Summer Internship Project
Report

Quantum walk of interacting Bosons in an optical lattice

Submitted by

S. Gauthameshwar
Int. MSc 2018
National Institute of Science Education and Research

Under the guidance of

Prof. Tapan Mishra
Reader-F

School of Physical Sciences
National Institute of Science Education and Research
Bhubaneshwar

Summer Internship 2022

Abstract

In this project, we reproduce the numerical results of quantum walks with one and two bosons on a 1D optical lattice of 20 sites as obtained by *Preiss et.al* [1]. Following the Continuous Time Quantum Walk (CTQW) formulation to simulate the quantum walk of a single boson in a lattice, we obtain the time evolution of the same. Moreover, by introducing a potential gradient in the lattice, we also study the phenomenon of Bloch oscillations using *Mathematica*. Continuing the simulation to a multiparticle basis, we use the python library *quspin* to simulate the dynamics of a pair of bosons that have on-site interaction. We reproduce the Hanbury Brown-Twiss (HBT) effect of bunching and anti-bunching of bosons on tuning the well depth of our optical lattice as experimentally seen in *Preiss et.al*. With an adequate literature survey, the underlying theory behind our simulations is also provided.

Contents

Chapter 1

Introduction

Quantum walk, the quantum mechanical analog of the classical random walk has gained huge research interests in recent times due to its promising applications in understanding fundamental physics and in building advanced algorithms in quantum computers. This concept was first introduced by *Y. Aharonov*[7] in 1993 where he showed the faster spreading of a quantum random walker as compared to its classical counterpart [1] due to the existence of quantum interference. In the context of fundamental physics, the quantum walk helps simplify many-body quantum dynamics by predicting signatures of many-body phenomenons in a smaller set of particles. It provides intuitive access to important many body features of systems such as localization caused by interactions or fermionization of bosons which scale to larger system sizes to manifest phenomena such as quantum phase transitions, quasi-particles, or superfluidity[1]. Thus, it is an indispensable tool to understand the fundamental physics of multi-particle systems.

Besides fundamental physics, discrete time quantum walks are also used in the field of quantum computation. With the scientific community going through a paradigm shift in the field of computer engineering and information theory by integrating the concepts of quantum physics in processing information, quantum walks provide the basic computational algorithms to implement these problems in a quantum computer. It acts as a connecting bridge between theoretical computer science and the physical devices that perform these calculations in a quantum computer. In fact, quantum walks have already been proposed as a universal model for performing any quantum computation[8].

The concept of quantum walk can be categorized into two. Discrete time

[1] Today scientists no longer use the word "random" in quantum walks since it is a unitary evolution which can be traced back to its initial state. Random processes, however, can't be traced back like so.

and continuous time quantum walks (DTQW and CTQW). Both have their differences and applications subjected to their definition.

Discrete time quantum walks are a stochastic process analogous to discrete walks done by a drunkard. Assuming the drunkard is drunk enough to not differentiate between left and right (and sober enough to walk around without falling), he randomly chooses one direction every step and moves along the line. The same intuition follows when we talk about a discrete QW. But here, we leave our drunkards state to be ambiguous and exist everywhere simultaneously. This is perfectly acceptable in QM as we can only be certain about the drunkard's position if we measure his position at an instant. Classically, choosing any random direction can be associated with tossing an unbiased coin and deciding one direction based on the outcome. In quantum analogy, we perform a coin toss by acting as a unitary operator on our drunkard's space. Weird as it may sound, DTQW is realizable experimentally. We can have a fermion with an up/down spin denoting the outcome of the "heads" and "tails" of our measurement. Each time we toss a quantum coin, we mix the heads and tails state to a superposed state that the particle will have as it walks. DTQW is primarily used in designing quantum algorithms as it involves fermions and its spin direct product with space of integers to propagate it in time. Search algorithms have been proposed based on this concept that gives a quadratic boost in finding the queries in data, graphs, etc.

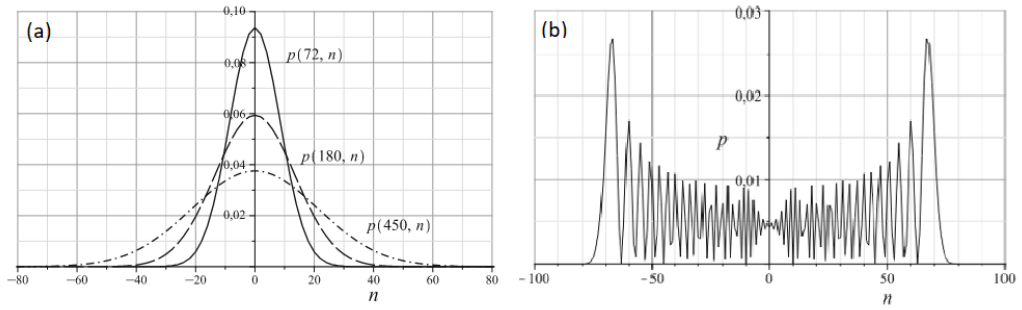


Figure 1.1: Comparison of classical and quantum diffusion over lattices [5]. (a) Classical walk of a single particle starting from origin in a lattice at different time steps. (b) CTQW of a single particle starting from origin in a lattice at t=100.

CTQW on the other hand involves the quantization of classical random walks. Classical random walks are the outcome of Markov processes that has a probability of having a transition probability to go from state i to j. Sending the limit of very large steps in a very small time interval, we arrive

at a continuum limit where the probability of being at state i at time t is analytically given in terms of the laplacian matrix exponential operated on an initial state. However, this classical treatment does not hold as we go to the quantum regime as we have to respect the uncertainty relations that govern the universe. So, appropriate quantization of states and operators leads to the Schrodinger equation analog of quantum walks where the laplacian of graph precisely plays the role of Hamiltonian. CTQW is seen in nature in many places, even in optical lattices with atoms performing the walk. So, it is often used to describe the physics of systems, unlike DTQW which is primarily used to design algorithms. We take the help of this formulation to derive the interesting physics that is observed in the paper we follow.

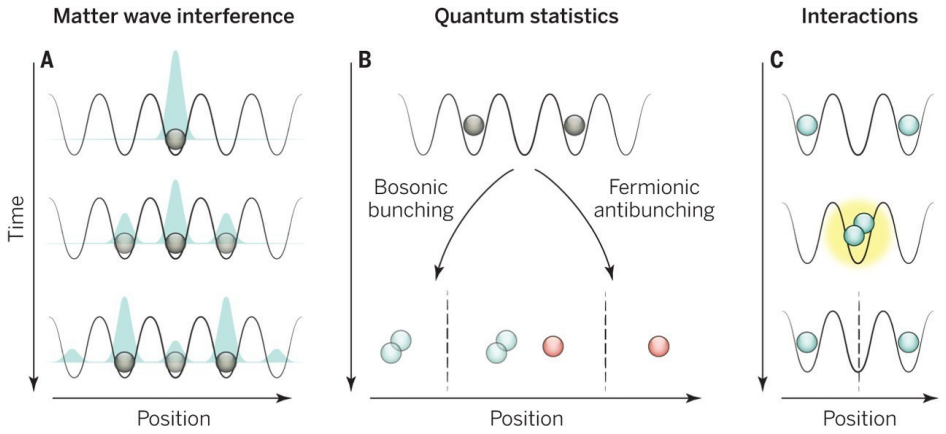


Figure 1.2: Types of physics when bosons perform QW in an optical lattice.

In our work, we will be reproducing quantum walks of ultracold bosonic atoms in optical lattices by *Preiss* [1]. We shall also investigate interesting physical properties such as the HBT effect, formation of repulsively bound pairs, phenomenon of Bloch oscillations, and simulate them using an open-source python library *quspin*.

The experiment we wish to simulate is an optical lattice that consists of potentials in space created by the interference of lasers. The interference leads to wells at destructive interference, and walls at constructive. This artificially made lattice is then filled with bosonic Rubidium atom ^{87}Rb in an ultracold regime. We define the dynamics of this system using a Bose-Hubbard model.

Implementing numerical methods involving exact diagonalization of our multiparticle Hilbert space, we get the time evolution of our system and predict the physics of bunching/ antibunching of adjacent bosons depending on the extent of interaction strength, also known as the Hardbury Brown-

Twiss (HBT) interference. We obtain Bloch oscillation in presence of a finite gradient in our lattice. These oscillations have a high degree of coherence as it restores to their initial state during oscillation. Another interesting physics happens when we place both bosons on the same site instead of adjacent sites. They form a repulsively bound pair that exhibits quantum walks as one particle with twice the frequency and half the spatial spreading of Bloch oscillation of a single boson. We shall now explain the theory of quantum walks in detail, and also touch upon the necessary background of the physics that we expect to observe before presenting the codes and simulation results.

Chapter 2

Theory

2.1 Discrete-time quantum walks

With a short introduction to what quantum walks are, we proceed to formally define them. Although we shall not be using DTQW much, we still provide a summary of it so that it acts as a motivation to dig deeper into more physical processes such as CTQW. DTQW is an algorithm (more than a physical process) where a random decision is made with the help of a unitary operator known as the "Quantum coin", and based on it the walker propagates across space. In general, we consider a space of $\mathcal{H}_p = |n\rangle$, n being an integer, as the space where our walker walks. Now, this walker is carrying a tag of possible outcomes as he walks. If it is a fermion, it can be the spin of the fermion that has the outcome of "up" or "down". This decision space $\mathcal{H}_c = |0\rangle, |1\rangle$ is therefore a space of two states. If the spin is up, our particle walks left, and if down, it moves right. Thus, the space of our particle $\mathcal{H}$ that we use to perform QW is the combined space of spin and integer space. The quantum coin acts only on the spin space, and the propagation $\hat{S}$ only happens in integer space depending on the state of the spin. Formally writing the above, we have:

$$\mathcal{H} = \mathcal{H}_c \otimes \mathcal{H}_p \tag{2.1}$$

$$\hat{S}|0, n\rangle = |0, n+1\rangle$$
$$\hat{S}|1, n\rangle = |1, n-1\rangle$$

Thus, summing over all sites, our propagation operator becomes:

$$\hat{S} = |0\rangle\langle 0| \otimes \sum_{n=-\infty}^{\infty} |n+1\rangle\langle n| + |1\rangle\langle 1| \otimes \sum_{n=-\infty}^{\infty} |n-1\rangle\langle n| \tag{2.2}$$

Now we are left with defining the quantum coin to perform a quantum walk. In principle, any unitary operator in $\mathcal{H}_c$ should do the job. However, it must be a unitary operator since the particle must preserve the norm (and thus the probability axioms), as it walks. Each unitary operator will produce its own unique QW in $\mathcal{H}_p$. However, one of the most commonly used coins is the Hadamard coin defined as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.3}$$

So at each step of the walk, we first flip the coin and then act the propagator on it to do the walk.

$$U = \hat{S}.(H \otimes I) \tag{2.4}$$

So, the state of our particle at time t is given by:

$$|\psi(t)\rangle = U^t|\psi\rangle \tag{2.5}$$

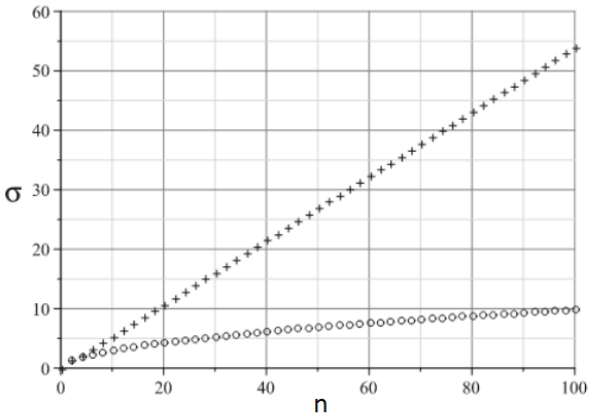


Figure 2.1: σ of Classical (o) and Quantum walks (+) vs iterations

Even though this algorithm is pretty straightforward, it has interesting properties. The walk is very different from the classical walk as at each step, the fermion is in a vague superposition of both up and down spin and hence it propagates faster to the edges than the classical walk where superposition doesn't make sense. The quantification of how "fast" the particle propagates in the lattice is defined by the standard deviation σ. $\sigma_{clasical} \sim \sqrt{t}$ whereas $\sigma_{quantum} \sim t$. Fig.2.1 shows the comparison of it. The probability distribution of our walk is extremely sensitive to the initial configuration of our spin

even if the initial position, the quantum coin, and everything else remain unchanged. Fig.2.2 shows the effect of different initial spin configurations. Another feature that doesn't exist in CW.

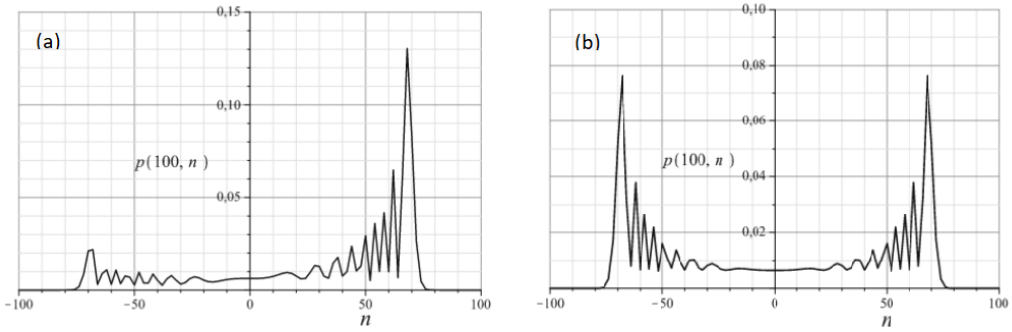


Figure 2.2: Probability distribution of DTQW with different initial spin configurations[5]. (a) QW with initial spin $|0\rangle$. (b) QW with initial spin $\frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$.

2.2 Classical walks and its quantization

Physical quantum walks are defined on a directed graph where the vertices play the role of the states our walker can be in, and the directed edges play the role of going from one state to the other. Now in this system, we introduce stochasticity (i,e) the walker takes a decision at each step of the walk and proceeds to the next state depending on the decision he takes. We consider one such stochastic process called the Markov process where the stochasticity is choosing one of the adjacent vertices with a probability corresponding to the weight of the directed edge.

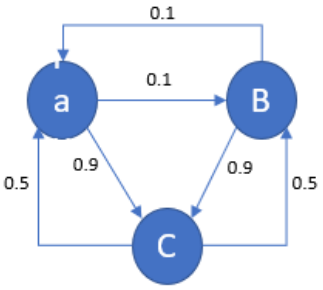
2.2.1 Classical Markov process

A Markov process is a stochastic process that has two main underlying properties:

1. The decision of going to $n+1$th state from nth state only depends on the decision taken at nth step and NOT on the decisions that he might've taken at previous states.

2. The next state is determined by randomly choosing a directed edge originating from his current state with a probability equal to the weight of the corresponding edge.

Fig. 2.3 shows a Markov process with 3 states and directed edges denoting transition between them. We can describe the state of the walker at time t



$$P = \begin{bmatrix} 0 & 0.1 & 0.9 \\ 0.1 & 0.0 & 0.9 \\ 0.5 & 0.5 & 0.0 \end{bmatrix}$$

Figure 2.3: An example of a Markov process with its corresponding transition matrix

by a state vector with its ith coefficient denoting the probability of being in the ith state.

$$\bar{p}(t) = \begin{bmatrix} p_1(t) \\ p_2(t) \\ \dots \\ p_n(t) \end{bmatrix} \tag{2.6}$$

Here t can be in steps, or continuous. We start with discrete t and then proceed to its continuum limit. From the second property of Markov processes, we can say the probability of reaching state 'i' at time t is the sum over all possible state j it could've been in time t-1 and then jumping to state i with probability T_{ij}. This can be written as:

$$p_i(t) = \sum_j T_{ij} p_j(t-1)$$

So, if we construct a transition matrix T with its coefficient T_{ij} denoting the probability of going from state j to i, we have the recursive stochastic equation:

$$\bar{p}(t+1) = M.\bar{p}(t) \tag{2.7}$$

We observe that the $t+1$th step only depends on its decision in the previous step and not on its far past decisions and so, retains the conditions of a Markov process. Now that we have written the entire process as a linear equation, we can write the state at time t in terms of the state at t=0 by using the properties of linear equations:

$$\bar{p}(t) = M^t.\bar{p}(0) \tag{2.8}$$

There are much more interesting analyses of a classical Markov process such as finding the stationary state that the walker will be after a very large number of steps by finding eigenvalues and eigenstates of T. But the provided background suffices to understand the continuum limit of classical walks and quantizing it to obtain the CTQW equations that we will be using in our simulation of QW of bosons.

2.2.2 Continuous classical walks and their quantization

In the system we wish to study, we have an optical lattice in which bosonic particles perform a walk by hopping from one site to another. Since our lattice has an inherent property of symmetry, the lattice site i is identical to sites i+1, i-1, i+2, and so on. Hence, we can use translational invariance to say the hopping rate from a site i to any of its adjacent sites must be equal. Let this hopping rate be γ. Now consider an infinitesimal time step ϵ. The probability of going to one of the adjacent sites is $p_{ij} = \epsilon\gamma$. If our site j has d_j neighbors then, our boson has the probability of $1 - d_j\epsilon\gamma$ of being in site j. Thus, our infinitesimal transition matrix $T(\epsilon)$ is given as:

$$T_{ij}(\epsilon) = \begin{cases} 1 - d_j\epsilon\gamma & \text{if } i = j \\ \epsilon\gamma & \text{if } i \neq j \text{ and adjacent} \\ 0 & \text{if } i \neq j \text{ and not adjacent} \end{cases} \tag{2.9}$$

With the above definition, we use the properties of the Markov process to write:

$$\begin{aligned} M_{ij}(t+\epsilon) &= \sum_k M_{ik}(t)M_{kj}(\epsilon) \\ &= M_{ij}(t)M_{jj}(\epsilon) + \sum_{k\neq j} M_{ik}(t)M_{kj}(\epsilon) \\ &= M_{ij}(t) - \epsilon\gamma d_j M_{ij}(t) - \epsilon\gamma \sum_{<k>}(-1)M_{ik}(t) \end{aligned} \tag{2.10}$$

where $< k >$ considers only the adjacent terms to i. We define a generating matrix H to simplify the above expression and rewrite it in terms of it:

$$H_{ij}(\epsilon) = \begin{cases} d_j & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and adjacent} \\ 0 & \text{if } i \neq j \text{ and not adjacent} \end{cases} \tag{2.11}$$

We note that H is the same as the laplacian of a graph with uniform hopping to adjacent sites. Eq. 2.10 in terms of H simplifies to:

$$M_{ij}(t+\epsilon) = M_{ij}(t) - \epsilon\gamma \sum_k H_{ik}.M_{kj}(t)$$
$$\implies \frac{M_{ij}(t+\epsilon) - M_{ij}(t)}{\epsilon} = -\gamma \sum_k H_{ik}.M_{kj}(t)$$

Thus, we arrive at the Master equation for our Markov process:

$$\frac{d}{dt}M(t) = -\gamma H.M(t) \tag{2.12}$$

$$\frac{d}{dt}\bar{p}(t) = -\gamma H.\bar{p}(0) \tag{2.13}$$

For an initial state with the initial condition $M_{ij}(t=0) = \delta_{ij}$, this yields:

$$M(t) = e^{-\gamma Ht} \tag{2.14}$$

$$\bar{p}(t) = e^{-\gamma Ht}\bar{p}(0) \tag{2.15}$$

If we look at M(t) in its eigenbasis, it has a diagonal form with its entries as an eigenvalue of the form $\sim e^{-\gamma\lambda_i t}$. Thus, in a classical Markov process, our walker's probabilities exponentially decay to its stationary eigenstate or oscillate between states depending on whether λ_i is real or imaginary respectively.
Quantising Eq. 2.13 demands us to replace $\frac{d}{dt}$ operator with $i\hbar\frac{d}{dt}$, consider γH as our Hamiltonian, and replace the probability vector by a state vector in the space of lattice sites $|N\rangle$. Assuming $\hbar = 1$, Eq. 2.13 takes the form of Schrodinger equation on quantizing:

$$i\frac{d}{dt}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle \tag{2.16}$$

Thus, our unitary time propagator analogous to M(t) in Eq. 2.14 becomes:

$$\hat{U}(t) = e^{-i\hat{H}t} \tag{2.17}$$

The quantum analog of Eq. 2.15 is given by:

$$|\psi(t)\rangle = e^{-i\hat{H}t}|\psi(0)\rangle \tag{2.18}$$

After time t, the probability of our boson being in state n is given by:

$$p_n(t) = |\langle n|\psi(t)\rangle|^2 = |c_n(t)|^2 \tag{2.19}$$

Eq. 2.16 can be solved numerically by defining the appropriate H where the diagonal terms define the potential energy of being in its site, and the off-diagonal terms are the hopping energy of going from one site to another. However, in our problem of QW by a single boson in a 1D lattice with hopping between the nearest sites, we can exactly solve the equation to get the analytical expression of state after time t. The 1D Hamiltonian is defined as:

$$H = \begin{cases} 2\gamma & \text{if } i = j \\ -\gamma & \text{if } i \neq j \text{ and adjacent} \\ 0 & \text{if } i \neq j \text{ and not adjacent} \end{cases} \tag{2.20}$$

If our particle is in the state $|n\rangle$, then:

$$H|n\rangle = 2\gamma|n\rangle - \gamma|n+1\rangle - \gamma|n-1\rangle$$

We can write the k power of H acting on $|0\rangle$ as:

$$H^k|0\rangle = \gamma^k \sum_{n=-k}^{k} (-1)^k \binom{2k}{k-n} |n\rangle \tag{2.21}$$

where $\binom{2k}{k-n}$ represents the possible ways to reach state $|n\rangle$ after k steps. So, using Eq. 2.18, and assuming our particle starts from $|0\rangle$, we get:

$$\begin{aligned} |\psi(t)\rangle &= e^{-iHt}|0\rangle \\ &= \sum_{k=0}^{\infty} \frac{(-it)^k}{k!} H^k|0\rangle \\ &= \sum_{k=0}^{\infty} \frac{(-it)^k}{k!} \gamma^k \sum_{n=-k}^{k} (-1)^k \binom{2k}{k-n} |n\rangle \\ &= \sum_{k=0}^{\infty} \sum_{n=-k}^{k} \frac{(-i\gamma t)^k}{k!} e^{i|n|\pi} \binom{2k}{k-n} |n\rangle \end{aligned}$$

Summing n from -k to k over k from 0 to ∞ is the same as summing n from -∞ to ∞ over k from $|n|$ to ∞. So we invert the sum to get:

$$|\psi(t)\rangle = \sum_{n=-\infty}^{\infty} \sum_{k=|n|}^{\infty} \frac{(-i\gamma t)^k}{k!} e^{i|n|\pi} \binom{2k}{k-n} |n\rangle \tag{2.22}$$

Now using the identity of the Bessels function:

$$e^{2ix} J_{|n|}(2x) = e^{i\frac{i}{2}|n|} \sum_{k=|n|}^{\infty} \frac{(-ix)^k}{k!} \binom{2k}{k-n} \tag{2.23}$$

we eliminate the sum over k in Eq. 2.22 to obtain:

$$|\psi(t)\rangle = \sum_{n=-\infty}^{\infty} e^{i\frac{\pi}{2}|n|-2i\gamma t} J_{|n|}(2\gamma t)|n\rangle \tag{2.24}$$

So the probability of being in state n is:

$$p_n(t) = |c_n(t)|^2 = |J_{|n|}(2\gamma t)|^2 \tag{2.25}$$

The probability is symmetric about n. So the boson propagates identically

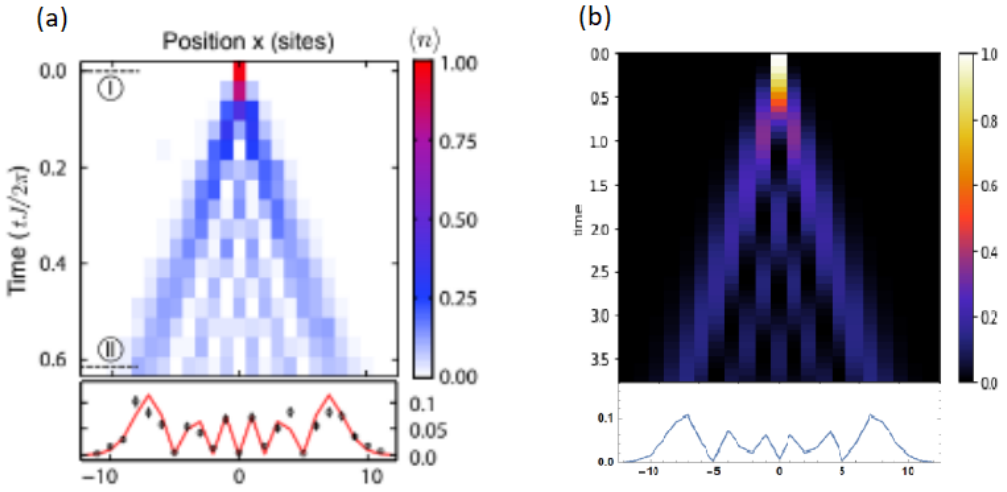


Figure 2.4: Experimental verification of the analytical probability distribution derived for a single particle CTQW. (a) Experimental result of a single Boson CTQW[1]. (b)Numerical plot of Eq. 2.25 with $\gamma = 1$ and $t = 3.7$.

on both sides of the lattice. Eq. 2.24 is confirmed by the experimental result of one boson QW by *Preiss et.al* as shown in Fig. 2.4 (a).

2.3 The Hanbury Brown-Twiss (HBT) interference

HBT interference is a phenomenon where our inherently quantum indistinguishable particles tend to group together or away from each other depending on the statistics they obey. Fundamentally, two kinds of indistinguishable particles exist that follow certain symmetries on exchange. Fermion is a class of particles where interchanging the position of two particles leaves a

phase of π to the combined two-particle wavefunction. Similarly, a boson is a class of particles where interchanging the position of two particles leaves the combined two-particle wavefunction invariant. To have a better physical understanding of this phenomenon, we present the experiment performed by *Jeltes et.al*[2] to understand these statistics better.

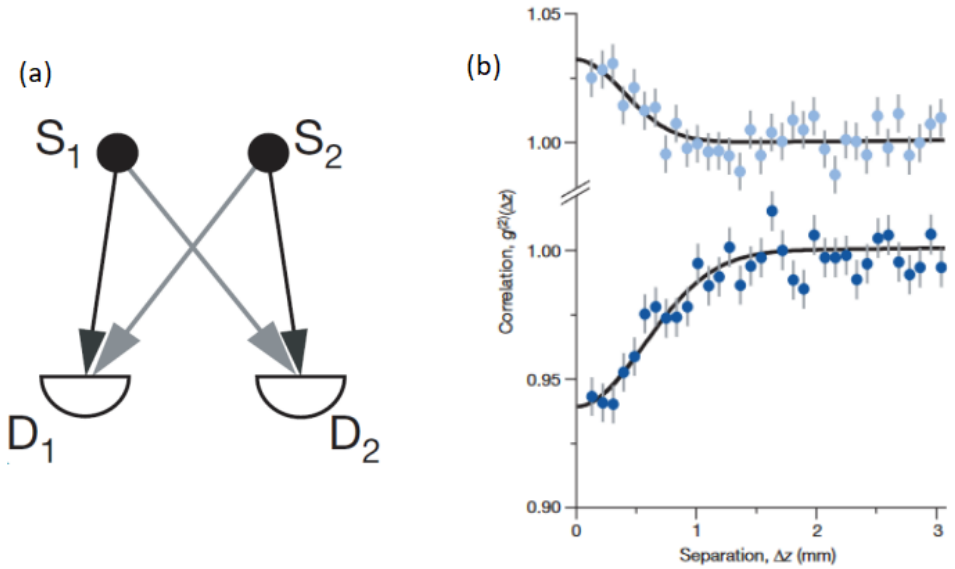


Figure 2.5: Experimental setup to show HBT effect in fermions and bosons[2]. (a) Schematic of the detector setup. Quantum particles are emitted from S1 and S2 while they are being detected at positions D1 and D2. (b) The correlation function obtained for fermions and bosons.

Consider Fig. 2.5 (a) where we have the detection of two particles from S1 and S2 that can reach any one of the sites D1 and D2. If it were a boson, then we can say if particle S1 hits D1 and S2 hits D2, the coefficient of the wavefunction remains invariant if we switch the position of the particles and make particle S1 hit D2 and S2 hit D1. The same switching of the particle comes with a pre-factor of π if it were a fermion. Mathematically, the same can be written as:

$$\psi_B(S1; D1, S2; D2) = \psi_B(S1; D2, S2; D1) \tag{2.26}$$

$$\psi_F(S1; D1, S2; D2) = -\psi_F(S1; D2, S2; D1) \tag{2.27}$$

So, if we try to measure the correlation between the two particles (effect of particle S1 given the state of S2), we end up superposing both these wave-

functions obtained after interchanging. So, if we look at a boson, the superposition leads to increased addition of the two wavefunctions and if $D1 = D2$, we have a strong probability of finding both the bosons at the same position and so, get a factor of 2. If it were fermions, the same superposition leads to reducing the coefficient of the wavefunction and leads to the annihilation of both the fermions if $D1 = D2$. But if we don't collapse both the particles at the same location, the same fermionic statistics is partially seen. Bosons tend to bunch together whereas fermions tend to repel each other. To quantify this effect for our particles, we define a correlation function:

$$g = \frac{P_{1,2}(D1, D2)}{P_1(D1)P_2(D2)} \tag{2.28}$$

where $P_{1,2}(D1, D2)$ is the probability of finding both particle 1 in D1, and particle 2 in D2 simultaneously, and $P_1(D1), P_2(D2)$ are individual probabilities of the particles hitting the respective detectors irrespective of where the other particle is. For $g > 1$ we get bunching as particles simultaneously tend to be in space together, and similarly for $g < 1$ we get anti-bunching. $g = 1$ represents the statistically independent motion of both particles. Fig.2.2 (b) shows the experimentally observed results of the bunching of bosons and anti-bunching of fermions. As we move the detectors farther away, both particles behave like statistically independent particles as their wavefunction starts to decohere in space. For closer spacing of detectors, we observe the predicted HBT effects in our respective particles. Theoretically, we expect g to be 2 for bosons and 0 for fermions. But since the experimental setup has an accuracy of 10% in detecting the particles, it is 1.3 for bosons and 0.94 for fermions[2].

2.4 The Bose-Hubbard Model

Earlier we have defined the Schrodinger equation of QW with its Hamiltonian having potential energy in its diagonal entries and hopping energy in its off-diagonal elements. In the Bose-Hubbard model, we have all those and in addition to it, we also have an interaction term that exists if more than one particles occupy the same site. In second quantization formalism, a boson can be created or annihilated by the operators $\hat{a}_j^\dagger$ and $\hat{a}_j$ respectively. These operators have the commutation property:

$$[\hat{a}_i, \hat{a}_j] = [\hat{a}_i^\dagger, \hat{a}_j^\dagger] = 0 \tag{2.29}$$

$$[\hat{a}_i, \hat{a}_j^\dagger] = \delta_{ij} \tag{2.30}$$

It can be derived from these commutation relations that more than one boson can exist on the same site. Thus, it is only apt to consider that bosons must have significant interaction with each other in the same site. A boson in the ith site can interact with one in site j, but we don't consider those here as they are not as dominant as the interaction between the bosons in the same site. The most general Bose-Hubbard Hamiltonian for such a system is:

$$\hat{H} = -J \sum_{<i,j>} \left(\hat{a}_i^\dagger \hat{a}_j + \hat{a}_j^\dagger \hat{a}_i \right) + \sum_i (\epsilon_i - \mu) + \frac{U}{2} \sum_i \hat{n}_i (\hat{n}_i - 1) \tag{2.31}$$

The first sum over all neighboring sites is the hopping from i to j and its hermitian conjugate is the hopping from j to i. The second term is the average energy of the lattice minus the chemical potential that is introduced to allow the freedom of changing the particle number in our lattice. The third term is the interaction of n bosons with n-1 bosons in the same site. $\hat{n}_i = \hat{a}_i^\dagger \hat{a}_i$ is the number operator of site i that returns the number of bosons in site i. J is analogous to the kinetic energy of the particle. U is the potential depth of each lattice site. As we deepen the potential well, we make the bosons interact strongly with each other.

Since we have the lattice created at ultra-low temperature and have good control of the particles being introduced in our lattice, we can set $\sum_i \epsilon_i = 0$ and $\mu = 0$. Simplifying our Hamiltonin by taking units of time in inverse hopping rate, we can non-dimensionalize the Hamiltonian. Additionally, we also apply a potential gradient in our lattice by tuning its magnetic field. This term increases as we traverse along i from left to right. So the final Hamiltonian we will consider becomes:

$$\hat{H} = -\sum_{<i,j>} \left(\hat{a}_i^\dagger \hat{a}_j + \hat{a}_j^\dagger \hat{a}_i \right) + \frac{u}{2} \sum_i \hat{n}_i (\hat{n}_i - 1) + \sum_i i \Delta \hat{n}_i \tag{2.32}$$

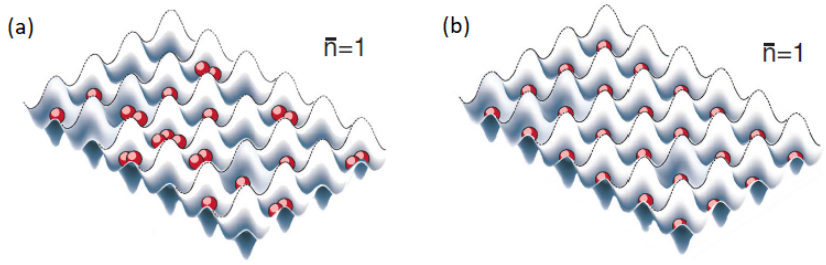


Figure 2.6: States of our bosons in an optical lattice[6]. (a) Superfluid state of a lattice with bosons. (b) Mott insulator state of a lattice with bosons

It is noteworthy to state that if we increase the potential beyond a threshold, we will have a phase transition from a superfluid bosonic system to a Mott insulator. A Mott insulator is a state of our lattice where each site is occupied with the same number of bosons and hopping from sites is forbidden. However, since we have no. of particles $n << D$ where D is the no. of potential wells in the lattice, we can safely assume our bosons are still in a superfluid state as they walk around our optical lattice.

2.5 Bloch oscillations

Bloch oscillation is a phenomenon where the wavepacket existing in a lattice experiencing a potential gradient oscillates about its mean position. Contrary to the classical intuitions where an electron accelerates towards the higher potential end of the lattice, the quantum mechanical analog of the same system leads to oscillations. This can be qualitatively explained by the fact that the wavefunction gets an additional phase shift of $\phi = e^{\frac{-iEt}{\hbar}}$ as it moves around the lattice, and at the end of each oscillation, it acquires a total phase of 2π, thus retaining its initial distribution. We shall however provide a rigorous derivation of the Bloch oscillation by following the Bloch theorem and applying it in our tight-binding Hamiltonian.

2.5.1 Tight-binding Hamiltonian on a k basis

The tight-binding H we shall consider is defined in the Hilbert space of $|n\rangle$ which represents the state of being in the nth site of the lattice.

$$\hat{H} = \sum_{n=-\infty}^{\infty} -J(|n\rangle\langle n+1| + |n+1\rangle\langle n|) + (\Delta n a)|n\rangle\langle n| \tag{2.33}$$

J is the hopping rate of our boson from site n to n+1 and vice versa. $na\Delta$ is the potential gradient at site n. It is zero at $|0\rangle$, decreases as we go left, and increases as we go right, in steps of $a\Delta$. a is the lattice spacing, and Δ is the strength of the gradient field applied. Since our lattice is periodic, we can exploit its translation invariance and write that our momentum basis acquires a phase of e^{ika} on translation by a in the lattice. This follows the Bloch theorem.

$$\langle n|k\rangle = \sqrt{\frac{a}{2\pi}} e^{inka} \tag{2.34}$$

$$\implies |k\rangle = \sqrt{\frac{a}{2\pi}} \sum_{n=-\infty}^{\infty} |n\rangle e^{inka} \tag{2.35}$$

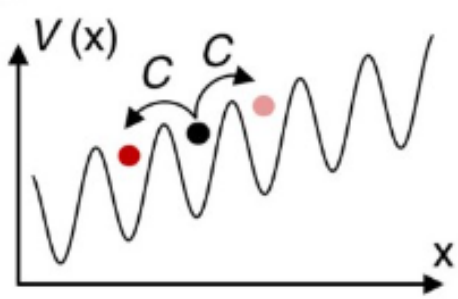


Figure 2.7: Potential of the lattice with a gradient along the lattice distance. Here a particle can hop to neighboring sites with rate 'c'.

So, the elements of H in k-basis can be written as:

$$\begin{aligned}
\hat{H}_{kk'} &= \langle k'| \left[\sum_{n=-\infty}^{\infty} -J(|n\rangle\langle n+1| + |n+1\rangle\langle n|) + (\Delta n a)|n\rangle\langle n| \right] |k\rangle \\
&= \frac{a}{2\pi} \left[-J(e^{ika} \sum_{n=-\infty}^{\infty} e^{in(k-k')a} + e^{-ika} \sum_{n=-\infty}^{\infty} e^{in(k-k')a}) + (\Delta a) \sum_{n=-\infty}^{\infty} n e^{in(k-k')a} \right] \\
&= -J(e^{ika}\delta(k-k') + e^{-ika}\delta(k-k') + (\Delta a)\frac{i}{a}\delta(k-k')\frac{d}{dk} \\
&= -J(e^{ika} + e^{-ika})\delta(k-k') + a\Delta\delta(k-k')\frac{d}{dk} \\
&= -2Jcos(ka)\delta(k-k') + i\Delta\delta(k-k')\frac{d}{dk}
\end{aligned} \tag{2.36}$$

Hence, we see that the Hamiltonian is diagonal on a k-basis. The eigenvalue equation on k-basis then becomes:

$$\frac{d}{dk}\psi(k) = -\frac{i}{\Delta}(E_k + 2Jcos(ka))\psi(k) \tag{2.37}$$

The solution of Eq 1.3 is of the form:

$$\psi(k) = c_0 e^{-i\left(\frac{Ek}{\Delta} + \frac{2Jsin(ka)}{a\Delta}\right)} \tag{2.38}$$

Applying Periodic Boundary Conditions (PBC) and normalizing the wave-function appropriately, we get:

$$E_m = ma\Delta \tag{2.39}$$

$$c_0 = \sqrt{\frac{a}{2\pi}} \tag{2.40}$$

The energy is now quantized (and is proportional to the potential gradient in lattice) in our system.

Putting all the above constraints into our wavefunction, we finally get:

$$\langle k|\psi\rangle = \sqrt{\frac{a}{2\pi}} e^{-i(mka+\gamma sin(ka))} \tag{2.41}$$

where $\gamma = \frac{2J}{a\Delta}$.

2.5.2 Unitary evolution of our state, and its time period and spatial spread

Now that we've done the difficult part of finding out the energy eigenfunctions of our tight-binding model, it is a matter of changing the basis to obtain the unitary operator in n-basis from our eigenbasis (k-basis). In the diagonal basis, U(t) is of the form:

$$U(t) = \sum_m e^{\frac{iE_m t}{\hbar}} |\psi_m\rangle\langle\psi_m| \tag{2.42}$$

Rewriting the same in the n-basis, we get:

$$U_{nn'}(t) = \sum_m e^{\frac{iE_m t}{\hbar}} \langle n'|\psi_m\rangle\langle\psi_m|n\rangle \tag{2.43}$$

To obtain $\langle n|\psi_m\rangle$, we simplify the expression:

$$\begin{aligned}
\langle n|\psi_m\rangle &= \int_{-b/2}^{b/2} dk\ \langle n|k\rangle\langle k|\psi_m\rangle \\
&= \frac{a}{2\pi}\int_{-b/2}^{b/2} dk\ e^{i((n-m)ka-\gamma sin(ka))} \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} dk\ e^{i((n-m)u-\gamma sin(u))} \\
&= J_{n-m}(\gamma)
\end{aligned} \tag{2.44}$$

where $J_n(x)$ is the Bessel function of the nth kind. The last step is obtained since the integral is another form of the Bessels function[3]. So, we proceed to simplify Eq 2.43 by resubstituting the integral in Eq 2.44, and obtaining:

$$U_{nn'}(t) = J_{n-n'}\left(2\gamma sin\left(\frac{\omega_B t}{2}\right)\right) e^{i\frac{n-n'}{2}(\pi-\omega_B t)-in'\omega_B t} \tag{2.45}$$

where $\omega_B = \frac{2\pi}{T_B}$ and, the time period of the Bloch oscillations is given by:

$$T_B = \frac{2\pi\hbar}{a\Delta} \tag{2.46}$$

Thus, if we have our particle at $|0\rangle$, by substituting $n' = 0$ in Eq 2.45, we have the coefficient of state $|n\rangle$ after time t as:

$$c_n(t) = J_n\left(2\gamma sin\left(\frac{\omega_B t}{2}\right)\right) e^{i\frac{n}{2}(\pi - \omega_B t)} \tag{2.47}$$

Our particle now spreads as proportional to the value of $J_n\left(2\gamma sin\left(\frac{\omega_B t}{2}\right)\right)$ in the lattice. From the properties of the Bessels function, the function has significant value when:

$$2\gamma\left|sin\left(\frac{\omega_B t}{2}\right)\right| < |n| \tag{2.48}$$

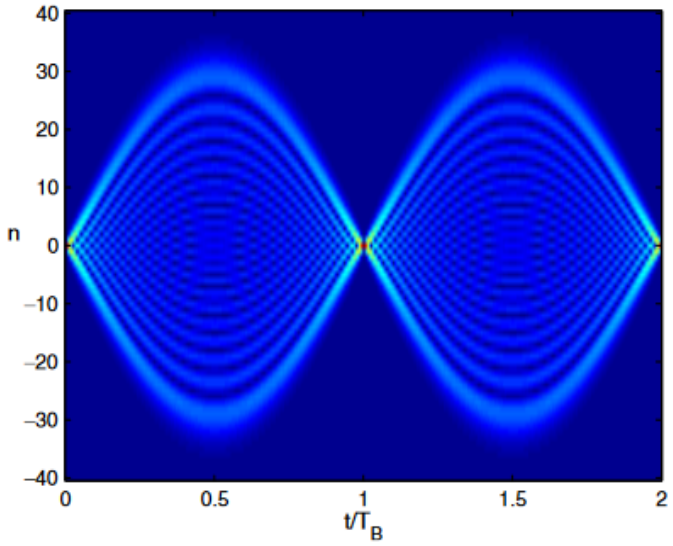


Figure 2.8: Lattice spacing vs Time heatmap of the wavefunction probability with boson at the center[4]. $\gamma = 15.8$

Fig. 2.8 shows the Bloch oscillation of a boson starting at the origin. We see that when $\left|sin\left(\frac{\omega_B t}{2}\right)\right| = 1$, we have the largest spread of $|n| = 2\gamma$. So, the Bloch oscillation spreads up to 2γ lattice sites on either side. And after some time, the particle comes back to the origin.

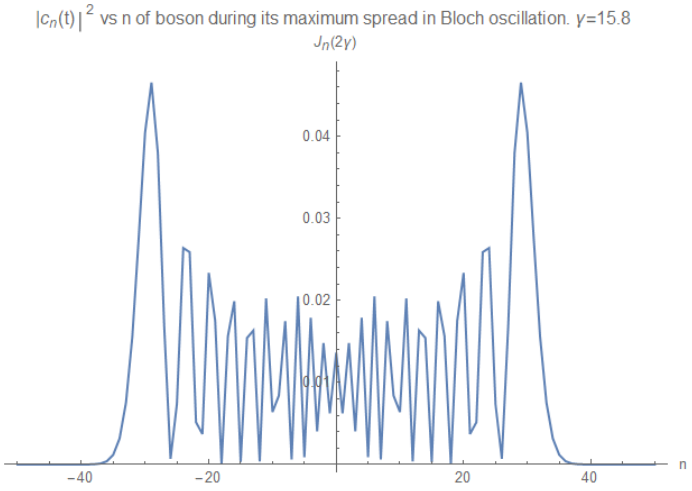


Figure 2.9: Numerical plot of the derived Eq. 2.48 at time $t/T_B = 0.5$. For $\gamma = 15.8$, we have $|n_{max}| = 2\gamma \approx 31$ sites. After 31 sites, the probabilities quickly decay as predicted.

Chapter 3

Results obtained

In this section, we proceed to simulate the physics of particles as described in chapter 3. For single particle QW, we simulate using Mathematica, and also in python. In python, we use a library for simulating Bose-Hubbard model systems called *quspin*. Its documentation can be found here.

We shall explain the codes and their syntax as we mention them:

3.1 Single Boson CTQW in our optical lattice

Our aim here is to get the time-evolved state of our boson in the lattice described by the Bose Hubbard Model Hamiltonian in Eq. 2.32. To define such a Hamiltonian, we first require to define the state vectors in the Fock space. Since we only have only one particle, each lattice only has either single occupancy or no occupancy on the Fock basis. So, our life becomes extremely simple and we can consider a vector of length L, L being the length of our lattice, as our fock basis. As an example, if we have a boson in the second lattice site then, the corresponding vector we define becomes:

$$|0,1,0,0...,0\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ ... \\ 0 \end{bmatrix}$$

On this basis, our Hamiltonian is a $L \times L$ matrix. Diagonal terms represent the potential of the boson by being in the ith site while the off-diagonal terms represent, by definition of a matrix, a transformation from state i to

j, in other words, hopping from i to j. Thus, the coefficient of $\hat{a}_i^\dagger \hat{a}_j$ terms take up the off-diagonal entries and $-u/2 + i\Delta$ take the ith diagonal entries. Since we only have one particle, it is acceptable to drop the interaction term in Eq. 2.32.

We shall implement the above Hamiltonian after defining the lattice in fock states in *Mathematica* as follows:

```
1  G = Graph[{1 \[UndirectedEdge] 2, 2 \[UndirectedEdge] 3,
2     3 \[UndirectedEdge] 4, 4 \[UndirectedEdge] 5,
3     5 \[UndirectedEdge] 6, 6 \[UndirectedEdge] 7,
4     7 \[UndirectedEdge] 8, 8 \[UndirectedEdge] 1}]
5  L = 2 IdentityMatrix[8] - AdjacencyMatrix[G];
6  state = {1, 0, 0, 0, 0, 0, 0, 0};
7  psi = MatrixExp[-I L t].state;
8  Manipulate[
9   ListPlot[Abs[psi]^2 /. t -> u, PlotRange -> {0, 1.1},
10   AxesLabel -> {"site i", "\!\(\*SubscriptBox[\(P\), \(i\)]\)(t)"},
11   Filling -> Axis], {u, 0, 20}]
```

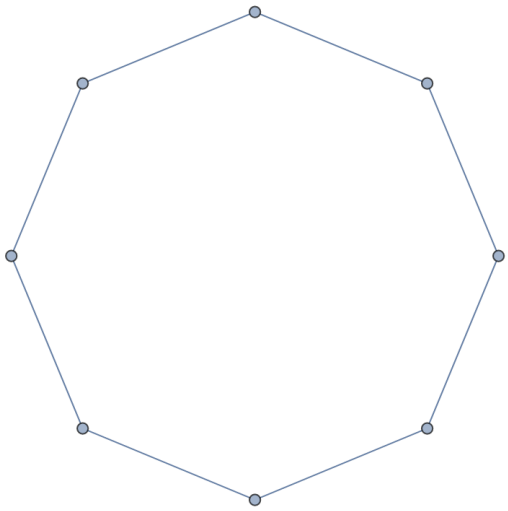


Figure 3.1: The graph in which we simulate CTQW with PBC.

In Line 1-4, we defined the lattice in which our boson exists. Each connected edge represents a hopping between sites described by the vertices. Fig. 3.1 shows the graph we consider to perform CTQW based on Markov processes discussed in section 2.2. Every edge has a unit weight and we consider periodic boundary conditions (PBC) by connecting the last vertex to

the first. Line 5 is where we define the Laplacian of our graph which also plays the role of the system Hamiltonian. Then in line 6, we initialize the state that we wish to evolve in time. Here, we consider the particle in the first site at the initial time. Then, we get the time evolved state by using Eq. 2.18 in line 7 and plotting its corresponding probability in line 8 using Eq. 2.19. The result is the animation of our p_i with time. Our wavefunction interferes with itself with time due to PBC and as a consequence, it oscillates between states 1 and 4 with time. Fig. 3.2 shows the probability distribution of our boson doing CTQW at different times.

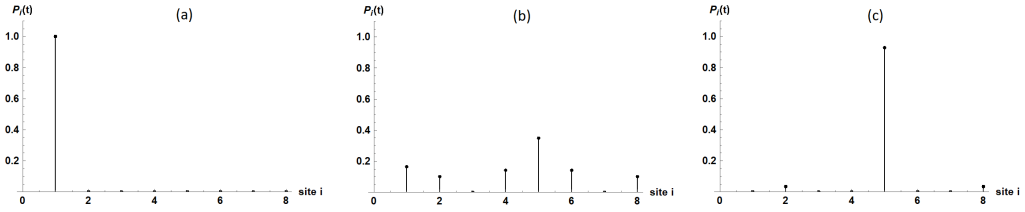


Figure 3.2: $p_i(t)$ vs i in a lattice of 11 sites. (a) Snapshot of $p_i(t)$ at $t = 0$, (b) $t = 3.2$, and (c) $t = 6.5$

The above code simulated a toy model of a small lattice and its CTQW. Now, we shall introduce the precise potential and gradient term in a 21-site long lattice that was experimentally realized by *Preiss*. The code is as follows:

```
1  lattice = Graph[Table[i \[UndirectedEdge] i + 1, {i, -10, 9}]]
2  A = lattice // AdjacencyMatrix // Normal;
3  \[CapitalDelta] = .4;
4  U = 4;
5  L = Table[
6      If[i == j, -1/2 U + \[CapitalDelta] i, 0], {i, 21}, {j, 21}] - A;
7  state4 = Table[If[i == 10, 1, 0], {i, 21}];
8  Manipulate[
9   ListPlot[Abs[MatrixExp[I L t].state4]^2, PlotRange -> {0, 1.1},
10   AxesLabel -> {"site i", "\!\(\*SubscriptBox[\(P\), \(i\)]\)(t)"},
11   Filling -> Axis], {t, 0, 20}]
```

We have considered open boundary conditions (OBC) in our lattice defined in line 1 as shown in fig. 3.3. In lines 5-6, we define the Hamiltonian of our matrix which has unit hopping between sites (defined by adjacency

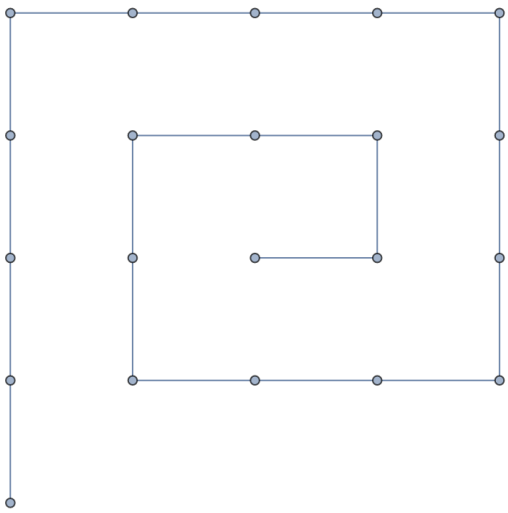


Figure 3.3: The graph in which we simulate CTQW with OBC.

matrix) and the diagonal terms $H_{ii} = \frac{-U}{2} + i\Delta$. Our particle is at the center of our lattice (i,e) in the 11th site initially. Then in line 9, we plot the probability of the time-evolved state with time varying from 0 to 20. Fig. 3.4-3.6 shows the snapshots of the time evolution of probability distribution at different times.

In the above code, we can also introduce Bloch oscillations as described in section 2.5 by changing Δ. For $\Delta = 0$, we have a normal quantum walk spreading to the edges of the lattice and reflecting. But if we increase the gradient to $\Delta = 0.4$, we observe Bloch oscillations that spread just as much as the edges of the lattice on each side and come back to their initial state without decohering. From Eq. 2.48, we have the maximum spatial spreading of Bloch oscillations to be $|n_{max}| = 2\gamma = \frac{4J}{a\Delta} = \frac{4}{\Delta}$ since we considered unit lattice spacing and hopping rate. As we increase Δ, the time period of oscillation keeps reducing as $T_B = \frac{2\pi}{\Delta}$ and is also in agreement with our simulation as shown in Fig. 3.4 and 3.5. Thus, our theoretical formula is confirmed in our simulation. For $\Delta < 0.4$ we don't recover the initial state perfectly as expected in Bloch oscillations since the state continues to spread beyond the edges and so, the reflected wave interferes with the existing state in sites and decoheres it. Fig. 3.6 shows this result.

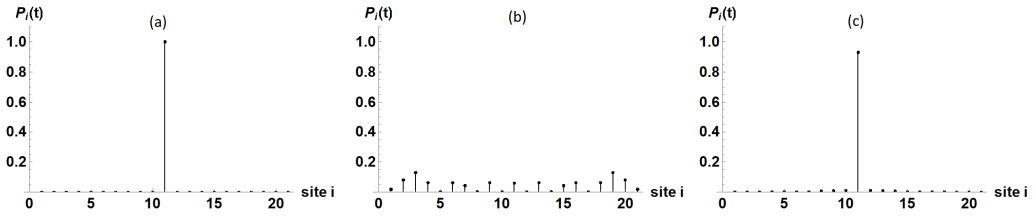


Figure 3.4: $p_i(t)$ vs i for $\Delta = 0.4$ in a lattice of 21 sites. $T_B = \frac{2\pi}{\Delta} = 15.71$, $|n_{max}| = \frac{4}{\Delta} = 10$. (a) shows snapshot of $p_i(t)$ at $t = 0$, (b) $t = 7.85$, (c) $t = 15.71$.

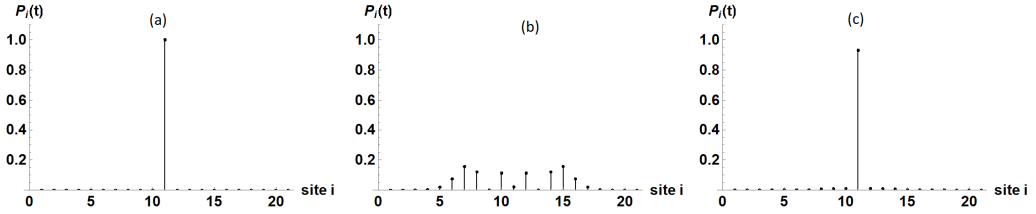


Figure 3.5: $p_i(t)$ vs i for $\Delta = \frac{\pi}{4}$ in a lattice of 21 sites. $T_B = \frac{2\pi}{\Delta} = 8$, $|n_{max}| = \frac{4}{\Delta} \approx 5$. (a) shows snapshot of $p_i(t)$ at $t = 0$, (b) $t = 4$, (c) $t = 8$.

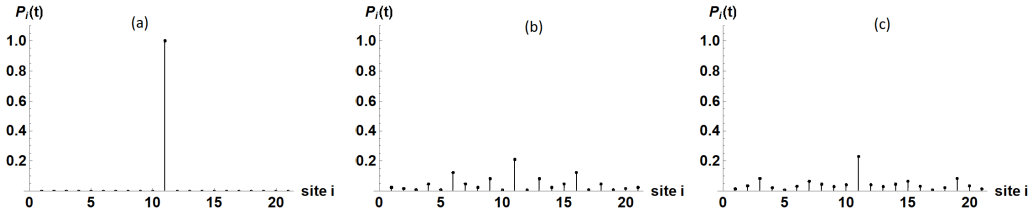


Figure 3.6: $p_i(t)$ vs i for $\Delta = 0.2$ in a lattice of 21 sites. $T_B = \frac{2\pi}{\Delta} = 31.4$, $|n_{max}| = \frac{4}{\Delta} = 20$. (a) shows snapshot of $p_i(t)$ at $t = 0$, (b) $t = 15.7$, (c) $t = 31.4$.

To reproduce the plots in *Preiss et.al[1]*, we perform the same simulation of quantum walks in python using *quspin*. We use this library to define the Hamiltonian in the bosonic Fock basis of a single particle. Although the computation in the single particle case could be done without calling the library, we still take this as an opportunity to introduce the syntax used in this library to extend it to simulate two particle bosonic walks that is not so straightforward in terms of defining the Fock state and its Hamiltonian in its basis.

Importing the necessary libraries to be used:

1. We import the class hamiltonian from quspin.operators to use it to define Hamiltonians on a multiparticle basis
2. We import boson_basis_1d from quspin.basis to make a multiparticle basis for our system in one dimension.
3. We import NumPy and expm from SciPy to use them in linear algebra and matrix exponentiation respectively.

```
from quspin.operators import hamiltonian # Hamiltonians and operators
from quspin.basis import boson_basis_1d # Hilbert space boson basis
import numpy as np # generic math functions
from scipy.linalg import expm
```

Then after defining the system parameters of our lattice, we create our basis states by calling *boson_basis_1d()* and inputting the size of our lattice and the total no. of bosons in our lattice. So, if we go ahead and print the basis we have created above, we will see all the possible Fock states a boson can have with the sum of all particles in the lattice sites $\sum_i n_i = 2$.

```
##### define model parameters #####
L = 21 # system size
J = 1.0 # hopping strength
U = 1.0 # interaction
grad = 0.0  # potential gradient

basis = boson_basis_1d(L,Nb=1) # ... + zero momentum and positive parity
print(basis)
```

The first five states of the output as seen in the terminal are given below:

```
reference states:
array index   /   Fock state   /   integer repr.
      0.          |1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>          1048576
      1.          |0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>           524288
      2.          |0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>           262144
      3.          |0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>           131072
      4.          |0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>            65536
      5.          |0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>            32768
```

Now we define the Hamiltonian as in Eq. 2.32. The helpful syntax from this library makes us define our Hamiltonian almost effortlessly if we follow the formula itself. The class hamiltonian takes a double list of the form [[...]...] as input to be defined. The double list is defined as such: [[coefficient of the term, indices of the term] over the sites where the summation of this term exists]. While inputting in the class, we have additional formatting of the form: ['type of indices', corresponding double list]. The type of index can be, for example, a creation at site i ('+'), annihilation at site i ('-'), or number operation at site i ('n'). So, we follow the above format and define the Hamiltonian in Eq. 2.32 term by term.

1. Hopping term $\hat{a}_i^\dagger \hat{a}_j$ are defined with only nearest neighbor terms for i and j. We also refrain from hopping bosons between the last and first site to respect OBC. So we have a summation only till the second last site,

2. Interaction term is also a two-index term like the hopping one but with both indices at the same site. So we define it accordingly for all sites in the lattice.

3. Similarly we also define the potential and gradient terms with a single index corresponding to the potential in site i.

After that, we define the type of each index in each term in a static array and input it into our class hamiltonian. The static array has both front and back hopping from the site i by defining the Hermitian conjugate of the same hop term. We do not have any time-dependent term in our Hamiltonian so we input an empty list in its slot. The hamiltonian is made in the basis we had defined earlier so we input it as well. Behind the screen, our hamiltonian class takes all these inputs, computes all non-zero actions of our terms in the static array, finds their index in the basis we provided, and makes the Hermitian matrix of the above!

```
# define site-coupling lists
hop=[[-J,i,(i+1)] for i in range(L-1)] # PBC
interact=[[0.5*U,i,i] for i in range(L)] # U/2 \sum_j n_j n_j
pot=[[-0.5*U,i] for i in range(L)] # -(\mu + U/2) \sum_j j_n
gra = [[i*grad, i] for i in range(L)] # i*E n_i

# define static and dynamic lists
static=[['+-',hop],['-+',hop],['n',pot],['nn',interact], ['n', gra]]
```

```
dynamic=[]

# build Hamiltonian
H=hamiltonian(static,dynamic,basis=basis,dtype=np.float64)
H=H.toarray()
```

Now we define the number operator that returns the average number of particles in site i, given state *psi*. Then after initializing the boson in the middle of the lattice, we define the time intervals at which the state is evaluated and store the wavefunction at each of these times (using Eq. 2.18) in an array *psi_t*.

```
# number operator
def nop(i, psi):
    nop=[[1, i]]
    static=[['n', nop]]
    n = hamiltonian(static, [], basis=basis)
    return psi.dot(n.dot(psi.conjugate()))

# defining the timesteps and the state evolution
ti=0
tf=0.6*2*np.pi # 16 for interference of the walk
steps=50

# stores the time value at each evaluation of Unitary evolution
tarr=np.linspace(ti, tf, steps)

# initial state: boson in middle of the lattice
psi_0 = np.zeros(basis.Ns)
psi_0[L//2] = 1

# array to store the probability of being at the ith site as time moves
psi_t=[]

# time evolution
for i in range(steps):
    UU=expm(-1j*H*tarr[i])
    psi=UU.dot(psi_0)
    psi_t.append([nop(i, psi) for i in range(L) ])
```

We plot the above number operator expectation value as it evolves in time.

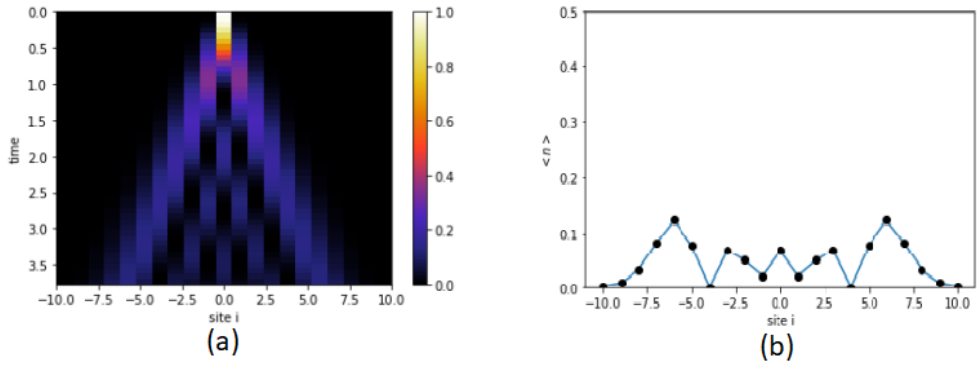


Figure 3.7: Simulated results. (a) shows the time evolution of the expected number operator and (b) shows the wavefunction after time evolution

```
# plot the probability map
import matplotlib.pyplot as plt
plt.imshow(np.array(psi_t).real, origin='upper',
cmap='CMRmap', extent=(-10,10,tf,ti))
plt.axis('tight') # what is this?
plt.xlabel('site i')
plt.ylabel('time')
plt.colorbar()
plt.show()

# plotting the state after evolution
plt.figure()
plt.plot([i-L//2 for i in range(L)], [nop(i,psi) for i in range(L)])
plt.plot([i-L//2 for i in range(L)], [nop(i,psi) for i in range(L)],
'k o')
plt.ylim([0,0.5])
plt.ylabel('$<n>$')
plt.xlabel('site i')
plt.show()
```

The obtained plots are shown in Fig. 3.7. They are in good agreement with the experimentally obtained quantum walk by *Preiss* (Fig. 3.8).

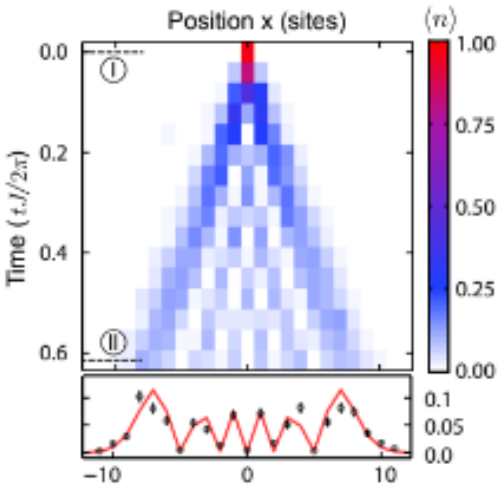


Figure 3.8: Experimentally obtained results by Greiner

By changing the grad to 0.5 and re-running the entire code, we get Bloch oscillations which are also as observed by Greiner in Fig. 3.9.

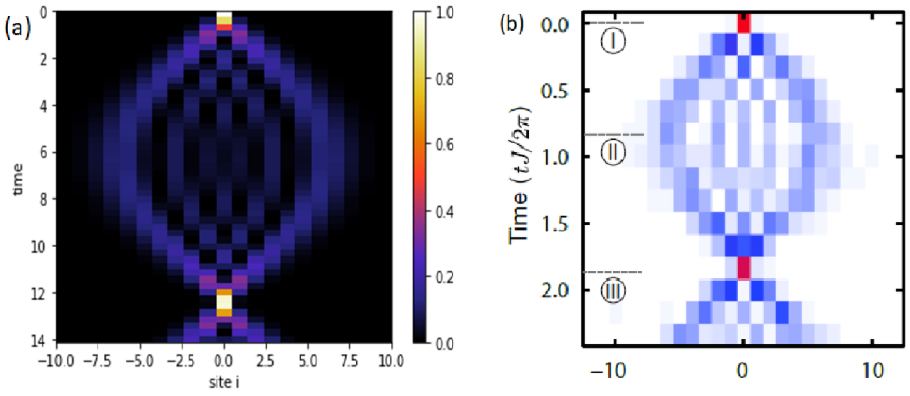


Figure 3.9: Comparison of simulated and experimental results of single boson Bloch oscillations. (a) shows the simulated Bloch oscillation of our boson with $\Delta = 0.5$ while (b) shows experimentally observed Bloch oscillations

We have now simulated all the physics observed in a single bosonic QW by Greiner!

3.2 Two Boson CTQW in our optical lattice

In two boson statistics, we have the possibility of influencing another particle by your current dynamics, if you were a particle in the optical lattice. This sort of long-range connection between particles is called correlation. We have already provided an example of such correlation in HBT interference where particles tend to bunch/anti-bunch due to the statistics they follow. Here, in the case of our bosons, if we keep two particles in adjacent sites initially and propagate them in time, they tend to bunch around and move like superfluid condensate if the potential well isn't too deep. However, if we deepen the potential well, the interaction term dominates and tries to make the particle existence unstable if close together. This is a very similar effect observed in fermions where they have an inherent Pauli exclusion principle causing the same effect caused by interaction in the case of bosons. We simulate the above in a 21-site lattice and find the correlation between sites (i, j). In second quantisation notation, we find the ijth expectation value of the correlation operator: $\Gamma_{ij} = \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_i \hat{a}_j$.

We define the basis and the BHM hamiltonian as done in a single particle case but now with *Nb=2* since we consider a system with two bosons.

```
from quspin.operators import hamiltonian  # Hamiltonians and operators
from quspin.basis import boson_basis_1d  # Hilbert space boson basis
import numpy as np  # generic math functions
from scipy.linalg import expm
import matplotlib.pyplot as plt
import matplotlib.colors


L = 21  # system size
J = 1.  # hopping
U = 5.1 # interaction
lamda = 0.0 # gradient term

basis = boson_basis_1d(L, Nb=2)  # Nb: total number of particles

print(basis)

# define site-coupling lists
hop = [[-J, i, (i + 1)] for i in range(L - 1)]
interact = [[0.5 * U, i, i] for i in range(L)]
```

```
pot = [[-0.5 * U, i] for i in range(L)]
grad= [[i*lamda, i] for i in range(L)]
# define static and dynamic lists
static = [['+-', hop], ['-+', hop], ['n', pot], ['nn', interact], ['n', grad]]
dynamic = []
# build Hamiltonian
H = hamiltonian(static, dynamic, basis=basis, dtype=np.float64)

H = H.toarray()
```

The first five two boson basis in the terminal is shown below:

```
reference states:
array index   /   Fock state   /   integer repr.
       0.          |2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>          69735688
       1.          |1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>          46490458
       2.          |1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>          38742048
       3.          |1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>          36159245
       4.          |1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>          35298311
       5.          |1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0>          35011333
```

Now we define one particle in the 10th site and the other particle in the 11th site. The corresponding Fock state of the particle becomes:

$$|\psi_0\rangle = |0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0\rangle$$

We search for the index of this state in our basis and set that coefficient in our initial $|\psi_0\rangle$ to 1. Having defined the initial state, we evolve the wavefunction in time as in a single particle by exponentiating the matrix.

```
# we find 2 particles next to each other in middle state
s='0'*(L//2)+'11'+'0'*(L//2-1)
i0 = basis.index(s)

psi0 = np.zeros(basis.Ns)
psi0[i0] = 1

ti = 0
tf = np.pi
Nt = 50
ts = np.linspace(ti, tf, Nt)
```

```
dt = ts[1] - ts[0]

UU = expm(-1j * H * dt)

for n in range(Nt):
    # print(n)
    if n==0:
        psi = psi0
    else:
        psi = UU.dot(psi)
```

Now to find the correlation term coefficient between all sites (i, j) in our lattice, we define the correlation operator in a doubly nested loop and find its expectation value in $|\psi(t)\rangle$. By doing so, we obtain the correlation on a particle in site j by a particle in site i, for all positions of i and j in our lattice. We also plot the resulting correlation matrix and the final state. Below is the code implementation of it:

```
# correlation fn using correlator operator exactly defined
corr=[]
for i in range(L):
    for j in range(L):
        #define the correlator fn
        a = [[1., i, j, i, j]]
        a = [['++--', a]]

        correlator = hamiltonian(a, [], basis=basis)
        del[a]

        corr.append(psi.dot(correlator.dot(psi.conjugate())))

corr=np.array(corr).reshape(L,L)

cvals  = [1, 2, 3, 4]
colors = ["white","blue","purple","red"]

norm=plt.Normalize(min(cvals),max(cvals))
tuples = list(zip(map(norm,cvals), colors))
gautam = matplotlib.colors.LinearSegmentedColormap.from_list("", tuples)

```

```
# plotting the correlation function
img=plt.imshow(np.array(corr).real, origin='lower', cmap=gautam, extent=(-L//2, L//2, -L//2, L
plt.axis('tight')
clb = plt.colorbar(img)
clb.ax.set_title('<n>')
plt.xlabel('Site i')
plt.ylabel('Site j')
plt.tight_layout()
plt.show()


# plotting the state after evolution
plt.figure()
plt.plot([i-L//2 for i in range(L)], [nop(i,psi) for i in range(L)])
plt.plot([i-L//2 for i in range(L)], [nop(i,psi) for i in range(L)], 'k o')
plt.ylim([0,1])
plt.xlabel('site i')
plt.ylabel('<n>')
plt.show()
```

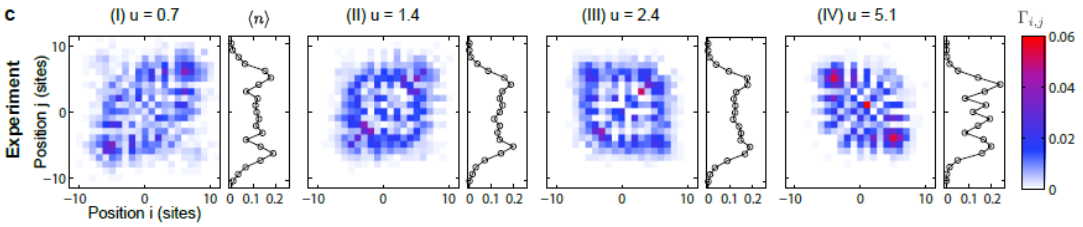


Figure 3.10: Results obtained by Greiner for two particles initialized next to each other after time $\tau_{max} \approx \pi$

Fig. 3.10 shows the experimental results and Fig. 3.11 the simulated results. In Fig. 3.11, we see the fermionization of our bosons as we increase the lattice potential, and thus, their interaction. For $u = 0.7$ we see both bosons tend to be in the same site and close to each other as shown by the high amplitude near the diagonal entries in Fig. 3.11 (a). However as we increase u to $u = 5.1$, we see that particles tend to be in opposite sites about their initial site (i,e), if particle 1 is in site i, the other particle 2 is most likely to be found in site -i. This is seen as increased anti-diagonal amplitudes as shown in Fig. 3.11 (d). Even though the wavefunction does not vary much for 3.11 (a) and (d), it turns out that the existence of one particle in site i

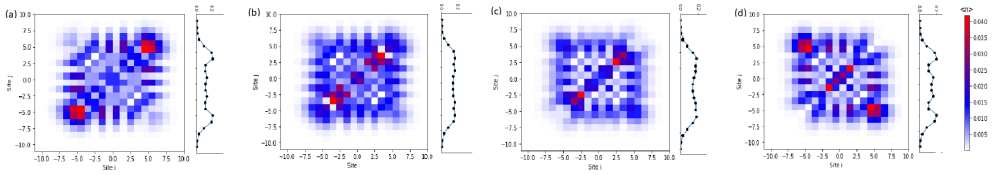


Figure 3.11: Simulation result of HBT interference in two bosons CTQW after $\tau = 3.14$. The correlation matrix and the number operator expectation of $|\psi(t)\rangle$ are shown. (a) shows the result with $u = 0.7$, (b) $u = 1.4$, (c) $u = 2.4$, and (d) $u = 5.1$.

then collapses the other particle closer or farther from it based on the above HBT interference.

We also see the formation of repulsively bound pairs of bosons if we initialize both the particle on the same site.

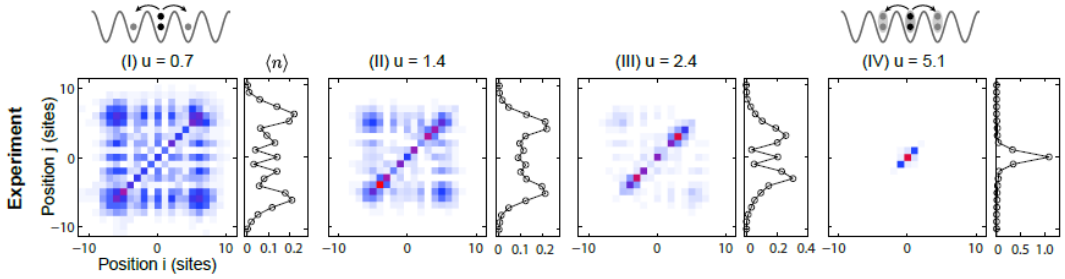


Figure 3.12: Results obtained by Greiner for two particles initialized in the same middle site, after time $\tau_{max} \approx \pi$

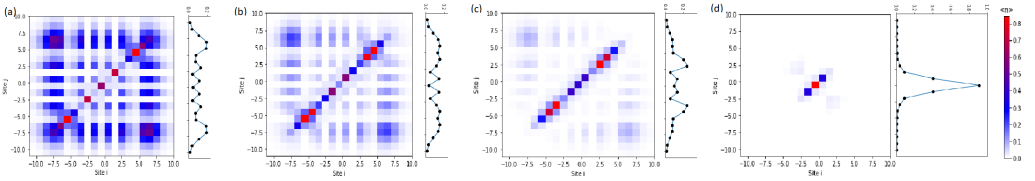


Figure 3.13: Simulation result of repulsively bound bosons in two bosons CTQW after $\tau = 3.24$. The correlation matrix and the number operator expectation of $|\psi(t)\rangle$ are shown. (a) shows the result with $u = 0.7$, (b) $u = 1.4$, (c) $u = 2.4$, and (d) $u = 5.1$.

Fig. 3.12 shows the experimental results and Fig. 3.13 the simulated results. In Fig. 3.13 we see that instead of producing the HBT effect like in

Fig. 3.11, two bosons in the same site produces a repulsively bound pair of boson. In low potential, both these bosons behave as free particles and perform independent uncorrelated quantum walks as seen in Fig. 3.13 (a) since the correlation amplitudes are symmetric about the center site. However, as we increase the potential to $u = 5.1$ as in Fig 3.13 (d), we observe strong localization of our particles, thus expecting them to be repulsively bound bosons. Repulsive because the coexistence of two bosons in the same site is unstable if the potential is large. However, they still exist in the lattice as if they were a single particle facing potential twice that of a single boson.

The formation of a repulsively bound pair is confirmed by looking at their Bloch oscillation dynamics. Fig. 3.14 shows the experimental data of Bloch oscillation while Fig. 3.15 shows the simulated results.

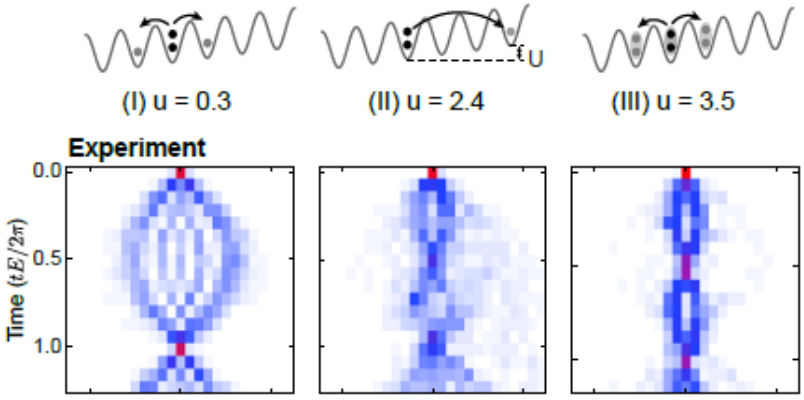


Figure 3.14: Results obtained by Greiner for two particles initialized in the same middle site, performing Bloch oscillations in a lattice with gradient $\Delta = 0.5$

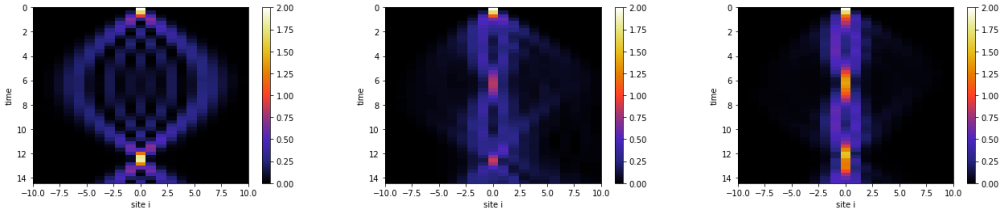


Figure 3.15: Simulation of two bosons performing Bloch oscillations with varying potential u with potential gradient $\Delta = 0.5$. (a) shows Bloch oscillations with $u = 0.3$, (b) $u = 2.4$, (c) $u = 3.5$.

We observe a very interesting outcome here. As we increase the potential to make our bosons repulsively bound (in Fig.3.15 (a) to (c)), the time period of Bloch oscillation halves. This is apparent if we consider the term $i\Delta\hat{n}_i$ term. If two bosons stay in the same site then the operator produces an additional two in the term coming from the number operator for every i, compared to one boson in the site. Effectively this can be interpreted as a coefficient $i(2\Delta)$ w.r.t single particle gradient term, that causes the time period to halve as per Eq. 2.46. Our simulation is also in good agreement with the experimental result shown in fig. 3.14.

Chapter 4

Conclusion and future work

In this internship project, we have familiarised ourselves with the concept of Classical walk in a graph, also described as a Markov process, its discrete, and its continuum limit, quantizing it to obtain continuous-time quantum walks, and also the theory of discrete-time quantum walks done with a quantum coin. A literature survey of the physics of all the relevant phenomena happening in a QW of bosons in an optical lattice such as the HBT effect, repulsively bound bosons, Bloch oscillations, and the theory of Bose Hubbard model was done. A rigorous derivation of all the formulas used was also given. In the simulation, we used *Mathematica* to simulate CTQW on graphs based on Markov processes. Simulations on an optical lattice with bosons were done in python using *quspin* and all experimental observations in *Preiss et.al* were reproduced with acceptable accuracy. Codes written to obtain those results were also provided and explained where necessary.

This internship has acted as a primary exposure to the currently active field of quantum computation and quantum physics. By coding and simulating the physics of particles in an optical lattice and reading the theory of the setup and its experimental implementation as we reproduce their results, we have gained some knowledge on quantum walks that is required before pursuing research in the future. We shall take this internship forward by working on quantum algorithms based on QW.

Acknowledgment

I take this opportunity to express my heartfelt gratitude to Prof. Tapan Mishra for giving me this project and introducing me to the field of quantum walks and assisting me whenever I needed him. A big thanks to his Ph.D. bhaiyaas Mrinal Kant Giri, and Ashirbad Pradhan for assisting me in writing the codes and having the patience to clarify my doubts and debug my codes. Thanks to Prof. Andrew Mitchell from the University of Dublin for providing open access to detailed lectures on condensed matter physics on youtube. Thanks to my parents for believing in me and motivating me to do research, and Dr. Pratap Kumar Sahoo for supporting me during my existential crisis and suggesting me useful tips to proceed in science. Finally a big thanks to the Almighty (Yes, I'm a religious person).

References

[1] Philipp M. Preiss and Ruichao Ma and M. Eric Tai and Alexander Lukin and Matthew Rispoli and Philip Zupancic and Yoav Lahini and Rajibul Islam and Markus Greiner. *Strongly correlated quantum walks in optical lattices*. Science 2015. https://doi.org/10.1126/science.1260364

[2] T. Jeltes, J. M. McNamara, W. Hogervorst, W. Vassen, V. Krachmalnicoff, M. Schellekens, A. Perrin, H. Chang, D. Boiron, A. Aspect C. I. Westbrook. *Comparison of the Hanbury Brown –Twiss effect for bosons and fermions*. Nature 2007. doi:10.1038/nature05513

[3] Abramowitz M and Stegun I A 1972 Handbook of Mathematical Functions (New York: Dover)

[4] T. Hartmann, F. Keck, H. J. Korsch, and S. Mossmann, *Dynamics of Bloch Oscillations*. New J. Phys. 6, 2 (2004)

[5] Portugal, R. (2018). *Introduction to Quantum Walks. Quantum Walks and Search Algorithms*, 19–40. doi:10.1007/978-3-319-97813-0

[6] Markus Greiner, *Ultracold quantum gases in three-dimensional optical lattice potentials*. PhD Thesis in Physics department of the Ludwig-Maximilians-Universität München (2003)

[7] Y. Aharonov, L. Davidovich, and N. Zagury. *Quantum random walks*. Phys. Rev. A 48, 1687 (1993)

[8] N.B. Lovett, S. Cooper, M. Everitt, M. Trevers, and V. Kendon. *Universal quantum computation using the discrete-time quantum walk*. Phys. Rev. A, 81(4):042330 (2010)

[9] Salvador E. Venegas-Andraca. *Quantum walks: A comprehensive review*. Quantum Information Processing vol. 11(5), pp. 1015-1106 (2012)

[10] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic quantum computation is equivalent to standard quantum computation. SIAM Journal of Computing, vol. 37(1), pp. 166–194, 2007.

[11] S.E. Venegas-Andraca and S. Bose. Quantum computation and image processing: New trends in artificial intelligence. Proceedings of the International Conference on Artificial Intelligence IJCAI-03, pp. 1563–1564, 2003

[12] M. Lanzagorta and J. Uhlmann. Quantum algorithmic methods for computational geometry. Mathematical Structures in Computer Science, vol. 20(6), pp. 1117–1125, 2010.

[13] C. Trugenberger. Quantum pattern recognition. Quantum Information Processing, vol.1(6), pp. 471–493, 2002.

[14] G. Abal, R. Donangelo, and H. Fort. Conditional strategies in iterated quantum games. Physica A, vol. 387, pp. 5326–5332, 2008.

[15] M. Lanzagorta. Quantum Radar. Morgan and Claypool, 2011.

[16] R.P. Feynman. Simulating physics with computers. International Journal of Theoretical Physics, vol. 21(6/7), pp. 467–488, 1982.

[17] I. Kassal, S.P. Jordan, P.J. Love, M. Mohseni, and A. Aspuru-Guzik. Quantum algorithms for the simulation of chemical dynamics. Proc. Natl. Acad. Sci., 105:18681, 2008.

[18] A. M. Iliyasu, P.Q. Le, F. Dong, and K. Hirota. A framework for representing and producing movies on quantum computers. International Journal of Quantum Information, vol. 9(6), pp. 1459–1497, 2011.

[19] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete algorithms on a quantum computer. SIAM Journal on Computing, vol. 26(5), pp. 1484–1509, 1997.