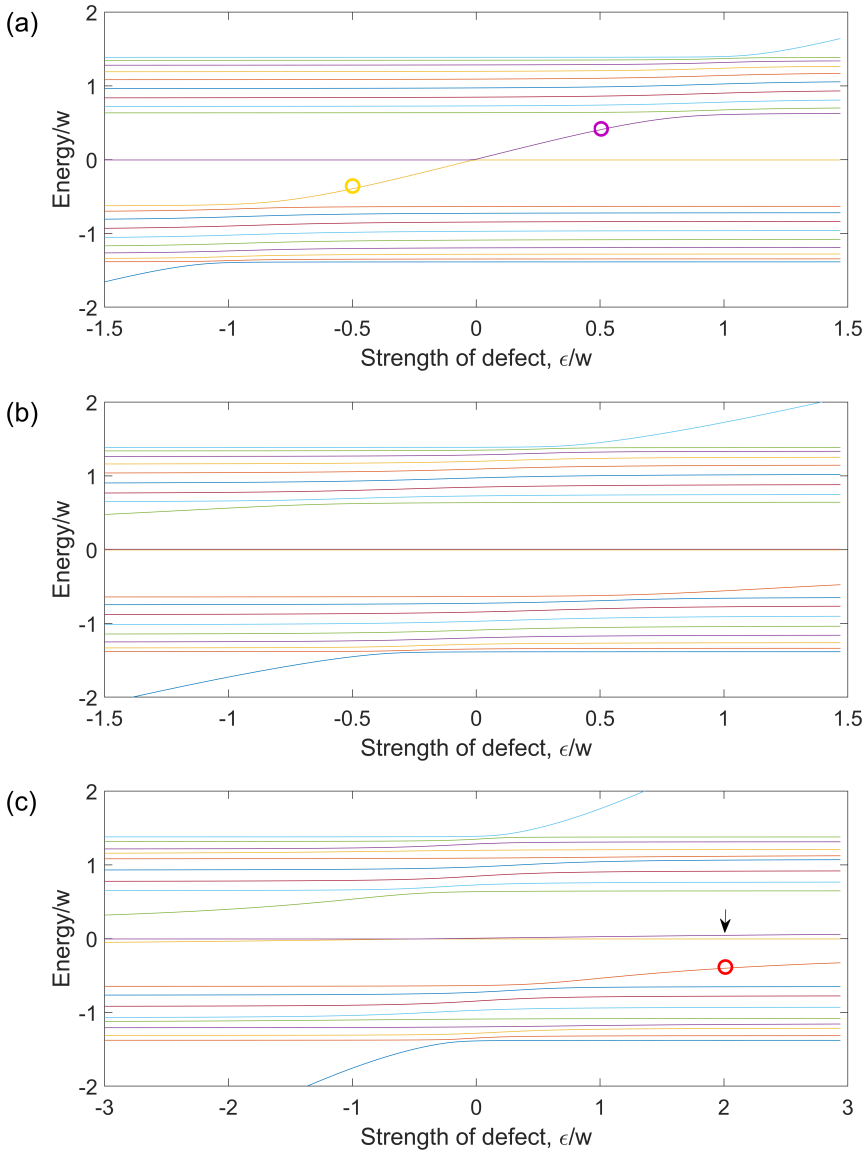


Figure 3.1: Energy spectrum against strength of onsite potential defects for i =1(a), 2(b), and 5(c). v and w are set to be 0.4 and 1 respectively. States presented in fig 3.2 are marked in the figures. Yellow and purple circles in (a) represent states in 3.2 (a) and (b) respectively. Red circle and arrow in (c) represent states in 3.2 (c) and (d).

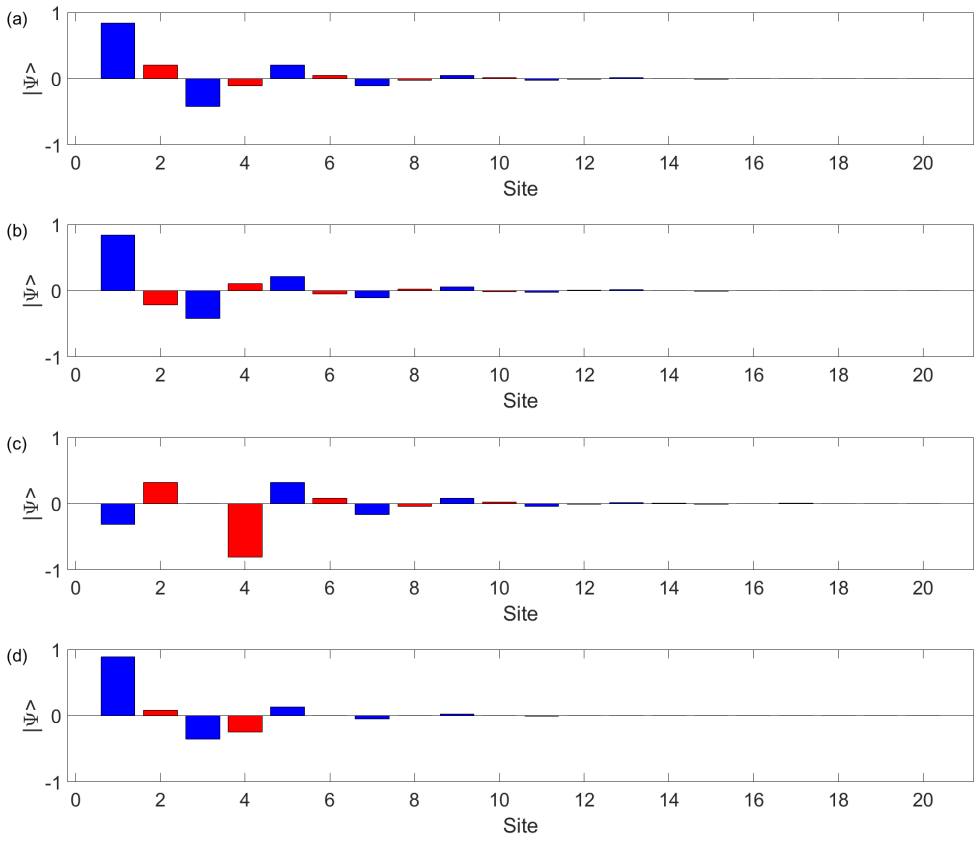


Figure 3.2: (a) and (b) shows the left edge state when a defect with strengths $\epsilon/w = -0.5$ and 0.5 respectively is put on the first site $i = 1$. (c) An eigenstate with an eigenenergy in the gap when $i = 5$ $\epsilon/w = 2$. (d) left edge state when $i = 5$ $\epsilon/w = 2$.

3.2 Case II: Defects in the Hopping Amplitudes

In this section we investigate the consequences of an imperfection in the hopping amplitudes. More precisely, we analyse the Hamiltonian

$$H' = H_{SSH} + \epsilon(|i\rangle \langle i+1| + h.c.) \tag{3.2}$$

where a defect ϵ affects the hopping rate between sites i and $i+1$. Unlike the imperfection in the onsite energy, imperfection in the hopping amplitudes

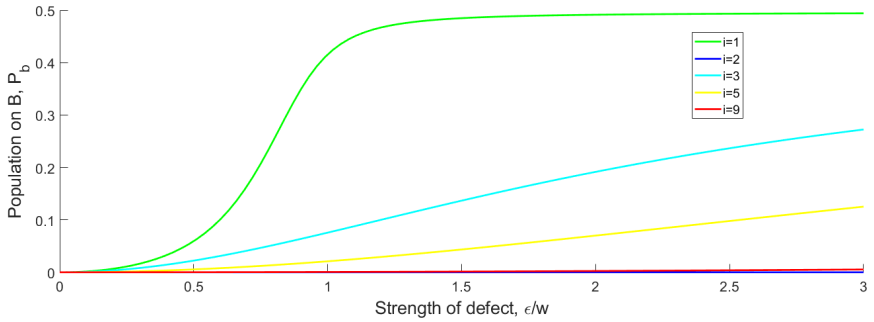


Figure 3.3: Population on the B sublattice against defect.

does not break the chiral symmetry of the system. Thus, more robustness of the edge states is expected against this kind of defect. Fig. 3.4 shows the the energy spectrum of the Hamiltonian (3.2) against defect strengths for $i = 1$, 2, and 10. We observe that the spectra always remain symmetric, a consequence of the unbroken chiral symmetry.

The energy line representing the zero-energy edge states are clearly unpertubed given hopping defects in all cases when $i = 1$, 2, or 10 regardless of ϵ. This suggests that the edge states are indeed robust against this defect. From fig. 3.4(b) and (c) we observe that just as in the case of onsite defect when $i = 5$ discussed in the previous section, an energy line is pulled towards the gap. By comparing 3.4(b) and (c), we notice that the line remains in the bulk for a wider range of ϵ when the defect is further away from the edge.

Fig.3.5, which represents the eigenstate with an energy in the gap, when $i = 10$ and $\epsilon = -1.2$ shows that the state represented by this line is also a localized state. More precisely, this localized state is in fact a hybridized edge state that appears in the middle of the lattice as the lattice is being cut by the defect introduced. These edge states are found to persist more when the defect is further away from the edge. This state will have a significant impact on the efficiency of transport.

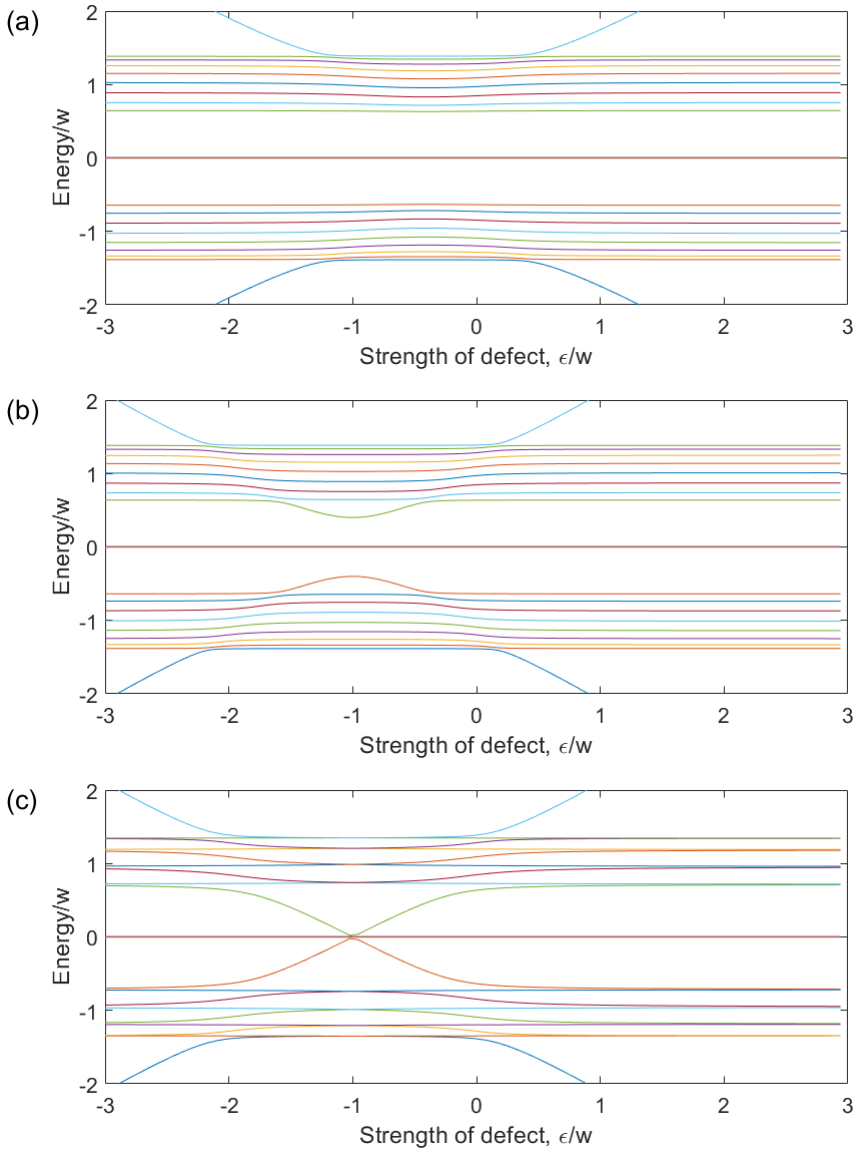


Figure 3.4: Energy spectrum against strength of hopping defects for i =1, $i = 2$, and $i = 10$.

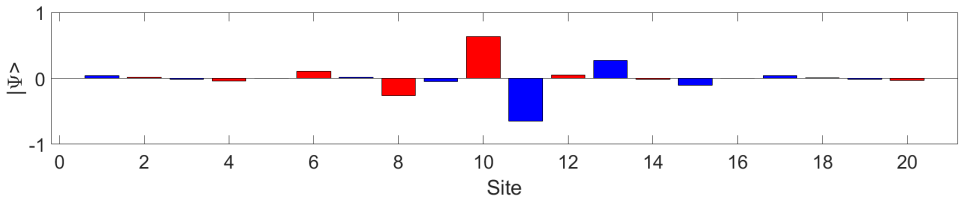


Figure 3.5: Wavefunction of n localized state (hybridized edgestates in the middle of the lattice produced by the hopping defect) for $i = 10$ and $e = -1.2$

Chapter 4

Effects of Defects on the Transport Efficiency

The goal of this honours project is to answer the question of how robust or how protected the transport scheme is against random defects, and therefore what degree of precision is required for a physical implementation of such a transport scheme. In this chapter, we address this question. The effects of individual defects on the transport efficiency is studied first to understand the effect of random defects later presented at the end of this chapter.

We recall how the transport is done when there is no defect as explained in section 2.5. The time-dependent Hamiltonian (2.14) reads,

$$H(t) = v(t)\sum_{m=1}^{N}(|m,B\rangle\langle m,A| + h.c.) + w(t)\sum_{m=1}^{N-1}(|m+1,A\rangle\langle m,B| + h.c.) + u(t)\sum_{m=1}^{N}(|m,A\rangle\langle m,A| - |m,B\rangle\langle m,B|) \quad (4.1)$$

and the parameters are changed according to equations (2.15)-(2.17) which reads,

$$u(t) = -sin(\omega t) \quad (4.2)$$

$$v(t) = -cos(\omega t) + 1 \quad (4.3)$$

$$w(t) = 1 \quad (4.4)$$

To probe the efficiency of transport scheme against defects, the fidelity of transport given specific defects is measured. The initial state of the transport

is given by,

$$|\Psi(t=0)\rangle = |1, A\rangle \tag{4.5}$$

where $|N, A\rangle$ is a single excitation on the left edge site of the lattice. When there is no defect in the lattice, the final transported state after the Hamiltonian is changed over time according to equation (2.7) for one period, T, is given by,

$$|\Psi(t=T)\rangle = |N, B\rangle \tag{4.6}$$

where $|N, B\rangle$ is a single excitation on the right edge site of the lattice.

When a defect is introduced to the transport, an additional term is added to equation (4.1) and after the parameters are driven for one period according to (4.2)-(4.4), final state, $|\Psi(t=T)\rangle$, should no longer be $|N, B\rangle$ in general. The fidelity of transport, F, is the projection of $|\Psi(t=T)\rangle$ onto $|N, B\rangle$ squared, given by

$$F = |\langle N, B|\Psi(t=T)\rangle|^2 \tag{4.7}$$

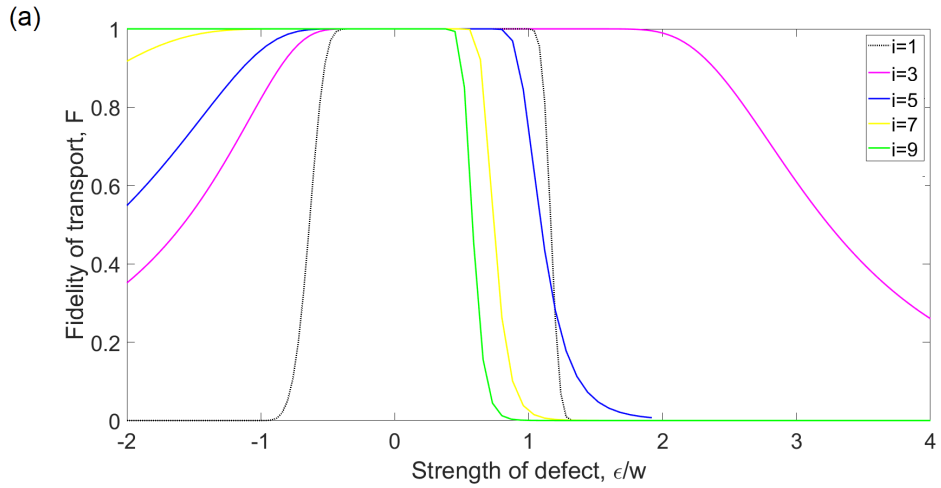
4.1 Case I: Defects in the Onsite Potential

In this section the effect of individual static local onsite potential defect at different strengths and at different positions on the efficiency of transport is studied. We introduce an onsite potential defect identical to the defect in section 3.1, $\epsilon |i\rangle \langle i|$.

Fig. 4.1 shows the fidelity of transport against the strength of local static onsite potential defects placed at odd (a) and even (b) sites. The drop in fidelity for $i = 1$ starts at $\epsilon = 1$. This is explained by fig. 4.3(a). In 4.3(a) (and other similar plots for other ϵ), as indicated, the effect of the added defect is to increase the energy of the initial state, which is an excitation only in the first site. When ϵ is increased beyond 1, the line intersects all the bulk states and thus the transport will fail as transitions to the different bulk states will occur.

The drop for the fidelity for defects in the rest of the odd sites drops faster when i is further away from the left edge. This is explained by figs. 4.3(b) and (c). When i is closer, i.e. $i = 3$ to the left edge, the gap at $t \approx 500$ marked in the two figures, is increased. Such that if the parameters are changed slowly enough, the state can avoid the transition to the eigenstate with the lower energy, which is the localized states which is a result of the dislocation produced by the onsite defect. The nature of this state is covered

in section 3.1. When i is increased however, the finite gap is decreased, such that the state will get stuck in the dislocation and thus the fidelity drops. Due to this mechanism, the fidelity as a function of ϵ is more sensitive to an increase in the speed of the change of parameters. This result is shown in fig. 4.2. On the other hand, when the onsite defect on the even sites is strong enough, it closes of the energy gap later on in the transport process as shown in fig. 4.3(b). This also kills the fidelity as the transported state is stuck in the dislocation produced by this defect.



4.2 Case II: Defects in the Hopping Amplitudes

In this section the effect of a single static local onsite potential defect at different strengths and at different positions on the efficiency of transport is studied. We introduce a defect identical to defect in 3.2, $\epsilon(|i\rangle\langle i+1| + h.c.)$, which represents a defect on the i^{th} hopping (between sites i and $i+1$) with a strength of ϵ.

Figure 4.4 shows the fidelity against strength of Hopping defect. The fidelity is clearly not robust against defect in the first hopping, while it is robust for the rest of the hopping. The defect on the first hopping kills the transport significantly since an excitation in the left edge site is no longer an eigenstate of the initial Hamiltonian if we introduce the defect. Furthermore

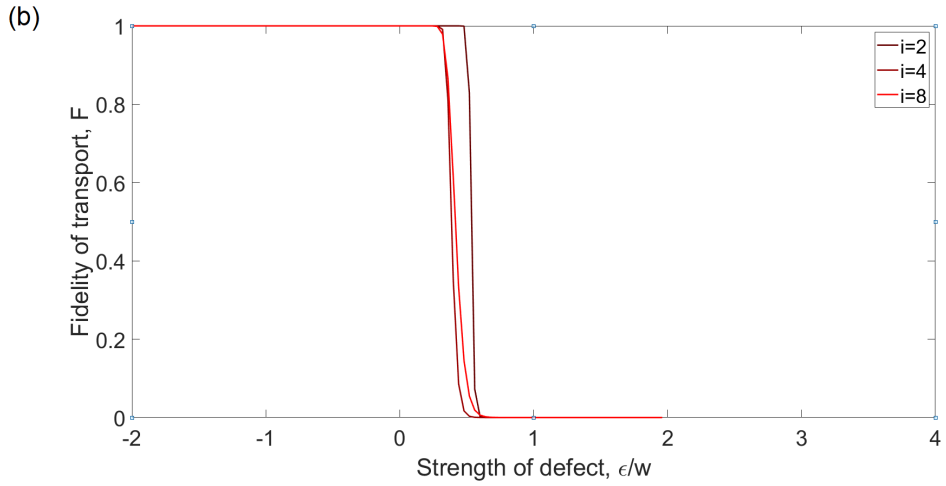


Figure 4.1: Fidelity against onsite potential defects on odd (a) and even (b) sites.

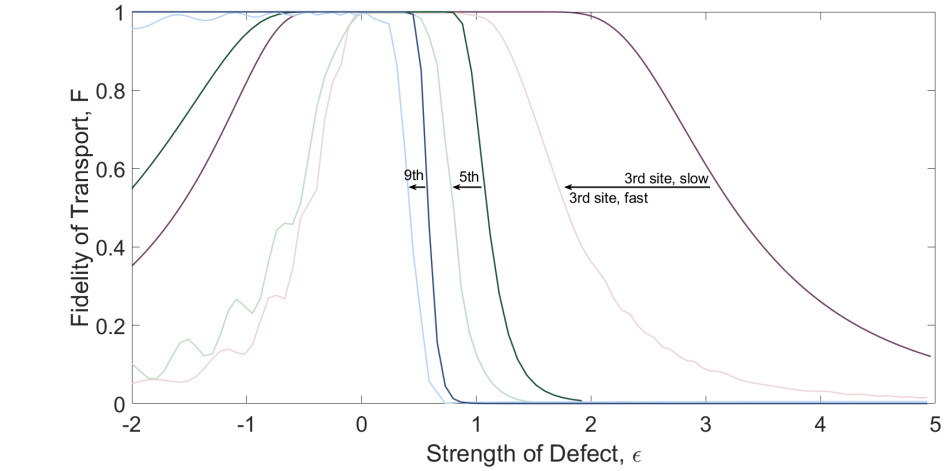


Figure 4.2: Response of Fidelity as a function of time to a change in the speed of the parameter driving.

for the first defect, we notice that there is a sharp dip in fidelity when $\epsilon = -2$, similar to the rest of the odd defects (odd is). This is due to the lattice being cut when $\epsilon = -2$, since v takes the range of $[-2, 2]$ in our transport scheme. The lattice is also cut with even defects when $\epsilon \approx -1$, this is for the same

reason. However, we find that the dip widens as i moves away from the edge. This is due to the new edge states, like the one presented in fig. 3.5, produced by the cut that persists even if the lattice is not completely cut. This also matches our observation in section 3.2 that the persistence of these edge states increases as i moves away from the edge, thus, it is easier to get stuck in these localized state in the middle.

4.3 Random defects

With an understanding of the various qualitatively different effects of the kinds of defects presented in the earlier sections, we study the aggreagate effect of a uniformly distributed random defects on all onsite potentials and hoppings on the effectiveness of transport. To do this, we simulate a large number of transport processes with random defects for a given average defect strengths. The fidelity of these transports are averaged and plotted against the average defect strengths. Fig 4.5 presents the plot. From fig. 4.5 we conclude that the transport scheme is indeed robust, when the strength of the defect is less than 10 percent of the intercell hopping w. As such a physical implementation of the transport scheme studied extensively in this project will work as long as the uncertainty of the hopping amplitudes and onsite potentials is less than 10 percent.

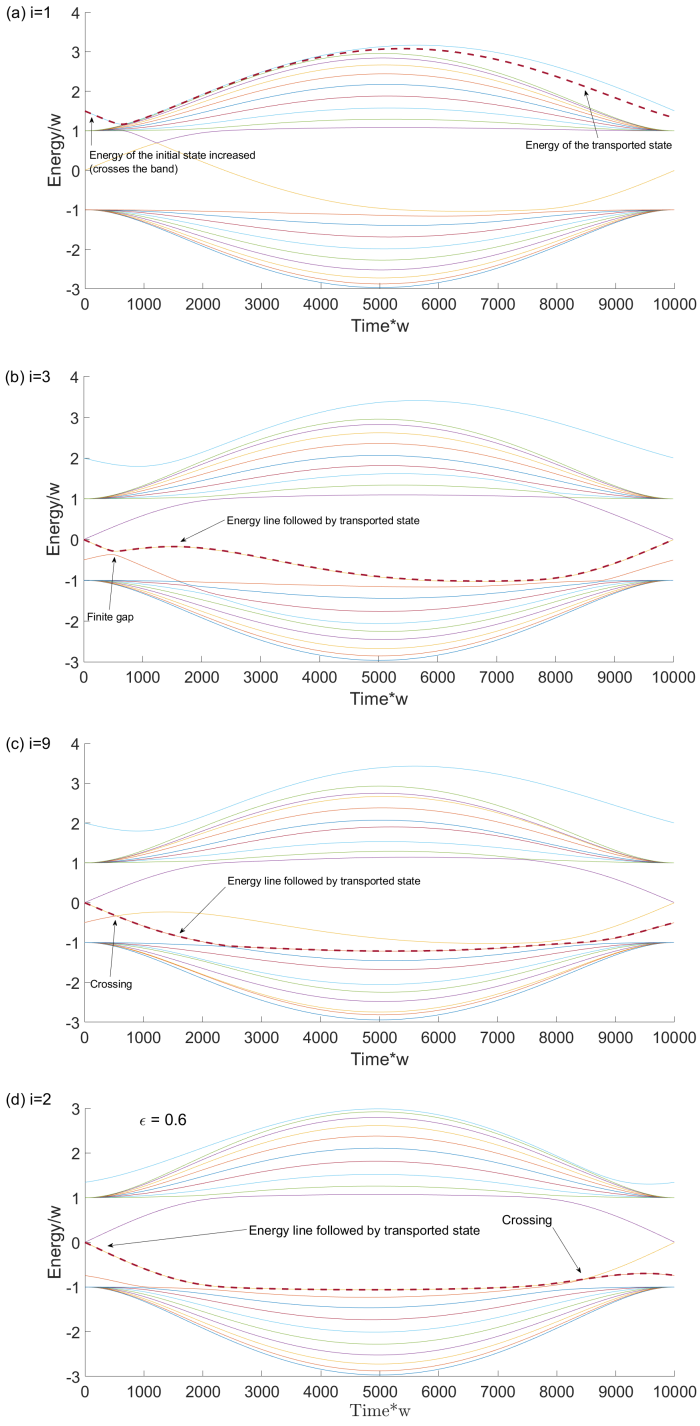


Figure 4.3: Energy of the transported states over time with the energy spectra as a background. (a) $i = 1$

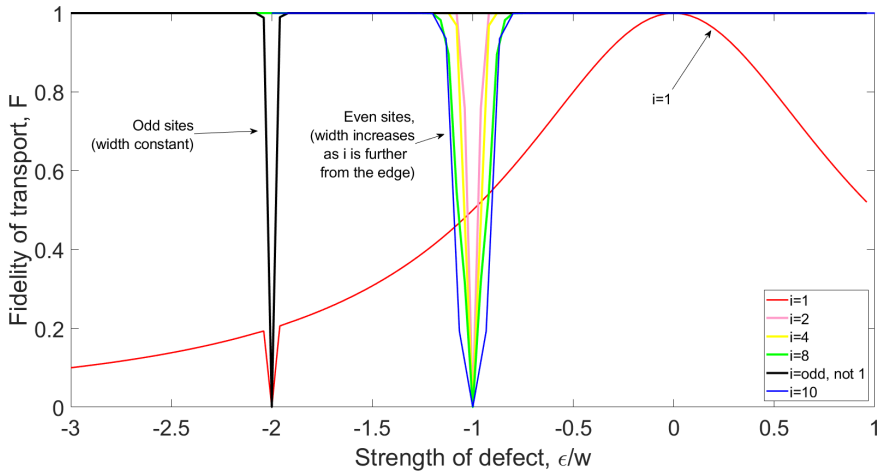


Figure 4.4: Fidelity against strength of hopping defects.

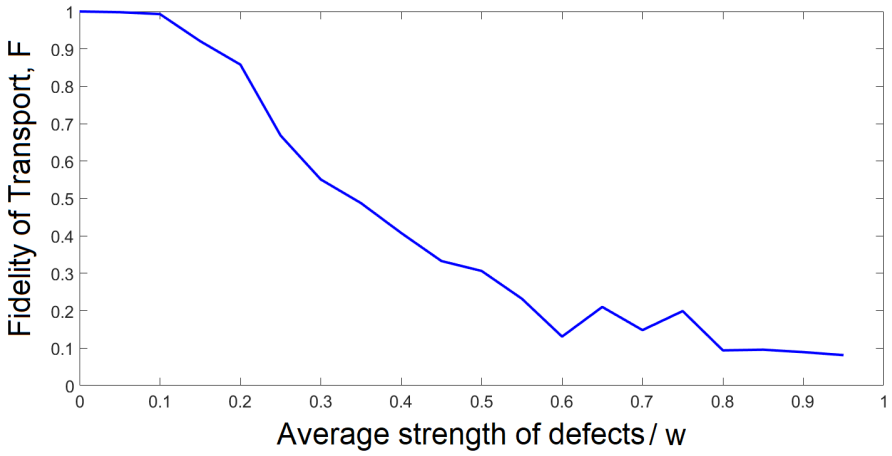


Figure 4.5: Fidelity for random static defects

Chapter 5

Physical Implementation

In this last chapter, we look at a specific physical implementation of the transport scheme studied in the preceding chapters. This chapter will first introduce photonic chips and its usefulness to simulate quantum processes. At the end we look at how the photonic chip can be utilized to specifically simulate our transport.

While the SSH Hamiltonian was first proposed to describe electrons in polyacetylene in [7], its applications have since then covered a much wider range of physical systems. Because the fermionic character does not play any role in the limit of a single particle (excitation) system, the SSH model has also been succesfully applied to single bosonic excitation models. A remarkable example is the implementation of the SSH Haniltonian using on-chip photonic circuits [10]. In this chapter, we describe such an implementation where electromagnetic waves propagating along a photonic waveguides simulate a single excitation SSH model, which then undergoes the transport scheme described in chapter 3.

5.1 Photonic Chips

As described in ref. [12], it is possible to fabricate a photonic chip (fig 5.1(b)) from a bulk of glassy Silica where one-dimensional waveguides are engineered using high-power femtosecond laser. (illustration of the fabrication process shown in fig 5.1(a)) To do so, one locally increases the index of refraction within the silica chip by focusing the high-intensity laser beam, resulting in a reconfiguration of the Silica bonds. By total internal refraction,

propagating light field are then guided along those higher refractive-index lines, leading to high quality optical waveguides.

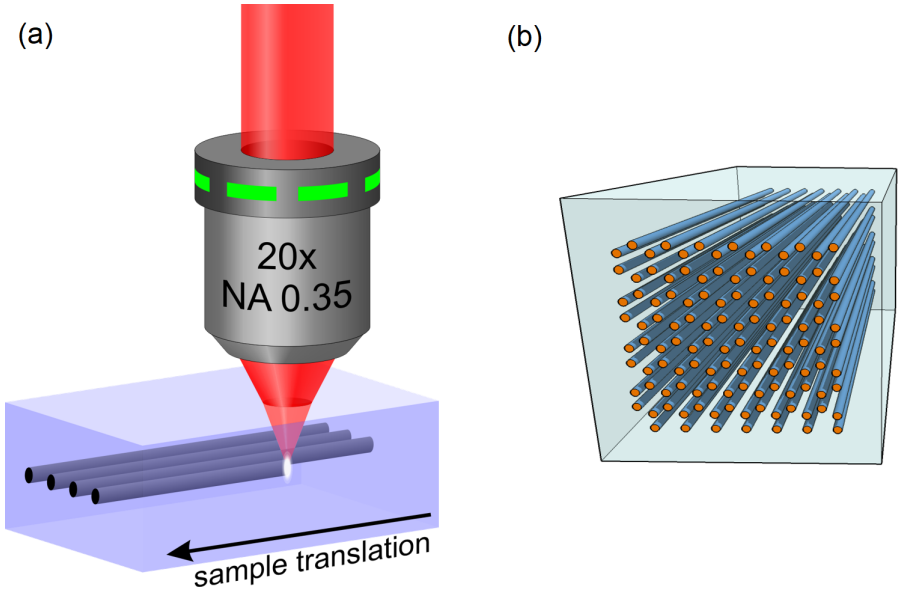


Figure 5.1: Fabrication process of a photonic chip (a). An illustration of a photonic chip (b). Pictures taken from [12].

5.2 Photonic Quantum Simulation

An electromagnetic wave propagating in a one dimensional waveguide follows Helmholtz equation

$$i\lambda\partial_z E(x,z) = -\left(\frac{\lambda^2}{2n_0}\partial_x^2 + \Delta n(x,z)\right)E(x,z) \tag{5.1}$$

where λ represents the reduced wavelength, $\lambda = \lambda/2\pi$, $E(x,z)$ represents the electric field envelope, n_0 represents the initial refractive index of the glassy silica bulk, and $\Delta n(x,z)$ represents the change in refractive index introduced by the femtosecond laser. Equation (5.1) already shows resemblance with the Schrodinger equation for a quantum particle moving within some external potential (e.g. a single electron within the potentials in a polyacetylene

molecule, the SSH),

$$i\hbar\partial_t\Psi(x,t) = -\left(\frac{\hbar^2}{2m}\partial_x^2 - V(x,t)\right)\Psi(x,t) \tag{5.2}$$

A map between equations (5.2) and (5.1) is possible if the wavefunction of the quantum particle, $\Psi(x,t)$, is associated with the electric field envelope of light, $E(x,z)$ and time, t, is mapped into the z direction. In that case the mass of the particle, m, corresponds to the bulk refractive index, n_0 and the potential $V(x,t)$ is related to the change in the refractive index, $\Delta n(x,z)$. Thus, a propagation of electromagnetic wave within these photonic structures serves as a simulation of the evolution of a quantum particle subjected to an external potential.

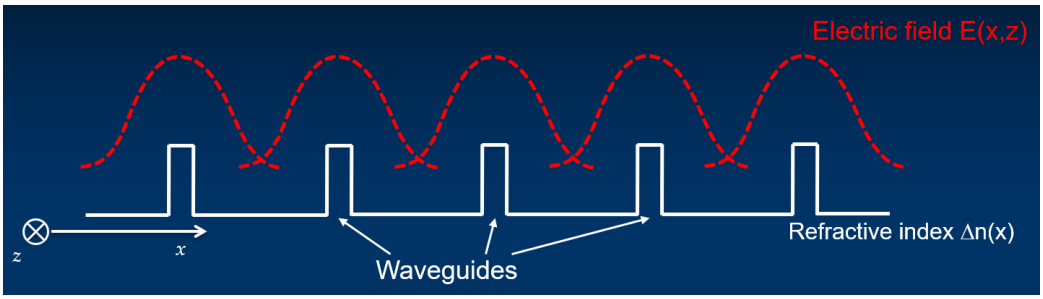


Figure 5.2: An illustration of the periodic change of refractive index and a propagating electric field envelope in the waveguides.

By taking (5.1), after separating the field envelope, $E(x,z)$, into its z-dependent and x-dependent components, $\varphi(z)$ and $\zeta(x)$ respectively and applying it to a linear combination of eigenmodes in the m^{th} waveguides, $E = \sum_m^N \varphi_m\zeta_m$ (see fig. 5.2), and , we get,

$$i\partial_z E = i\partial_z(\sum_{m=1}^{N}\varphi_m\zeta_m) = -\sum_{m=1}^{N}\sum_{\mu=-1}^{1} c_{m,m+\mu}\varphi_{m+\mu}\zeta_{m+\mu} \tag{5.3}$$

where $c_{m,m+\mu}$ is the coupling between the m^{th} and $m+\mu^{th}$ waveguide, and is given by,

$$c_{m,m+\mu} = \frac{\omega\epsilon_0}{4}\int_x [\Delta n(x)]^2\zeta_m\zeta_{m+\mu}dx \tag{5.4}$$

where ω is the optical frequency and ϵ_0 is the dielectric constant. Simplifying (6.3) and together with an energy conservation requirement, $c_{n,n+1} = c_{n+1,n}$ gives us,

$$i\partial_z E = \Bigg(- \sum_{\text{odd n}}^{N} c_{n,n+1}(\zeta_n \zeta_{n+1}^{-1} + \zeta_{n+1}\zeta_n^{-1}) - \sum_{\text{even n}}^{N-1} c_{n,n+1}(\zeta_n \zeta_{n+1}^{-1} + \zeta_{n+1}\zeta_n^{-1}) - \sum_{n=1}^{N} c_{n,n}(\zeta_n \zeta_n^{-1}) \Bigg) E \quad (5.5)$$

Recalling the modified SSH Hamiltonian (3.2) and together with the Schrodinger equation applied to an arbitrary state, $|\Phi\rangle$, gives us

$$i\partial_t |\Phi\rangle = \Bigg(v \sum_{m=1}^{N} (|m, B\rangle \langle m, A| + h.c.) + w \sum_{m=1}^{N-1} (|m+1, A\rangle \langle m, B| + h.c.) + u \sum_{m=1}^{N} (|m, A\rangle \langle m, A| - |m, B\rangle \langle m, B|) \Bigg) |\Phi\rangle \quad (5.6)$$

Comparing the equations, we see that the electromagnetic wave propagating within the waveguides in the photonic structures described by equation (6.5) is analogous to the evolution of an arbitrary quantum state in the modified SSH described by equation (6.6) if z is associated with t, and the couplings between the waveguides are adjusted such that the odd and even $c_{n,n+1}$ represent the v and w respectively, and the odd and even $c_{n,n}$ represent the u and $-u$ respectively.

The coupling in (6.3) and (6.5) only describes nearest neighbour hopping in analogy with the tight-binding model used to derive the SSH Hamiltonian. This is justified by considering the exponentially decaying strength of the wave envelope in the x and y direction. The coupling between the waveguides serves as analogy for the hopping amplitudes between different sites.

5.3 Implementation

An implementation of the static SSH has been proposed previously by S. Longhi in [10]. In this proposal, primary waveguides are placed equally spaced and thus there is a uniform coupling between all next-neighbour

waveguides. The coupling between the waveguides are altered by adding auxiliary waveguides after every other waveguide (see fig. 5.3). The auxiliary waveguides increase the coupling between the primary waveguides as it allows an additional two-step hopping between the primary waveguides through the auxiliary waveguides. This technique is proposed in order to simulate the staggered hopping v and w needed to study the different topological phases of the SSH Hamiltonian.

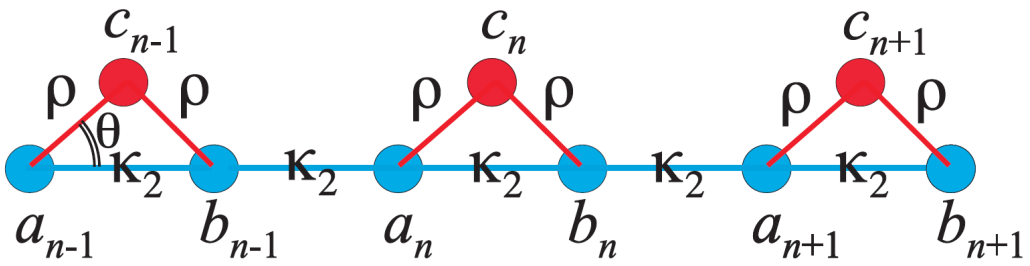


Figure 5.3: Set-up proposed by S. Longhi. Picture taken from [10]. Original caption: "Photonic realization of the SHH model based on an array of coupled optical waveguides with equal spacing $d/2$, [sites $\{a_n\}$ and $\{b_n\}$]. The auxiliary waveguides at sites c_n are used to control the hopping rates." κ_2 and ρ represent the couplings

The implementation by Longhi is an implementation of the static SSH Hamiltonian, which does not allow dynamical transport as studied in the report. Given the analogy between the z direction and time discussed previously, the analogous waveguides should also change in the z direction in terms of the refractive indices and the coupling between the neighbouring waveguides. Such an implementation was done by Y. E. Kraus *et al.* in [13]. In this experiment, they show a successful implementation of the transport of a topologically protected edge state, albeit in a different one dimensional model known as the Andre-Aubry-Harper model. A complete overview of the model is given in [14] and . In the experiment, the couplings between the waveguides are changed along the z direction by changing the distances between the waveguides (see fig. 6.2).

In the transport scheme presented in this report, we change the onsite potential in addition to the change of hopping amplitude. This could be done by changing the refractive indices of the waveguides along the z direction which is analogous to the onsite potential. Thus, by changing the distances

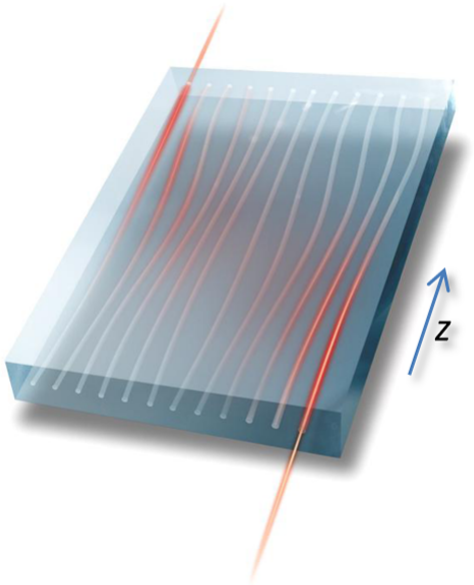


Figure 5.4: Changing coupling between waveguides along z. Picture taken from [13]. Original caption: "An illustration of the adiabatically modulated photonic quasicrystal, constructed by slowly varying the spacing between the waveguides along the propagation axis z"

between the waveguides and the refractive indices along the z direction, we could implement the dynamical transport described by the Hamiltonian (3.3) of section 3.1.

Bibliography

[1] K. v. Klitzing, G. Dorda, M. Pepper *New method for High-Accuracy Determination of the Fine-Structure Constant Based on Quantized Hall Resistance* Physical Review Letters, 45(6):494-497, August 1980

[2] D. J. Thouless, M. Kohmoto, M. P. Nightingale, and M. den Nijs *Quantized Hall Conductance in a Two-Dimensional Periodic Potential* Physical Review Letters 49(6):405-408, 1982.

[3] Yoichi Ando *Topological Insulator Materials* Journal of the Physical Society of Japan, arXiv:1304.5693v3 [cond-mat.mtrl-sci], 2013

[4] Jean-Luc Tambasco, Giacomo Corrielli, Robert J. Chapman, Andrea Crespi, Oded Zilberberg, Roberto Osellame and Alberto Peruzzo *Quantum interference of topological states of light* Science Advances, 4(9):eaat3187, 14 Sep 2018

[5] Ling Lu, John D. Joannopoulos, Marin Soljačić *Topological photonics* Nature Photonics volume 8, pages 821–829 (2014)

[6] Alexander B. Khanikaev, S. Hossein Mousavi, Wang-Kong Tse, Mehdi Kargarian, Allan H. MacDonald, Gennady Shvets *Photonic topological insulators* Nature Materials volume 12, pages 233–239 (2013)

[7] W.P. Su, J. R. Schrieffer, A. J. Heeger. *Solitons in Polyacetylene*. Physical Review Letters, 42(25):1698–1701, 1979.

[8] J. K. Asboth, L. Oroszlany, A. Palyi. *A Short Course on Topological Insulators: Band-structure topology and edge states in one and two dimensions*. arXiv:1509.02295, 2015.

[9] D. J. Thouless, *Quantization of particle transport*. Phys. Rev. B 27, 6083–6087 (1983).

[10] S. Longhi *Zak Phase of Photons in Optical Waveguide Lattices*. Optics Letters, 38(19):3716-3719, 2013.

[11] David S. Simon, Casey A. Fitzpatrick, Shuto Osawa, and Alexander V. Sergienko *Quantum simulation of topologically protected states using directionally unbiased linear-optical multiports* Physical Review Letters, 96(1):13858–13866, 2013.

[12] A. Szameit, S. Nolte *Discrete optics in femtosecond-laserwritten photonic structures* J. Phys. B: At. Mol. Opt. Phys. 43:163001(25pp), 2010

[13] Yaacov E. Kraus, Yoav Lahini, Zohar Ringel, Mor Verbin, and Oded Zilberberg *Topological States and Adiabatic Pumping in Quasicrystals* Physical Review Letters, 109(10):106402(5), 2012

[14] Sriram Ganeshan, Kai Sun, and S. Das Sarma *Topological Zero-Energy Modes in Gapless Commensurate Aubry-André-Harper Models* Phys. Rev. Lett. 110, 180403, May 2013

[15] P. G. Harper *Single Band Motion of Conduction Electrons in a Uniform Magnetic Field* Proc. Phys. Soc. A 68 874, 1955

Appendix A

Spectral Plots

```
cells=10

phaseadjust=@(v,kth)v*(v(kth)/abs(v(kth)))

%Making hamiltonian
N=2*cells

%making w
row=1
column=1
wmat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row-1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row+1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
```

```
    end
    wmat=[wmat;rmat]
    column=1
    row=row+1
end

%making v
row=1
column=1
vmat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row+1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row-1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    vmat=[vmat;rmat]
    column=1
    row=row+1
end

%making u
row=1
column=1
umat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
```

```
            end
        else
            if column==row
                rmat=[rmat,-1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    umat=[umat;rmat]
    column=1
    row=row+1
end

epssmat=hepsmat(cells,10)

A=@(u,v,w,eps)umat.*u+vmat.*v+wmat.*w+epssmat.*eps
title=strcat('Local onsite perturbation at ninth site')


%evolution
taunaught=-3
taufinal=3
taudiv=100
dtau=(taufinal-taunaught)/taudiv
t=[taunaught:dtau:taufinal-dtau]

m=(taufinal-taunaught)/(taufinal-taunaught)
v=@(t)0.4
u=@(t)0.005
w=@(t)1
eps=@(t)t

tau=taunaught
catedeige=[]
catedstate=[]
catedv=[]
catedu=[]
catedw=[]
catedeps=[]
while tau<=taufinal
    catedeige=[catedeige,eig(A(u(tau),v(tau),w(tau),eps(tau)))]
    catedv=[catedv,v(tau)]
    catedu=[catedu,u(tau)]
```

```
    catedw=[catedw,w(tau)]
    catedeps=[catedeps,eps(tau)]
    tau=tau+dtau
end

figure
plot(t,catedeige)
ylabel('Energy/w')
xlabel('Strength of defect, \epsilon/w')
ylim([-2 2])

figure
plot(t,catedv)
hold on
plot(t,catedu)
plot(t,catedw)
plot(t,catedeps)
legend
ylabel('parameter values')
set(gca,'XTick',[])
hold off

[V,D]=eig(A(u(taufinal),v(taufinal),w(taufinal),eps(taufinal)))

function y=epsmat(cells,site)
    row=1
    column=1
    umat=[]
    N=2*cells
    while row<=N
        rmat=[]
        while column<=N
            if column==row & row==site
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
            column=column+1
        end
        umat=[umat;rmat]
        column=1
        row=row+1
    end
y=umat
end
```

```
function y=hepsmat(cells,site)
    row=1
    column=1
    N=cells*2
    y=[]
    while row<=N
        rmat=[]
        while column<=N
            if (column==row+1 & row==site | column==row-1 & column==site)
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
            column=column+1
        end
        y=[y;rmat]
        column=1
        row=row+1
    end
end
```

Appendix B

Wavefunctions Generation

```
cells=10

phaseadjust=@(v,kth)v*(v(kth)/abs(v(kth)))

%Making hamiltonian
N=2*cells

%making w
row=1
column=1
wmat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row-1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row+1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
```

```
    end
    wmat=[wmat;rmat]
    column=1
    row=row+1
end

%making v
row=1
column=1
vmat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row+1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row-1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    vmat=[vmat;rmat]
    column=1
    row=row+1
end

%making u
row=1
column=1
umat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
```

```
            end
        else
            if column==row
                rmat=[rmat,-1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    umat=[umat;rmat]
    column=1
    row=row+1
end

M1=@(u,v,w)umat.*u+vmat.*v+wmat.*w
M2=@(u,v,w,epssite,eps,hepssite,heps)M1(u,v,w)+epsmat(cells,epssite).*eps+hepsmat(ce

%{
Hamiltonian construction ends here
--------------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
%}

A=@(u,v,w,eps)M2(u,v,w,5,0,10,eps)
X=size(A(0,0,0,0))
X1=X(1)


ut=0.005
vt=0.4
wt=1
epst=-1.2

bth=X1/2+2

[V,D]=eig(A(ut,vt,wt,epst));
a=V(1:X1,X1/2);
aval=D(X1/2,X1/2);
b=V(1:X1,bth);
bval=D(bth,bth);

figure
c=-1*(phaseadjust(b,3))'
```

```
cc=[c(1),0,c(3),0,c(5),0,c(7),0,c(9),0,c(11),0,c(13),0,c(15),0,c(17),0,c(19),0]
ccc=[0,c(2),0,c(4),0,c(6),0,c(8),0,c(10),0,c(12),0,c(14),0,c(16),0,c(18),0,c(20)]
bar(cc,'FaceColor','b','EdgeColor','black','LineWidth',0.0000000000000001)
hold on
bar(ccc,'FaceColor','r','EdgeColor','black','LineWidth',0.0000000000000001)
hold off
ylim([-1,1])
set(gca,'XTick',[], 'YTick', [])
b=strcat('E=',num2str(bval))
%text(10,-0.5, b)
%{
bth=X1/2

[V,D]=eig(A(ut,vt,wt,epst));
a=V(1:X1,X1/2);
aval=D(X1/2,X1/2);
b=V(1:X1,bth);
bval=D(bth,bth);

figure
c=-1*(phaseadjust(b,3))'
cc=[c(1),0,c(3),0,c(5),0,c(7),0,c(9),0,c(11),0,c(13),0,c(15),0,c(17),0,c(19),0]
ccc=[0,c(2),0,c(4),0,c(6),0,c(8),0,c(10),0,c(12),0,c(14),0,c(16),0,c(18),0,c(20)]
bar(cc,'FaceColor','b','EdgeColor','black','LineWidth',0.0000000000000001)
hold on
bar(ccc,'FaceColor','r','EdgeColor','black','LineWidth',0.0000000000000001)
hold off
ylim([-1,1])
set(gca,'XTick',[], 'YTick', [])
b=strcat('E=',num2str(bval))
%text(10,-0.5, b)

bth=X1/2+1

[V,D]=eig(A(ut,vt,wt,epst));
a=V(1:X1,X1/2);
aval=D(X1/2,X1/2);
b=V(1:X1,bth);
bval=D(bth,bth);

figure
c=-1*(phaseadjust(b,3))'
cc=[c(1),0,c(3),0,c(5),0,c(7),0,c(9),0,c(11),0,c(13),0,c(15),0,c(17),0,c(19),0]
ccc=[0,c(2),0,c(4),0,c(6),0,c(8),0,c(10),0,c(12),0,c(14),0,c(16),0,c(18),0,c(20)]
bar(cc,'FaceColor','b','EdgeColor','black','LineWidth',0.0000000000000001)
```

```
hold on
bar(ccc,'FaceColor','r','EdgeColor','black','LineWidth',0.0000000000000001)
hold off
ylim([-1,1])
set(gca,'XTick',[], 'YTick', [])
b=strcat('E=',num2str(bval))
%text(20,-0.5, b)

bth=X1/2+2

[V,D]=eig(A(ut,vt,wt,epst));
a=V(1:X1,X1/2);
aval=D(X1/2,X1/2);
b=V(1:X1,bth);
bval=D(bth,bth);

figure
c=1*b'
cc=[c(1),0,c(3),0,c(5),0,c(7),0,c(9),0,c(11),0,c(13),0,c(15),0,c(17),0,c(19),0]
ccc=[0,c(2),0,c(4),0,c(6),0,c(8),0,c(10),0,c(12),0,c(14),0,c(16),0,c(18),0,c(20)]
bar(cc,'FaceColor','b','EdgeColor','black','LineWidth',0.0000000000000001)
hold on
bar(ccc,'FaceColor','r','EdgeColor','black','LineWidth',0.0000000000000001)
hold off
ylim([-1,1])
set(gca,'XTick',[], 'YTick', [])
b=strcat('E=',num2str(bval))
%text(20,-0.5, b)

bth=X1/2+3

[V,D]=eig(A(ut,vt,wt,epst));
a=V(1:X1,X1/2);
aval=D(X1/2,X1/2);
b=V(1:X1,bth);
bval=D(bth,bth);

figure
c=-1*b'
cc=[c(1),0,c(3),0,c(5),0,c(7),0,c(9),0,c(11),0,c(13),0,c(15),0,c(17),0,c(19),0]
ccc=[0,c(2),0,c(4),0,c(6),0,c(8),0,c(10),0,c(12),0,c(14),0,c(16),0,c(18),0,c(20)]
bar(cc,'FaceColor','b','EdgeColor','black','LineWidth',0.0000000000000001)
hold on
bar(ccc,'FaceColor','r','EdgeColor','black','LineWidth',0.0000000000000001)
hold off
```

```
ylim([-1,1])
set(gca,'XTick',[], 'YTick', [])
b=strcat('E=',num2str(bval))
%}
%{
Function definitions starts here
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
%}

%making eps function
function y=epsmat(cells,site)
    row=1
    column=1
    umat=[]
    N=2*cells
    while row<=N
        rmat=[]
        while column<=N
            if column==row & row==site
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
            column=column+1
        end
        umat=[umat;rmat]
        column=1
        row=row+1
    end
y=umat
end


%making heps function
function y=hepsmat(cells,site)
    row=1
    column=1
    N=cells*2
    y=[]
    while row<=N
        rmat=[]
        while column<=N
            if (column==row+1 & row==site | column==row-1 & column==site)
```

```
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
            column=column+1
        end
        y=[y;rmat]
        column=1
        row=row+1
    end
end
```

Appendix C

Simulation of Transport

```
%usefulfunctions
dag=@(M)ctranspose(M)
mag=@(v)(sqrt(sum(v.*v)))
normalize=@(v)v/mag(v)
phaseadjust=@(v,kth)v*(v(kth)/abs(v(kth)))

%Making hamiltonian
cells=10
N=2*cells

%making w
row=1
column=1
wmat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row-1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row+1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
```

```
            end
        end
        column=column+1
    end
    wmat=[wmat;rmat]
    column=1
    row=row+1
end

%making v
row=1
column=1
vmat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row+1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row-1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    vmat=[vmat;rmat]
    column=1
    row=row+1
end

%making u
row=1
column=1
umat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row
```

```
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row
                rmat=[rmat,-1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    umat=[umat;rmat]
    column=1
    row=row+1
end

%making eps/heps
hepssmat=hepsmat(cells,3)
epssmat=epsmat(cells,2)

M1=@(u,v,w)umat.*u+vmat.*v+wmat.*w
M2=@(u,v,w,epssite,eps,hepssite,heps)M1(u,v,w)+epsmat(cells,epssite).*eps+hepsmat(ce

%{
Hamiltonian construction ends here
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
%}

%A=@(u,v,w,eps)M2(u,v,w,20,eps,6,0)
A=@(u,v,w,eps)umat.*u+vmat.*v+wmat.*w+epssmat.*eps

%propagator
U=@(u,v,w,eps,dtau)expm((A(u,v,w,eps).*(-1.0i).*dtau));

eps=0.6

%evolution
Tpara=10000
wpara=2*pi/Tpara
taunaught=0
taufinal=10000
```

```
taudiv=1000
dtau=(taufinal-taunaught)/taudiv
t=[taunaught:dtau:taufinal-dtau]

v=@(t)-cos(t.*wpara)+1
u=@(t)-sin(t.*wpara)
w=@(t)1

%setting up state
counter=1
initialstate=[1]
while counter<N
    initialstate=[initialstate;0]
    counter=counter+1
end
counter=1
finalstate=[1]
while counter<N
    finalstate=[0;finalstate]
    counter=counter+1
end


tau=taunaught
state=initialstate
catedeige=[]
catedstate=[]
catedv=[]
catedu=[]
catedw=[]
catedenergy=[]
%i=1
while tau<taufinal
    catedeige=[catedeige,eig(A(u(tau),v(tau),w(tau),eps))]
    state=U(u(tau),v(tau),w(tau),eps,dtau)*state
    catedstate=[catedstate,abs(state)]
    %catedv=[catedv,v(tau)]
    %catedu=[catedu,u(tau)]
    %catedw=[catedw,w(tau)]
    catedenergy=[catedenergy,dag(state)*A(u(tau),v(tau),w(tau),eps)*state]
    %{
    bar(1:1:20,state)
    ylim([0 1])
    M(i)=getframe(gcf);
    i=i+1;
```

```
    %}
    tau=tau+dtau
end

f = figure;
p = uipanel('Parent',f,'BorderType','none');
p.Title = strcat('')
p.TitlePosition = 'centertop';
p.FontSize = 12;
p.FontWeight = 'bold';

subplot(2,1,1, 'Parent',p)
hold on
plot(t,catedeige)
plot(t,catedenergy,'o')
ylabel('Energy')
xlabel('Time\times w}$','interpreter','latex')


subplot(2,1,2, 'Parent',p)
mesh(t,1:1:length(state),abs(catedstate))
ylabel('|Wavefunction|')
xlabel('Time/$\displaystyle\frac{1}{\hbar w}$','interpreter','latex')
%{}
figure
hold on
plot(t,catedu)
plot(t,catedv)
plot(t,catedw)
xlabel('Time$\times$w','interpreter','latex')
%}
%{
Function definitions starts here
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
%}

%making eps function
function y=epsmat(cells,site)
    row=1
    column=1
    umat=[]
    N=2*cells
    while row<=N
```

```
        rmat=[]
        while column<=N
            if column==row & row==site
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
            column=column+1
        end
        umat=[umat;rmat]
        column=1
        row=row+1
    end
y=umat
end


function y=hepsmat(cells,site)
    row=1
    column=1
    N=cells*2
    y=[]
    while row<=N
        rmat=[]
        while column<=N
            if (column==row+1 & row==site | column==row-1 & column==site)
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
            column=column+1
        end
        y=[y;rmat]
        column=1
        row=row+1
    end
end
```

Appendix D

Fidelity of Transport Given Singular Defect

```
%usefulfunctions
dag=@(M)ctranspose(M)
mag=@(v)(sqrt(sum(v.*v)))
normalize=@(v)v/mag(v)
phaseadjust=@(v,kth)v*(v(kth)/abs(v(kth)))

%Making hamiltonian
cells=10
N=2*cells

%making w
row=1
column=1
wmat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row-1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row+1
                rmat=[rmat,1]
```

```
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    wmat=[wmat;rmat]
    column=1
    row=row+1
end

%making v
row=1
column=1
vmat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row+1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row-1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    vmat=[vmat;rmat]
    column=1
    row=row+1
end

%making u
row=1
column=1
umat=[]
while row<=N
    rmat=[]
    while column<=N
```

```
        if rem(row,2)==1
            if column==row
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row
                rmat=[rmat,-1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    umat=[umat;rmat]
    column=1
    row=row+1
end

%making eps/heps
hepssmat=hepsmat(cells,10)
epssmat=epsmat(cells,9)

M1=@(u,v,w)umat.*u+vmat.*v+wmat.*w
M2=@(u,v,w,epssite,eps,hepssite,heps)M1(u,v,w)+epsmat(cells,epssite).*eps+hepsmat(ce

%{
Hamiltonian construction ends here
--------------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
%}

%A=@(u,v,w,eps)M2(u,v,w,20,eps,6,0)
A=@(u,v,w,eps)umat.*u+vmat.*v+wmat.*w+hepssmat.*eps

%propagator
U=@(u,v,w,eps,dtau)expm((A(u,v,w,eps).*(-1.0i).*dtau));

%evolution
Tpara=10000
wpara=2*pi/Tpara
taunaught=0
taufinal=10000
```

```
taudiv=10000
dtau=(taufinal-taunaught)/taudiv
t=[taunaught:dtau:taufinal-dtau]

v=@(t)-cos(t.*wpara)+1
u=@(t)-sin(t.*wpara)
w=@(t)1

counter=1
initialstate=[1]
while counter<N
    initialstate=[initialstate;0]
    counter=counter+1
end

counter=1
finalstate=[1]
while counter<N
    finalstate=[0;finalstate]
    counter=counter+1
end

epsnaught=-2
epsfinal=2
epsdiv=60
deps=(epsfinal-epsnaught)/epsdiv
eps=epsnaught
catedfidelity=[]
while eps<epsfinal
    tau=taunaught
    state=initialstate
    i=1
    while tau<taufinal
        state=U(u(tau),v(tau),w(tau),eps,dtau)*state
        tau=tau+dtau
        %{
        if rem(i,10)==0
            bar(1:1:20,abs(state))
            txt = num2str(eps);
            text(10,0.5,txt)
            ylim([0 1])
            M(i)=getframe(gcf);
        end
        i=i+1;
        %}
```

```
    end
    catedfidelity=[catedfidelity,abs(dag(finalstate)*state)^2]
    eps=eps+deps
end

figure
plot(epsnaught:deps:epsfinal-deps,catedfidelity)

%{
Function definitions starts here
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
%}

%making eps function
function y=epsmat(cells,site)
    row=1
    column=1
    umat=[]
    N=2*cells
    while row<=N
        rmat=[]
        while column<=N
            if column==row & row==site
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
            column=column+1
        end
        umat=[umat;rmat]
        column=1
        row=row+1
    end
y=umat
end


%making heps function
function y=hepsmat(cells,site)
    row=1
    column=1
    N=cells*2
    y=[]
```

```
    while row<=N
        rmat=[]
        while column<=N
            if (column==row+1 & row==site | column==row-1 & column==site)
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
            column=column+1
        end
        y=[y;rmat]
        column=1
        row=row+1
    end
end
```

Appendix E

Fidelity of Transport Given Random Defects

```
%usefulfunctions
dag=@(M)ctranspose(M)
mag=@(v)(sqrt(sum(v.*v)))
normalize=@(v)v/mag(v)
phaseadjust=@(v,kth)v*(v(kth)/abs(v(kth)))

%Making hamiltonian
cells=10
N=2*cells

%making w
row=1
column=1
wmat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row-1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row+1
                rmat=[rmat,1]
```

```
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    wmat=[wmat;rmat]
    column=1
    row=row+1
end

%making v
row=1
column=1
vmat=[]
while row<=N
    rmat=[]
    while column<=N
        if rem(row,2)==1
            if column==row+1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row-1
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    vmat=[vmat;rmat]
    column=1
    row=row+1
end

%making u
row=1
column=1
umat=[]
while row<=N
    rmat=[]
    while column<=N
```

```
        if rem(row,2)==1
            if column==row
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
        else
            if column==row
                rmat=[rmat,-1]
            else
                rmat=[rmat,0]
            end
        end
        column=column+1
    end
    umat=[umat;rmat]
    column=1
    row=row+1
end

%making eps/heps
heps1mat=hepsmat(cells,1)
heps2mat=hepsmat(cells,2)
heps3mat=hepsmat(cells,3)
heps4mat=hepsmat(cells,4)
heps5mat=hepsmat(cells,5)
heps6mat=hepsmat(cells,6)
heps7mat=hepsmat(cells,7)
heps8mat=hepsmat(cells,8)
heps9mat=hepsmat(cells,9)
heps10mat=hepsmat(cells,10)
heps11mat=hepsmat(cells,11)
heps12mat=hepsmat(cells,12)
heps13mat=hepsmat(cells,13)
heps14mat=hepsmat(cells,14)
heps15mat=hepsmat(cells,15)
heps16mat=hepsmat(cells,16)
heps17mat=hepsmat(cells,17)
heps18mat=hepsmat(cells,18)
heps19mat=hepsmat(cells,19)

eps1mat=epsmat(cells,1)
eps2mat=epsmat(cells,2)
eps3mat=epsmat(cells,3)
eps4mat=epsmat(cells,4)
```

```
eps5mat=epsmat(cells,5)
eps6mat=epsmat(cells,6)
eps7mat=epsmat(cells,7)
eps8mat=epsmat(cells,8)
eps9mat=epsmat(cells,9)
eps10mat=epsmat(cells,10)
eps11mat=epsmat(cells,11)
eps12mat=epsmat(cells,12)
eps13mat=epsmat(cells,13)
eps14mat=epsmat(cells,14)
eps15mat=epsmat(cells,15)
eps16mat=epsmat(cells,16)
eps17mat=epsmat(cells,17)
eps18mat=epsmat(cells,18)
eps19mat=epsmat(cells,19)
eps20mat=epsmat(cells,20)

%this defects function is not used but serve as a master copy for the
%defects part for A
defects=@(strength)(heps1mat.*(rand*2-1)+heps2mat.*(rand*2-1)+heps3mat.*(rand*2-1)+h

B=@(u,v,w,strength)umat.*u+vmat.*v+wmat.*w+(heps1mat.*(rand*2-1)+heps2mat.*(rand*2-1

A=@(u,v,w)umat.*u+vmat.*v+wmat.*w

%{
Hamiltonian construction ends here
------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------------------------------------------------------
%}

%propagator
U=@(u,v,w,strength,dtau)expm((umat.*u+vmat.*v+wmat.*w+(heps1mat.*(rand*2-1)+heps2mat

%setting up state
counter=1
initialstate=[1]
while counter<N
    initialstate=[initialstate;0]
    counter=counter+1
end

counter=1
finalstate=[1]
```

```
while counter<N
    finalstate=[0;finalstate]
    counter=counter+1
end

%three loops loop time, loop random, loop strength

stri=0
strf=1
strdiv=20
dstr=(strf-stri)/strdiv

trialsno=50

taui=0
tauf=10000
taudiv=10000
dtau=(tauf-taui)/taudiv

Tpara=10000
wpara=2*pi/Tpara
v=@(t)-cos(t.*wpara)+1
u=@(t)-sin(t.*wpara)
w=@(t)1

str=stri
cated=[]
strstr=[stri:dstr:strf]
while str<=strf
    trial=1
    tobeaveraged=[]
    while trial<=trialsno
        tau=taui
        state=initialstate
        def=defects(str)
        while tau<=tauf
            state=(expm((A(u(tau),v(tau),w(tau))+def).*(-1.0i).*dtau))*state
            tau=tau+dtau
        end
        tobeaveraged=[tobeaveraged,abs(ctranspose(finalstate)*state)^2]
        trial=trial+1
    end
    cated=[cated,mean(tobeaveraged)]
    str=str+dstr
end
```

```
plot(strstr,cated)

%{
Function definitions starts here
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
%}

%making eps function

function y=epsmat(cells,site)
    row=1
    column=1
    umat=[]
    N=2*cells
    while row<=N
        rmat=[]
        while column<=N
            if column==row & row==site
                rmat=[rmat,1]
            else
                rmat=[rmat,0]
            end
            column=column+1
        end
        umat=[umat;rmat]
        column=1
        row=row+1
    end
y=umat
end


function y=hepsmat(cells,site)
    row=1
    column=1
    N=cells*2
    y=[]
    while row<=N
        rmat=[]
        while column<=N
            if (column==row+1 & row==site | column==row-1 & column==site)
                rmat=[rmat,1]
```

```
            else
                rmat=[rmat,0]
            end
            column=column+1
        end
        y=[y;rmat]
        column=1
        row=row+1
    end
end
```