A driven one dimensional quantum many-body system

Master Thesis

Eleana Makri

Supervisor: Xenophon Zotos

Physics Department
University of Crete

November 2011

A driven one dimensional quantum many-body system

by

Eleana Makri

Physics Department, University of Crete

2011

Abstract

The theoretical study of time-dependent phenomena in quantum many-body systems is a challenging problem and remains to a large extent unexplored. In this work we examine the time evolution of a half-filled 1-D lattice of hard-core bosons driven by a static field through the numerical study of the time-evolution operator of the system calculated over one Bloch period. The Hamiltonian of the problem is mapped to the spin-1/2 1-D Heisenberg model via the Jordan-Wigner transformation. The evolution operator is calculated as the time-ordered product of operators describing the system evolution in elementary steps. Then the effect of the system parameters on the structure of the spectrum of the evolution operator is examined. We observe a transition of the spectrum from irregular to regular as the intensity of the driving increases. On the other hand, we observe a wider distribution of the eigenvalues as the repulsive interaction between nearest-neighbours increases. The presence of an impurity breaks the integrability, causing a lift of the degeneracies in the spectrum. The particle current expectation value is expressed in terms of the evolution operator eigenvalues and eigenfunctions and is then related to the spectrum of the evolution operator for various system parameters. In most cases the structure of the spectrum is reflected to the behaviour of the particle current. Finally, a roughly linear increase of the particle current expectation value is observed for the integrable system with strong particle interactions at the adiabatic limit.

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Prof. Xenophon Zotos, for his valuable guidance and assistance throughout this work, as well as for all that he has taught me all these years. Furthermore, I would like to thank Prof. I. E. Perakis and Prof. G. P. Tsironis for joining the thesis committee. I am grateful to my family and especially to my mother for continuous support throughout my studies. I am also indebted to the "2nd floor" graduate students of the Physics Department, who have been helpful beyond description in any case I asked for help or advice. Finally, I would like to thank my close friends for being a constant source of encouragement and for contributing to the completion of the present thesis in many different levels.

Contents

Chapter 1

Introduction

One problem of interest in condensed matter physics is the motion of a quantum particle in a periodic potential in the presence of a static force acting on the particle. This model corresponds to the motion of an electron in a crystalline solid when an electric field is applied. According to quantum mechanics, the motion of the particle will be oscillatory and the oscillation period will be $T_B = h/(Fd)$, F being the force and d being the lattice constant [1]. In reality, these oscillations which are named after Felix Bloch who predicted them [2] decay before even one period is completed, due to scattering with lattice impurities and phonons [1]. Moreover, since the lattice constant is of the order of a few $\mathring{A}$, an enormous field intensity would be required to have small enough oscillation periods compared to the characteristic scattering time of the electrons. Consequently, Bloch oscillations have not been observed in crystalline solids.

Bloch oscillations are just an example in a range of theoretically predicted ideal lattice phenomena, such as Wannier-Stark ladders, which have not been observed in solids due to scattering [1]. The realization of novel experimental systems such as ultracold atoms in optical lattices and semiconductor superlattices during the last two decades have enabled the experimental observation of such phenomena [1],[3],[4],[5]. In more detail, a driven cold atom moving in an optical lattice is the perfect analogue of an electron in a crystal at the presence of an electric field. The driving force can be experimentally implemented by many methods apart from an electric field, such as gravity [6] or acceleration of the optical lattice [7]. The first experiments were carried out using dilute atomic samples, corresponding to negligible interactions among the particles [4], however experiments using Bose-Einstein condensates with stronger interparticle interactions have also been conducted [7].

The experimental progress induced a renewed theoretical interest in the field of driven quantum many-body systems [8]. Moreover, the development of extremely controllable experimental systems provided the opportunity for realization and experimental investigation of integrable quantum many-body models used in the field of correlated systems. This enabled experimental study of theoretical concepts related to the integrability of quantum systems, such as the difference of integrable and non-integrable systems in their linear-response to an external field [9],[10]. However, in spite of the advancements in experimental techniques, the theoretical description of the real-time response of quantum systems remains a largely unexplored field. The study of time-dependent phenomena in such systems is challenging and is usually carried out using numerical approaches [11].

The present study is an attempt towards the description of the time-evolution of such a system. More precisely, we examine the response of a system of hard-core bosons to driving by a static field. The hard-core boson model is mapped to the spin-1/2 1-D Heisenberg model. The time evolution of the system is formulated by numerically calculating the time evolution operator of the system over one Bloch period. This method is not very efficient in terms of computational time, however, the information contained in the evolution operator is expected to provide some insights concerning the dynamics of the system. Particularly, we will study the expectation value of the particle current for large times after expressing the dynamics of the system in terms of the eigenfunctions and eigenvalues of the evolution operator. In addition, the effect of variable system parameters on the spectrum of the evolution operator will be examined. Finally, we will analyse the particle current for a few choices of system parameters, using the breakdown of its analytical expression in terms of the evolution operator eigenfunctions and eigenvalues and we will relate the current behaviour to the evolution operator spectrum.

Chapter 2

Driven hard-core bosons

2.1 Model

We study a half-filled 1-D lattice of hard-core bosons with periodic boundary conditions. The system is driven by a static electric field F applied via a time-dependent magnetic flux $\phi(t) = -qFt$. Assuming that the field affects only the kinetic energy of the system and taking into account the repulsive particle interactions between nearest neighbours and the presence of an impurity at a lattice site, the Hamiltonian describing the system will be

$$\hat{H} = -t_h \sum_{l} (e^{i\phi(t)} b_{l+1}^{\dagger} b_l + h.c.) + \Delta \sum_{l} \hat{n}_l \hat{n}_{l+1} + w\hat{n}_1 \tag{2.1}$$

where $b^{\dagger}, b$ are the bosonic creation and annihilation operators respectively, $\hat{n}$ is the particle number, t_h the nearest site hopping term, Δ the nearest neighbour repulsive potential and w an impurity at a lattice site. As mentioned, the particles are hard-core bosons, so no more than one per lattice site is permitted. This model can be mapped to the spin-1/2 1-D Heisenberg model via the Jordan-Wigner transformation [12].

It should be mentioned that the location of the impurity on the lattice is unimportant due to the fact that periodic boundary conditions are used. We have taken $\hbar = q = d = 1$, q being the particle charge and d being the lattice constant.

2.2 Time evolution

The time-dependent Schrödinger equation of the system is

$$i\frac{d|\Psi(t)\rangle}{dt} = \hat{H}(-Ft)|\Psi(t)\rangle \tag{2.2}$$

By making the substitution $\phi = -Ft$, (2.2) becomes

$$i\frac{d|\tilde{\Psi}(\phi)\rangle}{d\phi} = -\frac{1}{F}\hat{H}(\phi)|\tilde{\Psi}(\phi)\rangle \tag{2.3}$$

We can now set $H(\phi) = -\frac{1}{F}\hat{H}(\phi)$ and rename $|\tilde{\Psi}\rangle$ for $|\Psi\rangle$ and proceed to our calculations using phase instead of time. We can formulate the phase evolution of the system by defining an operator U such that

$$U(\phi)|\Psi(0)\rangle = |\Psi(\phi)\rangle \tag{2.4}$$

This operator is essentially the equivalent of the time evolution operator of the problem in phase terms. Because of the translational symmetry of the system, knowledge of the evolution operator at 2π can allow as to describe the system dynamics at all phases $\phi = m \cdot 2\pi$, m being an integer. In particular, the evolution of a state of the system at 2π circles will be $|\Psi(\phi+2\pi)\rangle = U(2\pi)|\Psi(\phi)\rangle$. Hence if we calculate the operator at $\phi = 2\pi$, we can benefit from this property to easily calculate the evolution at all phases $\phi = m \cdot 2\pi$.

The evolution operator at $\phi = 2\pi$ can be expressed as a time-ordered product of operators U_n

$$U(2\pi) = \prod_n U_n = Te^{-i\int_0^{2\pi} H(\phi)d\phi} \tag{2.5}$$

which is equivalent to

$$U(2\pi) = U_N U_{N-1} ... U_2 U_1 = e^{-i\int_{\phi_{N-1}}^{\phi_N} H(\phi)d\phi} ... e^{-i\int_{\phi_1}^{\phi_2} H(\phi)d\phi} e^{-i\int_{\phi_0}^{\phi_1} H(\phi)d\phi} \tag{2.6}$$

$\phi_N = 2\pi$, $\phi_0 = 0$. This allows us to discretize the phase, which is necessary if we want to numerically calculate the evolution operator. Therefore, if we discretize the phase to N intervals of width $\Delta\phi$, we can calculate U_n by taking an approximation at the middle of each phase interval. Thus the evolution operator can be broken down to

$$U(2\pi) = e^{-iH((2N-1)\frac{\Delta\phi}{2})\Delta\phi} ... e^{-iH(3\frac{\Delta\phi}{2})\Delta\phi} e^{-iH(\frac{\Delta\phi}{2})\Delta\phi} \tag{2.7}$$

So naming $U_n = e^{-iH((2n-1)\frac{\Delta\phi}{2})\Delta\phi}$, each term of the time-ordered product can be approximated as follows

$$U_n = e^{-iH((2n-1)\frac{\Delta\phi}{2})\Delta\phi} = \frac{e^{-\frac{i}{2}H((2n-1)\frac{\Delta\phi}{2})\Delta\phi}}{e^{\frac{i}{2}H((2n-1)\frac{\Delta\phi}{2})\Delta\phi}} \approx \frac{1 - \frac{i}{2}H((2n-1)\frac{\Delta\phi}{2})\Delta\phi}{1 + \frac{i}{2}H((2n-1)\frac{\Delta\phi}{2})\Delta\phi} \tag{2.8}$$

which is the implementation of the Crank-Nicolson method on our problem [13]. At the beginning of our study we used this method, however, we discovered that it did not provide the accuracy we needed for reasons that will be explained in the following chapter. Eventually, the exponential form of the operator was used as it is as the term of the time-ordered product, i.e. $e^{-iH(\phi)\Delta\phi}$.

The calculation of the evolution operator at integer multiples of 2π can be further simplified if we diagonalize the evolution operator at $\phi = 2\pi$. The evolution operator is unitary, therefore its diagonal form will be

$$U = \sum_k |k\rangle e^{i\theta_k} \langle k| \tag{2.9}$$

Consequently, the evolution operator at $\phi = m \cdot 2\pi$ will simply be

$$U^m = \sum_k |k\rangle e^{im\theta_k} \langle k| \tag{2.10}$$

Thus the states of the system at $\phi = m \cdot 2\pi$ can be calculated by the action of U^m on the initial state $|\Psi(0)\rangle$

$$|\Psi_m\rangle = U^m |\Psi(0)\rangle \tag{2.11}$$

It should also be mentioned that the spectrum of the evolution operator is in the focus of the present work and will be analysed for various system parameters.

Another physical quantity of interest in our problem is the particle current.

The quantum mechanical operator describing it is

$$j(\phi) = -\frac{dH}{d\phi} = t_h \sum_l (ie^{i\phi} b^\dagger_{l+1} b_l + h.c.) \tag{2.12}$$

and using it we can calculate the current expectation value using the evolved states $|\Psi(\phi)\rangle$

$$\langle j(\phi)\rangle = \langle \Psi(\phi)|j(\phi)|\Psi(\phi)\rangle \tag{2.13}$$

This can be also written in terms of the evolution operator U at $\phi = m \cdot 2\pi$ as follows

$$\begin{aligned}\langle j_m\rangle &= \langle \Psi(\phi = 0)|U^{m\dagger} j U^m|\Psi(\phi = 0)\rangle \\ &= \langle \Psi(0)| \sum_k (|k\rangle e^{-im\theta_k} \langle k|) j \sum_{k'} (|k'\ \rangle e^{im\theta_{k'}} \langle k'|)|\Psi(0)\rangle \\ &= \sum_{k,k'} \langle \Psi(0)|k\rangle\langle k|j|k'\ \rangle\langle k'|\Psi(0)\rangle e^{im(\theta_{k'}-\theta_k)} \end{aligned} \tag{2.14}$$

The study of the particle current may provide some insights concerning the response of the system to driving, especially through the study of the terms in (2.14). It should be noticed that for $\phi = m \cdot 2\pi$, the operator j is independent of m, thus the only m dependence of the particle current expectation value emerges from the phases in the exponential that appears in (2.14), which come from the evolved states $|\Psi_m\rangle$.

Chapter 3

Numerical method

The numerical study was realized in a computer program written in Fortran 77 programming language. Double precision variables and arrays were used, as well as subroutines for numerical diagonalization contained in the EISPACK package. We will now proceed into explaining in more detail the steps we took to numerically study the system. A version of the source code is provided in Appendix A.

3.1 Construction of the Hamiltonian

The first step is the construction of the space of localized states. The lattice of our study consists of l sites and n particles are contained, no more than one at a site since the particles are assumed to be hard-core bosons. Therefore, the dimension of the Hilbert space will be $L = \frac{l!}{n!(l-n)!}$. In our problem we assume half-filling, so $n = \frac{l}{2}$.

One way to implement the localized states of an 1-D chain of hard-core bosons (or fermions) for a computer simulation is to represent them as a sequence of "0"'s an "1"'s depending on whether the respective lattice site is empty or contains a particle. For instance, if we have a lattice with $l = 6$ sites and the $n = 3$ particles are at the first, fifth and sixth lattice site, this state can be represented as $|100011\rangle$. This representation, which is much resembling the representation of a number in the binary system, implies a very memory-efficient way to store the states in the computer memory. Each state can be considered a binary number upon its creation, easy to be temporarily stored in a $1 \times l$ b[0:l-1] array and can then be converted to the decimal system making it easy to be stored as one single array element. In our previous example, the state $|100011\rangle$ will be converted to $|35\rangle$. In the mapping to the spin-1/2 Heisenberg model, the spin-up ($s = +1/2$) states are mapped to the 1's of the aforementioned representation (corresponding to the presence of a particle at a lattice site) and the spin-down ($s = -1/2$) states are mapped to the 0's (corresponding to an empty lattice site). This is simply done by using a 1×2 array sz indexed [0:1] with matrix elements sz[0]=-1/2 and sz[1]=+1/2. The spin of each lattice site b[i] will then be sz[b[i]]. In order to implement half-filling, the total spin of each state must be equal to 0. Therefore, in order to create the states, we use a counter that runs from 0 to 2^{L-1}. Each value of the counter is converted to a binary number, and then the total spin of the state is calculated. If the total spin is 0, the binary number describing the state is converted to its decimal representation and it is then stored at an $1 \times L$ array. Eventually we have all L states constructing the Hilbert space of our problem.

After the localized states are created, we move on to the construction of the Hamiltonian of the system. The repulsive particle interactions between nearest-neighbours correspond to the diagonal elements of the Hamiltonian. In order to compute them, each localized state is recalled, the decimal number representing it is converted back to a binary number, and we multiply the spin values of adjoining lattice sites. We will have a positive spin product for parallel spin values and negative for anti-parallel spin values. For each state, the products of this multiplication are summed and each sum is stored as an array element at a $1 \times L$ array. As mentioned before, periodic boundary conditions are used, so the interactions between the first and last lattice site are also included in the calculation. In addition to the particle interactions, the diagonal elements of the Hamiltonian must also contain the lattice impurity. This is simply done by saving the spin of the first lattice site corresponding to the impurity at another $1 \times L$ array for each state. Eventually, the diagonal elements of the Hamiltonian are calculated by recalling the respective array elements corresponding to the nearest-neighbour interaction and multiplying their values with Δ and then adding the array element corresponding to the impurity multiplied with w.

For the non-diagonal elements, the hopping is mapped to the spin flip of the Heisenberg model. As before, each one of the stored states is recalled and converted to its binary representation. Then, a do loop runs on all lattice sites of the binary number, which is stored in an array named ib[0:l-1]. After saving the initial state in an auxiliary array, 1 is added to value of each array element and 1 is subtracted from the following array element, i.e. if i is the counter, ib[i] becomes ib[i]+1 and ib[i+1] becomes ib[i+1]-1. There are conditional statements that ensure that this change is accepted only if the ib matrix elements are still valued either 1 or 0. In other words, if the value of a lattice site is 0, and the value of the following site is 1, then the first will become 1 and the second 0. Again, periodic boundary conditions are used, therefore the hopping amplitude between the last and first lattice site is also taken into account. For each original state, the number of states that can result from hopping is counted and this count is stored as an array element in an $1 \times L$ array named nspsm. Eventually, after all possible hops are done, each element of nspsm will have the total count of number of states that can result from the original state. The new binary number resulting after each hopping is converted to its decimal representation and is then mapped to the index of the corresponding state. Then, this index is stored in an integer valued $L \times 2l$ array named ispsm. For instance, the if there is hopping amplitude between states i and j, we'll have ispsm[i,nspsm[i]]=j. In addition, the product of the eigenvalues of the spin raising and lowering operators between the two lattice sites where the hopping occurred is stored in a double precision valued $L \times 2l$ array named espsm, again at the element [i,nspsm[i]]. Finally, the non-diagonal elements of the Hamiltonian can be calculated as the product of the hopping amplitude, the respective espsm array elements and the phase term $e^{i\phi} = \cos\phi + i\sin\phi$.

The Hamiltonian is stored in a single real $L \times L$ array. This array actually contains the lower triangle of the Hamiltonian, with the real parts of the matrix elements stored in the lower triangle of the matrix (including the diagonal) and the imaginary parts stored in the transposed positions of the upper triangle of the Hamiltonian. Then it is diagonalized using the appropriate sequence of subroutines for the diagonalization of hermitian matrices from the EISPACK package, which are *htrid3* (transforming the original matrix to its tridiagonal form), *tql2* (diagonalizing the tridiagonal matrix) and *htrib3* (forming the eigenvectors of the hermitian matrix from those of the real symmetric tridiagonal matrix determined by *htrid3*). Since we now know the energy eigenvectors and eigenvalues, we can proceed to the time evolution of the system.

3.2 Evolution

The evolution of the system is calculated using phase instead of time as in (2.3). The phase interval $[0, 2\pi]$ is discretized to n_t equal intervals, so $\Delta\phi = \frac{2\pi}{n_t}$.

For every interval the Hamiltonian is diagonalized in order to use the energy eigenvalues and eigenstates for the calculation of the evolution operator. The static field only enters the problem as a divisor of the energy eigenvalues as seen in (2.3). To numerically implement (2.7), we first used the approximation (2.8) to calculate U_n at the middle of each phase interval $\Delta\phi$. In more detail, U_n was broken down to its real and imaginary part and each one was calculated as in (2.8). In particular,

$$\begin{aligned} U_n = &\sum_{n_1} |n_1\rangle \frac{1 - 1/4(E_{n_1}/F)^2\Delta\phi^2}{1 + 1/4(E_{n_1}/F)^2\Delta\phi^2} \langle n_1| \\ &+ i \sum_{n_1} |n_1\rangle \frac{-(E_{n_1}/F)\Delta\phi}{1 + 1/4(E_{n_1}/F)^2\Delta\phi^2} \langle n_1| \end{aligned} \tag{3.1}$$

where $|n_1\rangle, E_{n_1}$ are the eigenstates and the eigenvalues of the Hamiltonian. For $F < 1$, the fraction $\frac{\Delta\phi}{F}$ becomes large and the approximation does not provide the accuracy needed. In particular, we observed large variations of the particle current as we varied the value of n_t. Consequently, the Crank-Nicolson method was rejected and the exponential form of the operator was used instead, as it provided adequate accuracy.

$$\begin{aligned} U_n &= e^{-iH(\phi)\Delta\phi} \\ &= \sum_{n_1} |n_1\rangle \cos((E_{n_1}/F)\Delta\phi) \langle n_1| - i \sum_{n_1} |n_1\rangle \sin((E_{n_1}/F)\Delta\phi) \langle n_1| \end{aligned} \tag{3.2}$$

The real and imaginary part of the evolution operator are each stored in a 2-D array. Then, as phase evolves, we take the matrix product $\tilde{U}_n = U_n \tilde{U}_{n-1}$, with $\tilde{U}_0$. Therefore, after n_t steps $U(2\pi) = \tilde{U}_n$. In order to test the validity of the calculation, we checked and confirmed that $U^\dagger U = I$, as well as that $U(4\pi) = U(2\pi) \cdot U(2\pi)$.

We then proceed to numerically diagonalize the evolution operator. The matrix representing the evolution operator is complex and non-Hermitian, therefore it was diagonalized using *cg* subroutine from the EISPACK package, which is suitable for diagonalizing a general complex matrix. A numerical problem we encountered in the integrable case where there is no impurity was that *cg* returned non-normalized eigenvectors. Therefore, we set $w = 10^{-6}$ in all cases of a ”perfect lattice” in chapter 4. This negligible impurity broke the degeneracy and normalized eigenvectors were returned as outputs from *cg*, without causing any considerable accuracy issues.

Since the evolution operator is unitary, the complex eigenvalues $w_k = w_r + iw_i$ have all norm $|w_k| = 1$, thus they can be represented using only the argument of the polar form. This can be calculated as $\theta_k = \arctan(\frac{w_i}{w_r})$. The range of (principal) values of the inverse tangent function is $[-\pi/2, \pi/2]$, therefore if $w_r < 0$ we have to modify the results to get to the $[-\pi, \pi]$ range. We do this using two conditional statements, mapping $\theta_k \rightarrow \theta_k + \pi$ if $w_i > 0$ and $\theta_k \rightarrow \theta_k - \pi$ if $w_i < 0$. We now have the evolution operator at its diagonal form (2.9).

Then, we calculate the evolved states at $m \cdot 2\pi$. We constructed a do loop that uses m as a counter. For each m, the evolution operator is (2.10) and the evolved state will be (2.11), so we calculate the evolved wave function by acting with U^m on the initial states, breaking it down to 2 one-dimensional arrays of size L to store the real and imaginary part of the wave function. Then we use the result to calculate the current at $m \cdot 2\pi$. The current was also calculated using the action of the time-ordered product of U_n on the initial state for small to large ϕ's to confirm that the methods coincide.

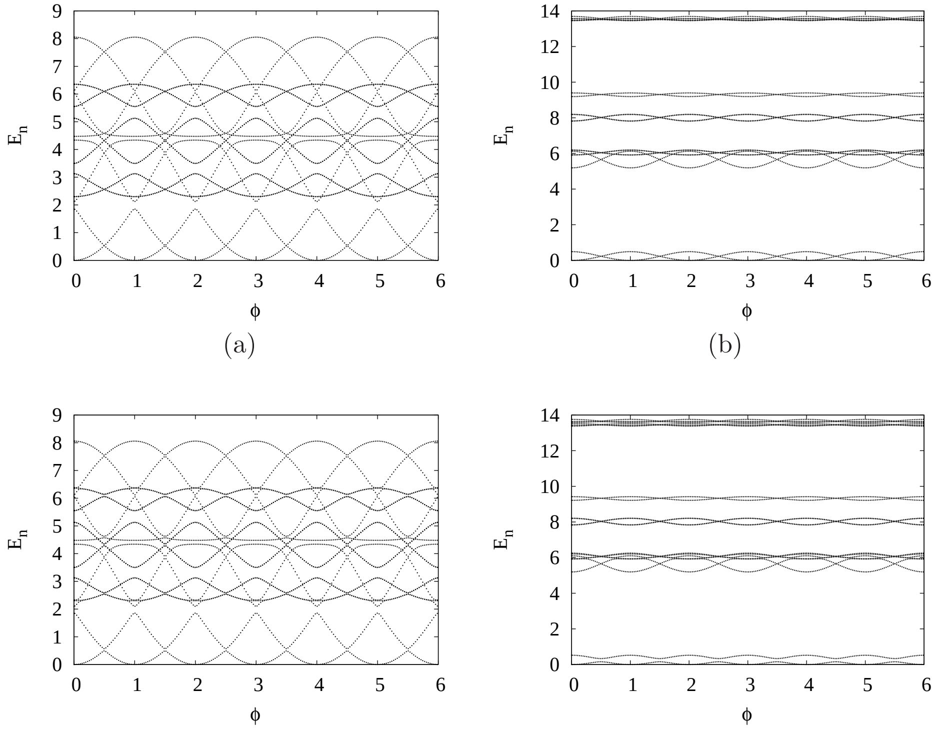
Finally, we calculate the terms of (2.14). More precisely, we calculate the current matrix elements on the evolution operator eigenvectors basis $\langle k|j|k'\rangle$ and the overlap of the initial state $|\Psi(0)\rangle$ with the evolution operator eigenstates $|k\rangle$.

Chapter 4

Results

As seen in (2.1), the parameters characterizing the system of this study are the particle interaction Δ, the impurity w and the intensity of the field F. The particle interaction and the impurity affect the spectrum of the Hamiltonian, while the driving force enters the problem as time evolves. In more detail, as shown in Fig. 4.1, the spectrum of the Hamiltonian of our system for a small Δ is gapless, whereas the spectrum for a large Δ is gapped. The transition of the system from the "metal" (gapless) to the "insulator" (gapped) phase occurs at $\Delta/t_h = 2$. In Figs. 4.1c and 4.1d, the lift of the degeneracies caused by the breaking of the integrability at the presence of w is also evident. On the other hand, the field intensity is related to the response of the system. A small F value corresponds to the adiabatic limit, meaning that the driving force is applied to the system slowly enough for it to adapt to the new conditions, while a large F value corresponds to the sudden limit, where the change of the conditions is too rapid, preventing the state of the system from adapting.

In this chapter, we will examine the effect of the system parameters on the evolution operator spectrum and how the spectrum is related to the expectation value of the particle current. We will also analyse the particle current behaviour using (2.14). Firstly, we obtained the spectrum of the evolution operator eigenvalues at the $[-\pi, \pi]$ range as a function of each of the system variables F, Δ, for different values of the other one, at the presence or absence of an impurity. In addition, the expectation value of the current $\langle j_m \rangle$ was calculated at phases $\phi = 2\pi \cdot m$ as described in the previous chapter using the evolved states (2.11). We obtained results for values of Δ and F from small to large in order to examine the response of the system from the gapless to the gapped case and from the adiabatic to the sudden limit, both for a perfect lattice ($w = 0.0$) as well as at the presence of an impurity ($w = 0.2$). In all cases we study a half-filled lattice of 6 sites. The hopping amplitude is $t_h = 1$.



(a) (b) (c) (d)

Figure 4.1: *Energy spectrum as a function of phase divided by $2\pi/l$ for* (a) *the metal phase ($\Delta = 0.6$) of the integrable case ($w = 0.0$),* (b) *the insulator phase ($\Delta = 6.0$) of the integrable case ($w = 0.0$),* (c)*the metal phase ($\Delta = 0.6$) of the non-integrable case ($w = 0.2$) and* (d) *the insulator phase ($\Delta = 6.0$) of the non-integrable case ($w = 0.2$).*

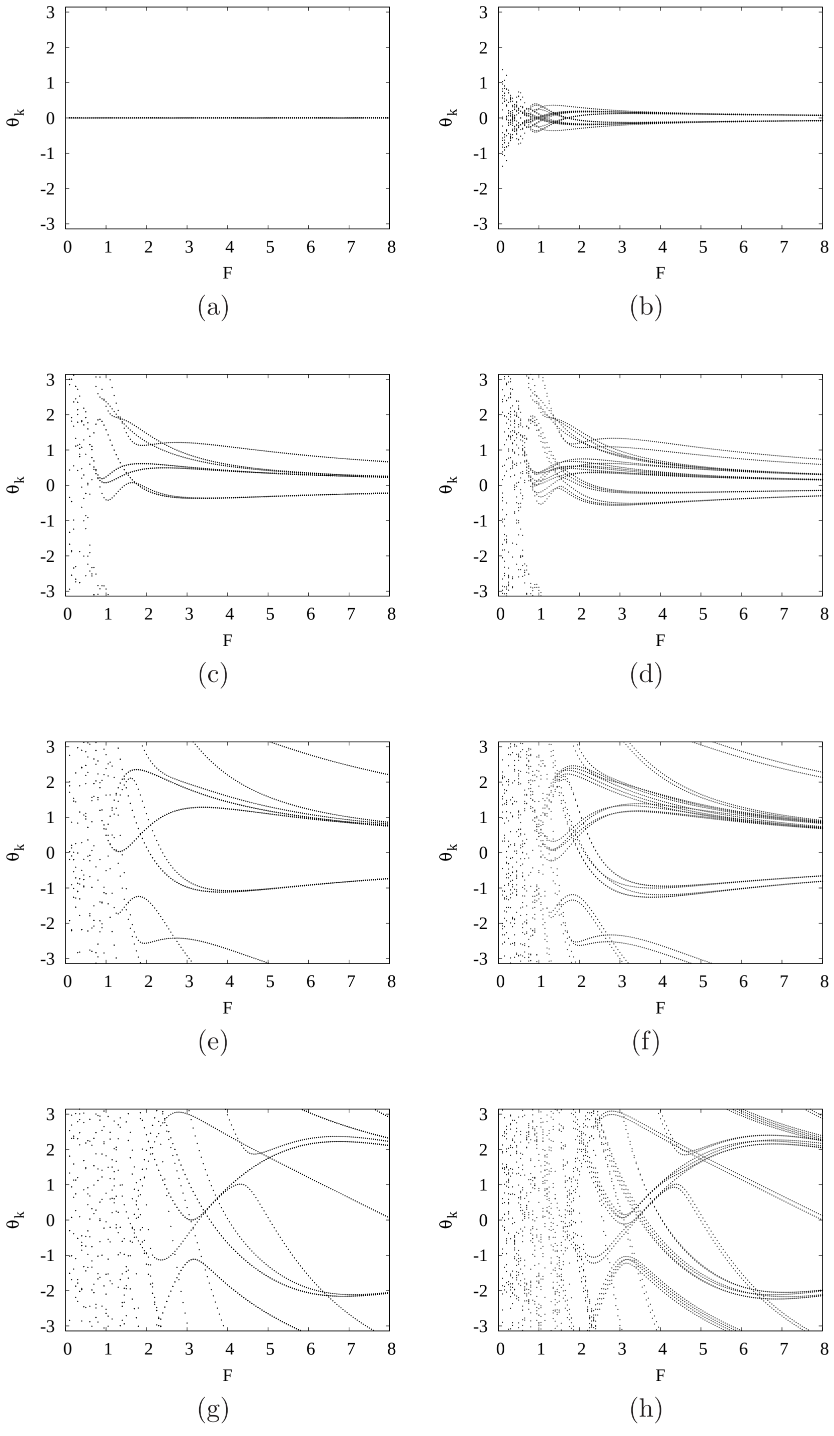
The phase interval from 0 to 2π was discretized to $n_t = 200$ intervals as we implemented the time evolution method described in the previous chapter.

4.1 The effect of the field intensity on the system dynamics

In this section we study the dependence of the spectrum of the evolution operator on the field intensity. We numerically calculated the eigenvalues varying the value of F from 0.10 to 8.00 at 0.05 increments. We present the results for θ_k values as a function of F, for different values of the repulsive interaction Δ, at the absence or presence of an impurity $w = 0.2$.

In the case of non-interacting particles ($\Delta = 0.0$), the many-body problem is reduced to that of a single particle. For a perfect lattice, the system can be easily solved analytically and the current in this case is $\langle j(\phi) \rangle = 2t_h F \sin(\phi)$. Therefore, the current at $\phi = 2\pi \cdot m$ is expected to be $\langle j_m \rangle = 0$ regardless of the field intensity. Our numerical simulation confirmed this result. The eigenvalues of $U(2\pi)$ in this case are all zero, which implies that the states are perfectly periodic with period 2π.

In all other non-trivial cases, it is apparent that as the field intensity increases from the adiabatic limit to the sudden, there is a clear transition of the structure of the spectrum from irregular to regular. The field intensity at which this transition occurs becomes larger as Δ increases. In addition, for even larger values of F, there appears to be a point where the spectral lines begin to converge. Comparing the spectra for different values of Δ, it is also observed that the distribution of the eigenvalues becomes wider at the sudden limit as the magnitude of the repulsive interactions increases. Finally, the presence of the impurity causes a lift of the degeneracy of θ_k eigenvalues leading to a split of the spectral lines.



(a) (b)

(c) (d)

(e) (f)

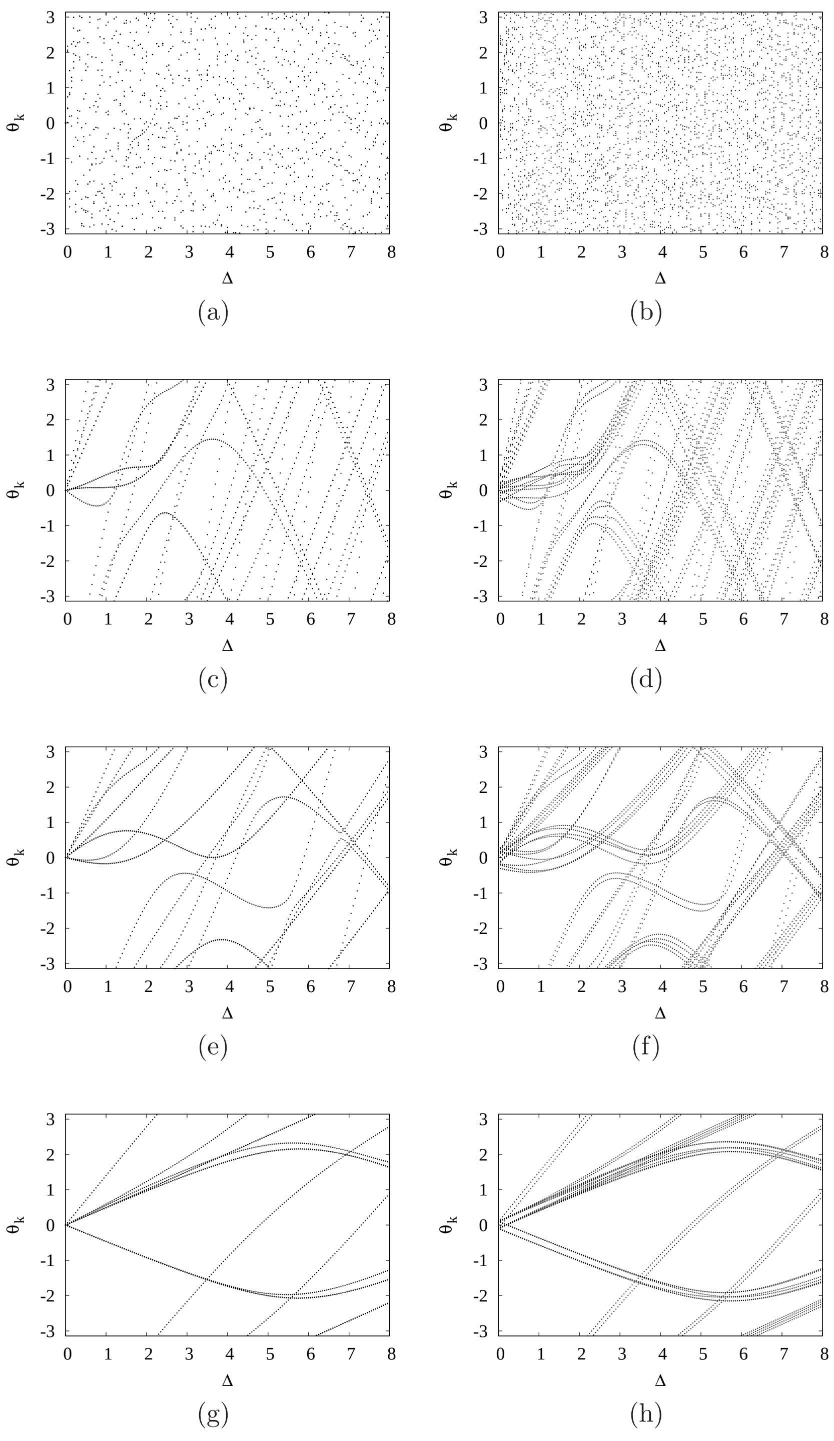
(g) (h)

Figure 4.2: *Evolution operator spectrum as a function of* F *for* (a) $\Delta = 0.0$, $w = 0.0$, (b) $\Delta = 0.0$, $w = 0.2$, (c) $\Delta = 0.6$, $w = 0.0$, (d) $\Delta = 0.6$, $w = 0.2$, (e) $\Delta = 2.0$, $w = 0.0$, (f) $\Delta = 2.0$, $w = 0.2$, (g) $\Delta = 6.0$, $w = 0.0$, (h) $\Delta = 6.0$, $w = 0.2$.

4.2 The effect of the particle interaction on the system dynamics

In this section we present results concerning the effect of the value of Δ on the distribution of the evolution operator eigenvalues. Again, we obtained the spectrum varying the value of Δ from 0.00 to 8.00 at 0.05 increments. In the following figures we demonstrate the results for θ_k values as a function of Δ, for variable values of the field intensity F, at the absence or presence of an impurity.

For small values of F, almost no lines appear in the spectrum, especially at the non-integrable case. The spectrum appears to be chaotic and the eigenvalues are distributed all over the $[-\pi, \pi]$ range regardless of the value of Δ. As the driving becomes stronger, the spectrum becomes canonical and clear lines are observed. The lines start from 0 at the integrable case (and near 0 at the non-integrable) and then diverge. The rate of divergence becomes slower as the field intensity increases and fewer spectral lines are observed, especially for small Δ values, implying more degeneracies. Again, the breaking of the degeneracy is evident for $w = 0.2$.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

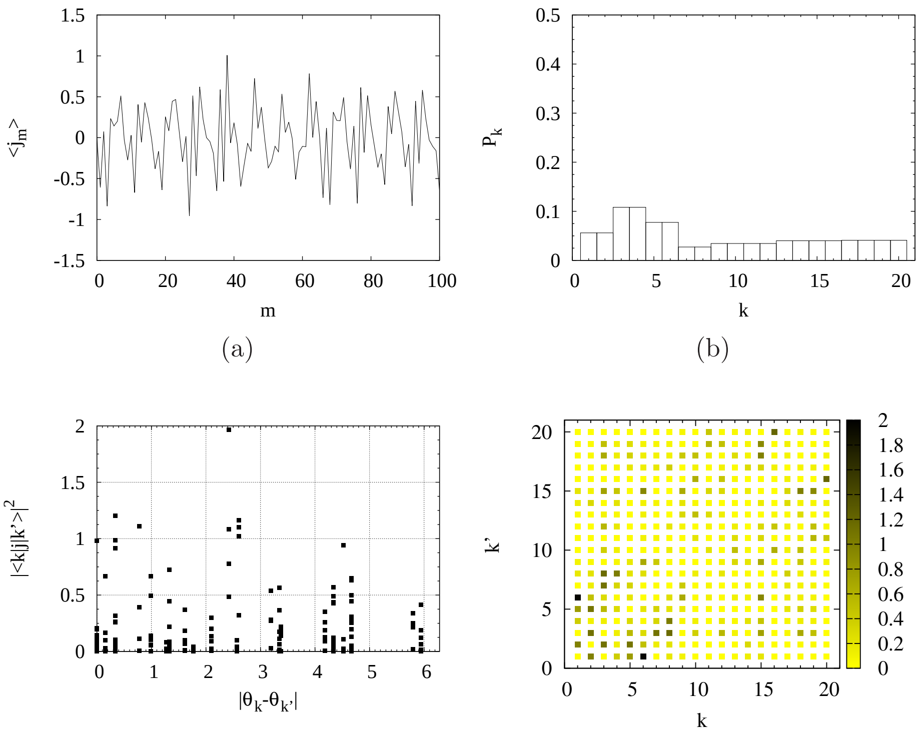
(h)

Figure 4.3: *Evolution operator spectrum as a function of* Δ *for* (a) $F = 0.1$, $w = 0.0$, (b) $F = 0.1$, $w = 0.2$, (c) $F = 1.0$, $w = 0.0$, (d) $F = 1.0$, $w = 0.2$, (e) $F = 2.0$, $w = 0.0$, (f) $F = 2.0$, $w = 0.2$, (g) $F = 6.0$, $w = 0.0$, (h) $F = 6.0$, $w = 0.2$.

4.3 Current analysis

As seen in (2.14), the particle current can be expressed in terms of the evolution operator eigenfunctions and eigenvalues. Each term of the sum in (2.14) is characterized by a frequency given by $|\theta_{k'} - \theta_k|$. These frequencies are related to the periods observed in the particle current. More precisely, the contribution of each term to the result is related to the overlap of the initial state with each of the evolution operator eigenstates and to the current matrix elements in the basis of the evolution operator eigenstates. Therefore, the larger the amplitude of the initial state overlaps and the matrix elements, the more important the contribution of the respective frequency in the result.

We will perform a study of the particle current for a few choices of system parameters and relate it to the spectrum of the evolution operator. Apart from the current, which was calculated up to $m_{max} = 100$, we obtained results for the squared modulus of the overlaps of the initial state with the $|k\rangle$ states $P_k = |\langle k||\Psi(0)\rangle|^2$. We also present results for the squared modulus of the current matrix elements in the evolution operator eigenstate basis $|\langle k|j|k'\rangle|^2$, plotted with the frequencies $|\theta_{k'} - \theta_k|$ as well as with k, k'.



(a) (b) (c) (d)

Figure 4.4: (a) *Current* $\langle j_m \rangle$ *as a function of* m, (b) *Squared modulus of the initial state overlaps with the evolution operator eigenstates* ($|\langle k|\Psi(0)\rangle|^2$), (c) *Squared modulus of the current matrix elements* ($|\langle k|j|k'\rangle|^2$) *plotted with respective frequencies* $|\theta_{k'} - \theta_k|$, (d) $|\langle k|j|k'\rangle|^2$ *plotted as color with the indices of the respective matrix elements* k, k', *for* $w = 0.0$, $\Delta = 0.6$ *and* $F = 0.1$.

The fist case we will examine is that of a system with a small interparticle interaction at the adiabatic limit. As seen in Fig. 4.4c, the distribution of the frequencies $|\theta_k - \theta_{k'}|$ is wide all over the $[0, 2\pi]$ range. This was expected by the spectra shown in Figs. 4.2c and 4.3a, since the spectrum for small F values is irregular. In Fig. 4.4b, we notice that the overlaps of the initial state with the eigenstates of the evolution operator are rather uniform, therefore we expect the contribution of each frequency to the current to be evoked mostly by the current matrix elements. We observe in Fig. 4.4c one matrix element with large modulus, corresponding to a frequency $f \approx 2.4$. The period related to this frequency will be $T = 2\pi/f \approx 3$ and it is observable in Fig. 4.4a, although the behaviour of the current is complex as there is significant contribution of more frequencies.

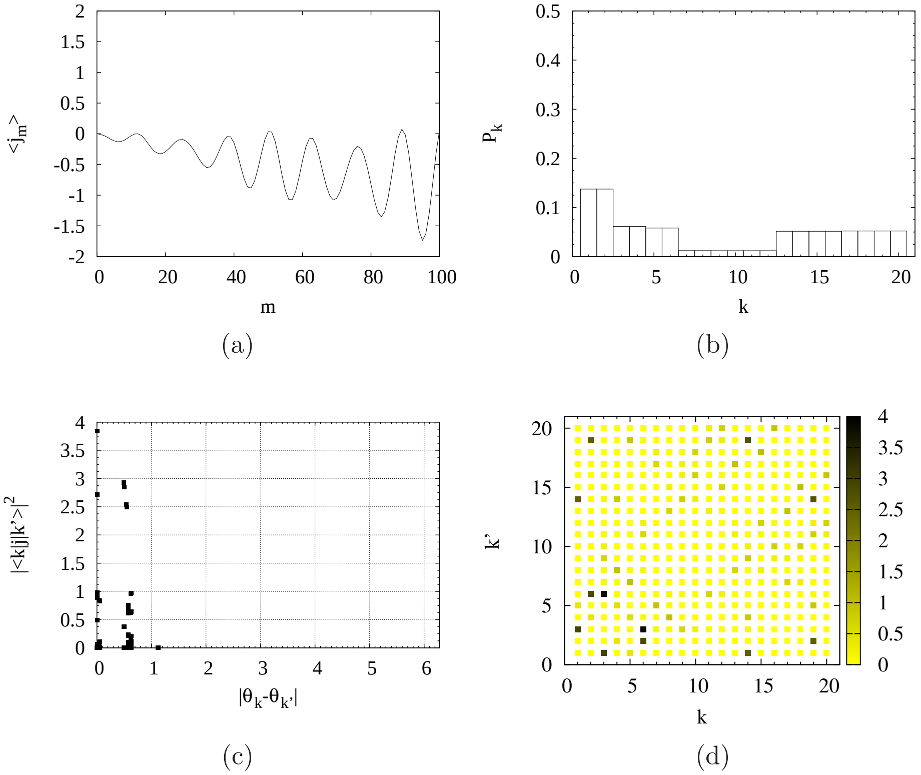
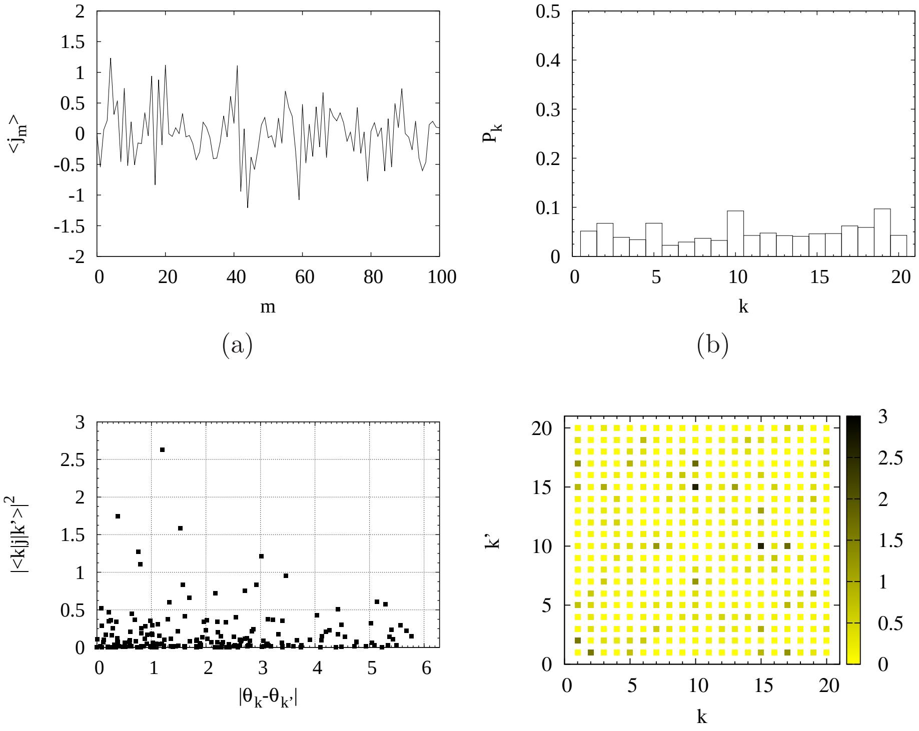


Figure 4.5: (a), (b), (c), (d) *as before, for* $w = 0.0$*,* $\Delta = 0.6$ *and* $F = 6.0$*.*

For sudden driving, the current behaviour is more canonical. Again, observing Figs. 4.2c and 4.3g, fewer frequencies are expected to appear and since the eigenvalues converge as the field becomes larger, these frequencies are expected to be small. Indeed, in Fig. 4.5c it is shown that the distribution of the frequencies is very narrow compared to that of the adiabatic case. The frequencies are small, leading us to expect $\langle j_m \rangle$ to exhibit a more canonical behaviour with large periods. We notice one matrix element with large modulus at a frequency very close to 0, therefore it does not correspond to an observable period. In addition, a significant contribution by two matrix elements is evident near a frequency $f \approx 0.5$. Therefore, we expect a $\langle j_m \rangle$ period $T = 2\pi/f \approx 13$, which is indeed observed in Fig. 4.5a. The overlaps P_k are less uniform than at the adiabatic case as seen in Fig. 4.5b.



(a) (b)

(c) (d)

Figure 4.6: (a), (b), (c), (d) *as before, for* $w = 0.2$, $\Delta = 0.6$ *and* $F = 0.1$.

In the non-integrable adiabatic gapless case, the current behaviour is very irregular. The impurity breaks the degeneracy of the eigenvalues, leading to the appearance of more frequencies in $\langle j_m \rangle$. Similarly to the integrable case, the frequencies which appear in the current are spread all over the $[0, 2\pi]$ range as seen in Fig. 4.6c. Again, this was expected by the structure of the spectra shown in Figs. 4.2d and 4.3b. As in the integrable case, P_k overlaps appear to be relatively uniform. A matrix element with large modulus appears at a frequency $f \approx 1.2$ corresponding to a period $T \approx 5$, however it can not be clearly observed in Fig. 4.6a due to the significant contribution of more small frequencies.

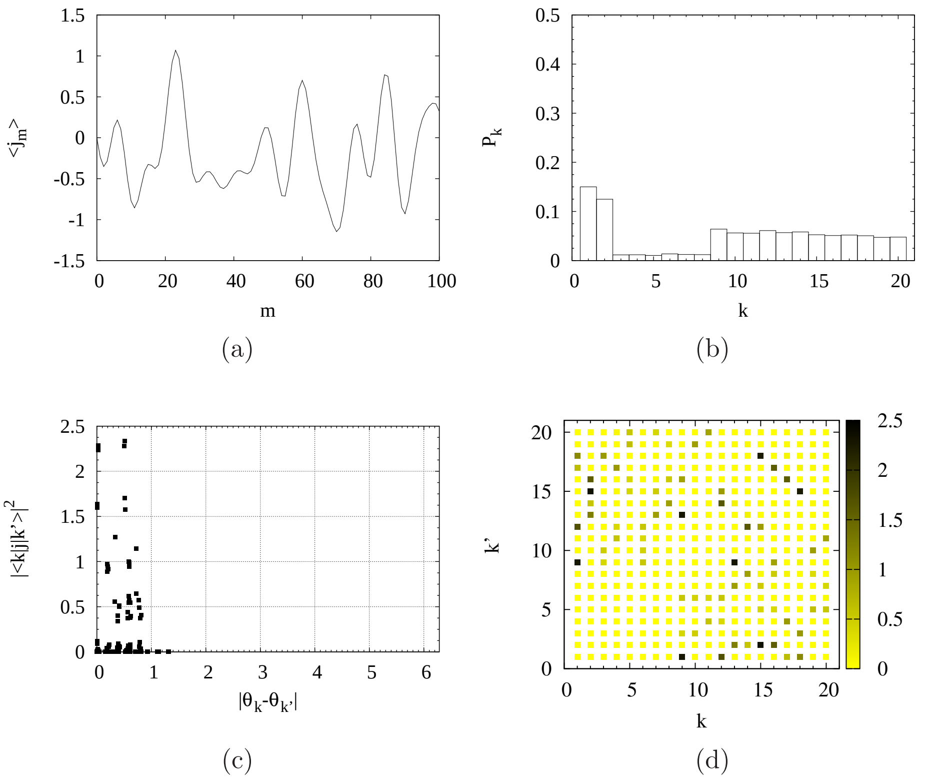
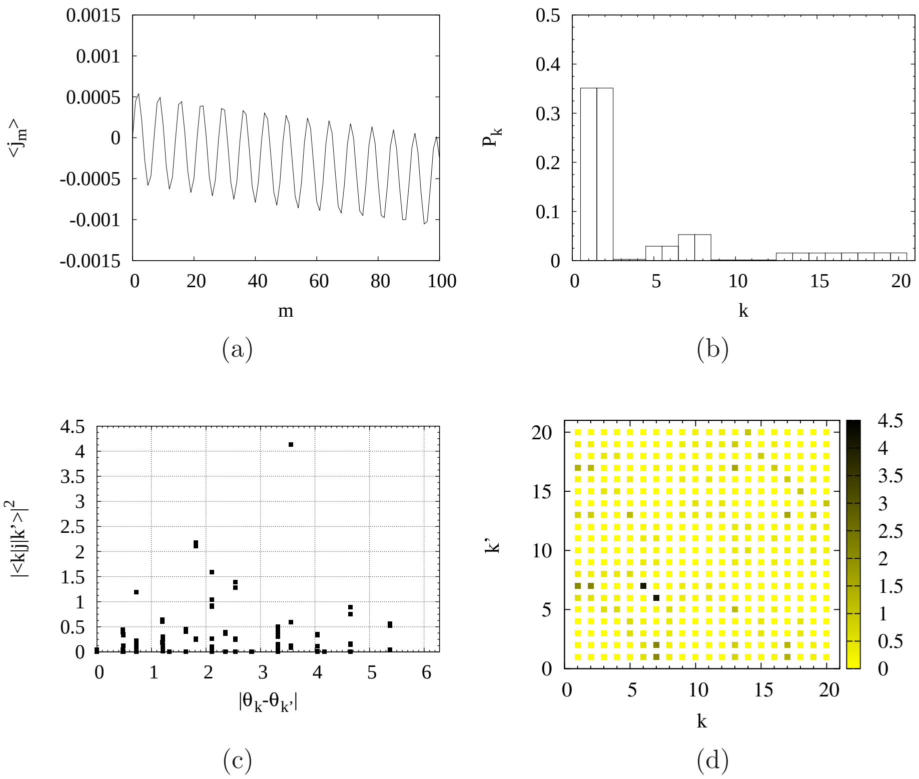


Figure 4.7: (a), (b), (c), (d) *as before, for* $w = 0.2$, $\Delta = 0.6$ *and* $F = 6.0$.

When the driving becomes stronger, as expected from what we see in Figs. 4.2d and 4.3h, the distribution of the frequencies shown in Fig. 4.7c becomes narrow again. We observe a few matrix elements with large modulus near $f \approx 0.5$ corresponding to $T \approx 13$. It is not clearly observable in Fig. 4.7a as the lift of the degeneracy causes more nearby frequencies to appear. As in the integrable case, we notice less uniform overlaps of the evolution operator eigenstates with the initial state.



(a)

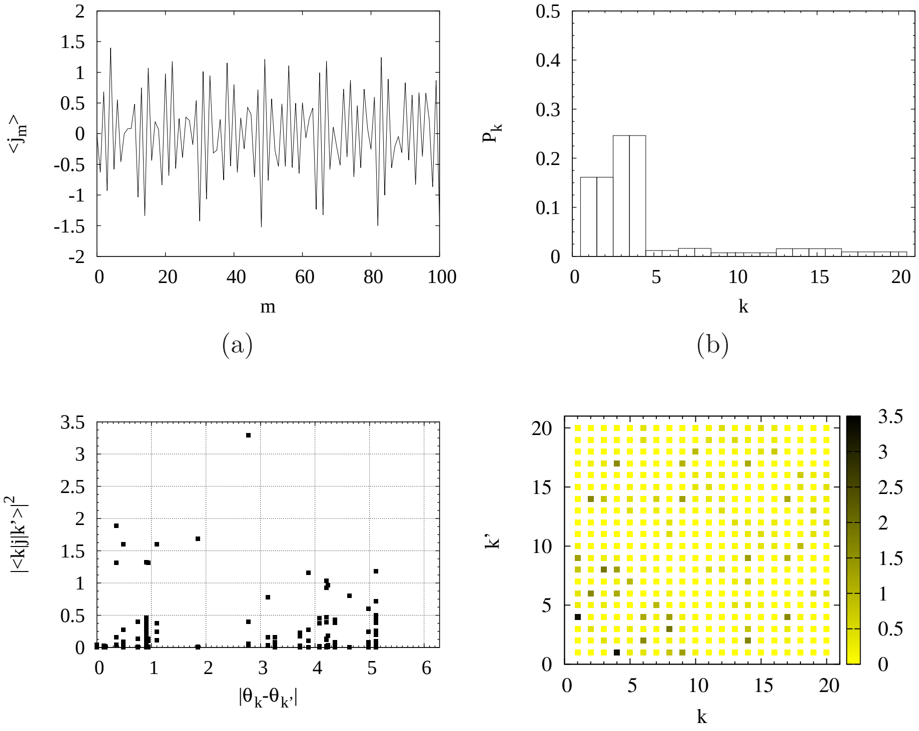
(b)

(c)

(d)

Figure 4.8: (a), (b), (c), (d) *as before, for* $w = 0.0$*,* $\Delta = 6.0$ *and* $F = 0.1$*.*

In the case of large interparticle interaction and a small field, we notice in Fig. 4.8a that the amplitude of the oscillations becomes very small. In Fig. 4.8c, it is confirmed that the frequency range is wide as we expected by the structure of the spectra shown in Figs. 4.2g and 4.3a. The overlaps of 2 evolution operator eigenstates with the initial state are very large. We observe one matrix element with large modulus at $f \approx 3.6$ in Fig. 4.8c, however in Fig. 4.8d we see that this value corresponds to the matrix element $\langle 6|j|7\rangle$ and its complex conjugate and, since the overlaps of states $|6\rangle$ and $|7\rangle$ with the initial state are small, we do not observe a large contribution of this frequency to the current expectation value. We observe a period $T \approx 7$ in Fig. 4.8a which corresponds to $f \approx 0.9$. In Fig. 4.8c no matrix elements corresponding to this frequency appear, however, there are matrix elements corresponding to $f_1 \approx 0.7$ and $f_2 \approx 1.2$, so the frequency appearing is probably their average contribution.



(a) (b)

(c) (d)

Figure 4.9: (a), (b), (c), (d) *as before, for* $w = 0.0$, $\Delta = 6.0$ *and* $F = 6.0$.

In Figs. 4.2g and 4.3g we see that near $F = 6.0$ and $\Delta = 6.0$ the spectra form clear lines, the distances among which are larger and differ more than those of the small Δ case. This is confirmed in Fig. 4.9c, where in contrast to the gapless case, the frequency range is spread all over the $[0, 2\pi]$ range. The appearance of large frequencies is reflected to the rapidly changing current as seen in Fig. 4.9a. The matrix element with the largest modulus corresponds to a frequency $f \approx 2.7$. It also corresponds to eigenstates of the evolution operator with large overlap with the initial state, maximizing the contribution of f in $\langle j_m \rangle$. Indeed, a period $\sim$ 2-3 is observed in Fig. 4.9a. The overlaps of the initial state with the evolution operator eigenstates are non-uniform in this case as well, with 4 of them out of the total 20 summing up to ~ 0.8.

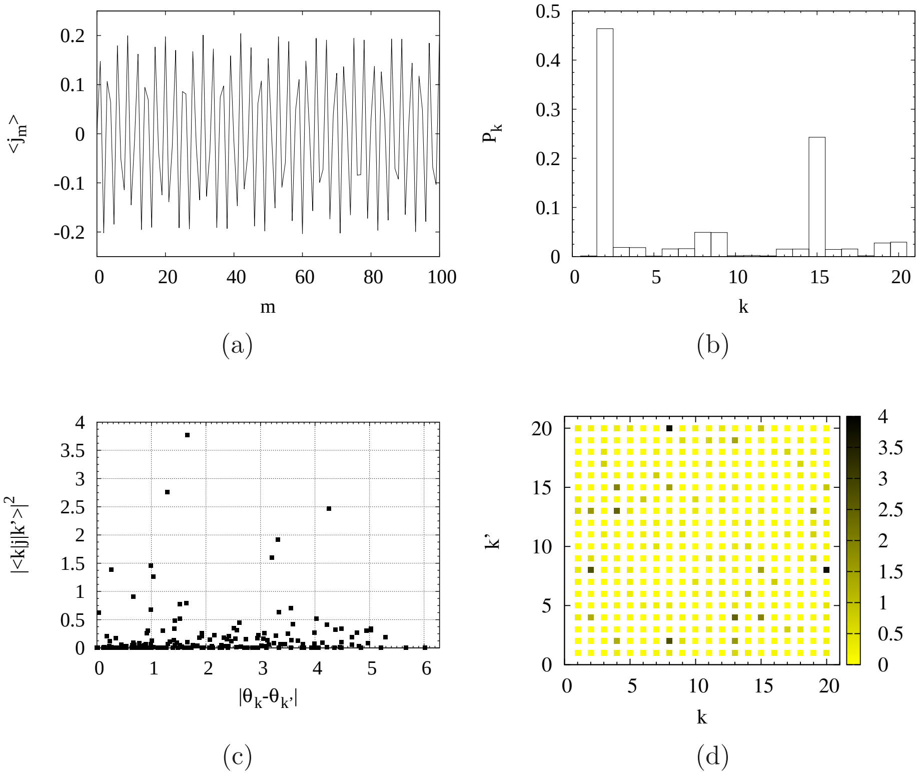
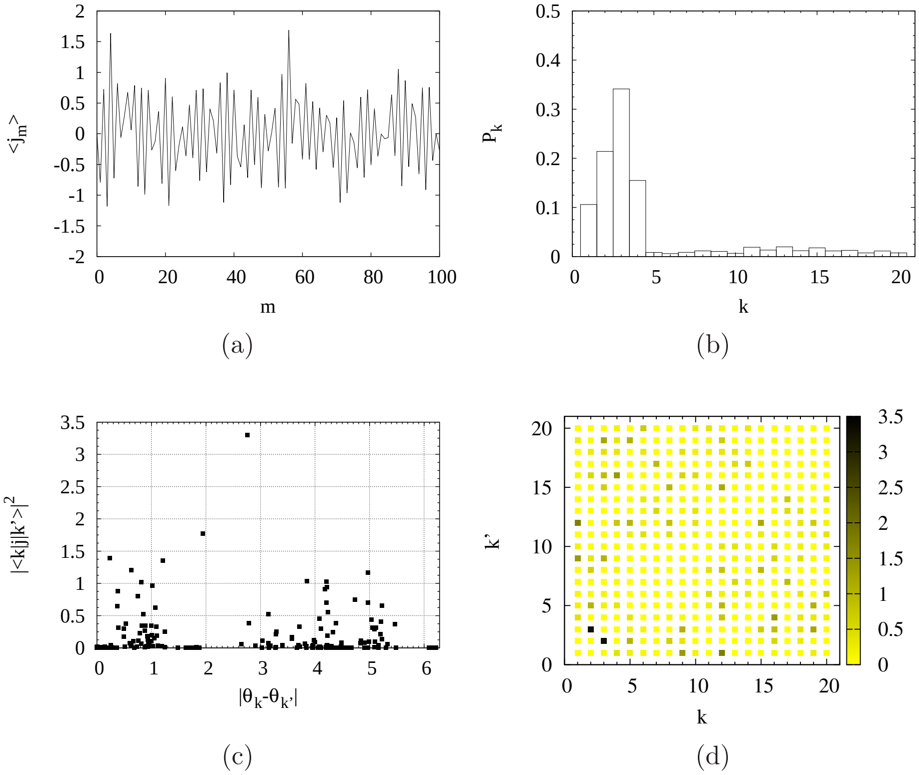


Figure 4.10: (a), (b), (c), (d) *as before, for* $w = 0.2$, $\Delta = 6.0$ *and* $F = 0.1$.

In the non-integrable case, at the adiabatic limit of the gapped system, as seen in Figs. 4.2h and 4.3b, there are eigenvalues all over the $[0, 2\pi]$ range similarly to the $w = 0.0$ case. The breaking of the degeneracy leads to the appearance of more frequencies as we see in Fig. 4.10c. The overlaps of the evolution operator eigenstates with the initial state are not uniform and most contribution comes from only 2 overlaps. In Fig. 4.10a we observe a frequency $T \approx 3$ which, commonly to the non-integrable case, is not directly observable in Fig. 4.10c.



(a)

(b)

(c)

(d)

Figure 4.11: (a), (b), (c), (d) *as before, for* $w = 0.2$, $\Delta = 6.0$ *and* $F = 6.0$.

For strong driving, as we see in Figs. 4.2h and 4.3h the distribution of the eigenvalues is again similar to the integrable case. More frequencies appear in this case due to the lift of the degeneracy and the current behaviour is complex as we see in Fig. 4.11a. The analysis of the initial state to the evolution operator basis is not uniform in this case either. We notice a matrix element with large modulus at $f \approx 2.7$ corresponding to $T \approx 2$ which is observable in $\langle j_m \rangle$, since the related states overlaps with the initial state are also large.

4.4 Current of the gapped system at the adiabatic limit

In Fig. 4.8a, which presents the expectation value of the particle current for the adiabatic gapped integrable case, it is evident that the center of the current oscillations tends to move towards negative values. This observation led us to further investigate the behaviour of the integrable system at the adiabatic limit. A roughly linear increase of $\langle j_m \rangle$ is observed for $\Delta > 2$, which is a point where a transition of the energy spectrum from gapless to gapped occurs as mentioned at the beginning of this chapter. In Fig. 4.12 we present the particle current for various values of Δ, calculated up to $m_{max} = 400$. In addition to the increase of the current in the gapped case, a sudden decrease of the amplitude by one order of magnitude is observed.

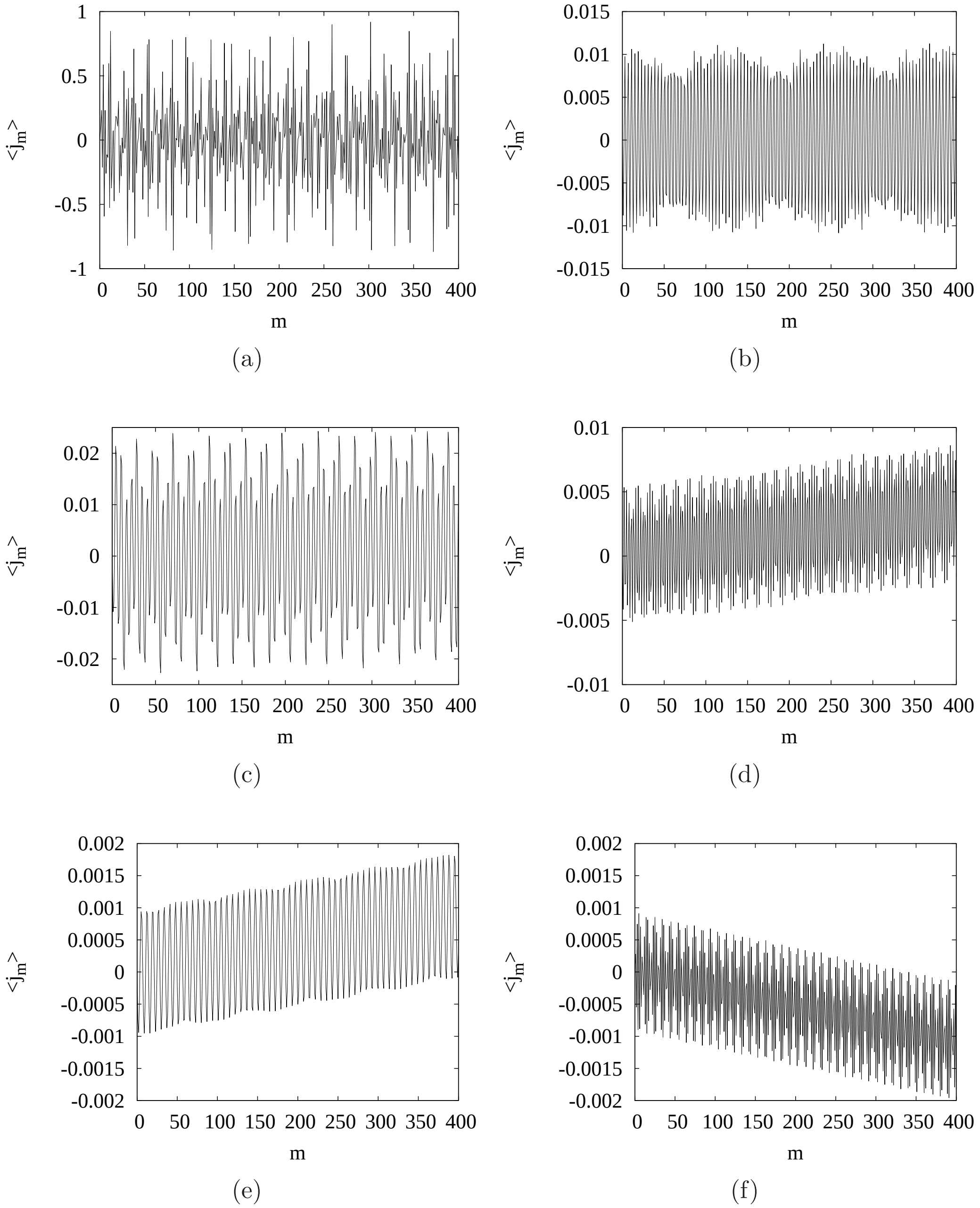


Figure 4.12: *Current for the integrable case for* $F = 0.1$ *and* (a) $\Delta = 1.0$, (b) $\Delta = 1.9$, (c) $\Delta = 2.0$, (d) $\Delta = 2.1$, (e) $\Delta = 4.0$, (f) $\Delta = 5.0$.

Chapter 5

Conclusions

In the present work we numerically studied the time evolution of a system of correlated hard-core bosons driven by a static field by calculating the evolution operator of the system over one Bloch period. After diagonalizing the evolution operator, we studied its spectrum and related it to the expectation value of the particle current.

Concerning the effect of the system parameters on the spectrum of the evolution operator, we found that the spectrum undergoes a clear transition from irregular to regular as the intensity of the field increases. The field intensity at which this transition occurs increases with the repulsive interaction. In addition, the lines of the spectrum diverge as the repulsive interaction becomes larger. In all cases, the presence of a lattice impurity lifts the degeneracy of the eigenvalues, causing a splitting of the lines in the spectrum.

After expressing the expectation value of the particle current in terms of the evolution operator eigenfunctions and eigenvalues and calculating the terms in (2.14), we confirmed that in most cases the behaviour of the current can be predicted by the structure of the spectrum given the system parameters. However, this does not happen in all cases, as most frequencies appearing in the spectrum exhibit low contribution due to the small amplitude of the respective current matrix elements. For the gapped integrable case, at the adiabatic limit, we observed a roughly linear increase of the current expectation value.

This work is an introductory step towards the study of the dynamics of quantum many-body systems. A perspective for the future is the investigation of larger systems. Moreover, a semi-analytical study of the evolution operator may be feasible for the integrable case.

Appendix A

Source code

```
c drive system
c base code
c spin/hcb version
      implicit double precision(a-h,o-z)
      parameter (s=0.5d0, ibase=2, n=6, id=ibase**n)
      parameter (sz0=+0.0d0)
      parameter (ir=6*5*4/2/3)
      parameter (mspsm=2*n)
      parameter (gspsm=+1.0d0, w=0.000001d0)
      parameter (field=0.1d0)
      parameter (delta=0.6d0)
      parameter (nt=200)
      parameter (matz=1)
c
      dimension ib(0:n-1),ib2(0:n-1)
      dimension mbs(id),msb(ir)
      dimension sz(0:ibase-1),cjp(0:ibase-1),cjm(0:ibase-1)
      dimension eszsz(ir),eszsz2(ir)
      dimension espsm(ir,mspsm),ispsm(ir,mspsm),nspsm(ir)
      dimension a(ir,ir),d(ir),e(ir),e2(ir),tau(2,ir)
      dimension zr(ir,ir),zi(ir,ir),cr(ir,ir),ci(ir,ir)
      dimension ur(ir,ir),ui(ir,ir),unr(ir,ir),uni(ir,ir)
      dimension wr(ir),wi(ir),zur(ir,ir),zui(ir,ir),ov2(ir)
      dimension fv1(ir),fv2(ir),fv3(ir),theta(ir),ov(ir)
      dimension utr(ir,ir),uti(ir,ir),uar(ir,ir),uai(ir,ir)
      dimension uudr(ir,ir),uudi(ir,ir),udr(ir,ir),udi(ir,ir)
      dimension usr(ir,ir),usi(ir,ir),ut1r(ir,ir),ut1i(ir,ir)
      dimension dui(ir,ir),dur(ir,ir),ut11r(ir,ir),ut11i(ir,ir)
      dimension t(ir,ir),c(ir,ir),yer(ir),yei(ir)
      dimension znr(ir,ir),zni(ir,ir),pc(ir,ir)
      dimension yr(ir),yi(ir),yr0(ir),yi0(ir),zuar(ir),zuai(ir)
      dimension yr2(ir),yi2(ir),yr3(ir),yi3(ir),yr4(ir),yi4(ir)
      character*20 oname
c
      pi=4.0d0*datan(1.0d0)
      idum=-137849
      write(6,*) 'o-name'
      read(5,'(a20)') oname
      open(unit=8,file=oname)
c      do 900 field=0.1D0, 8.0D0, 0.05D0
      write(6,*) 's=',s,'  n=',n,'  delta=',delta, ' field=',field
      write(6,*)
      call oper(s,ibase,sz,cjp,cjm)
      call sub(n,ibase,sz,id,ib,sz0,mbs,ir,msb)
c matrix elements
      call szsz(n,ibase,sz,id,mbs,ir,msb,eszsz,eszsz2,ib)
      call spsm(n,ibase,cjp,cjm,id,mbs,ir,msb,
     1mspsm,espsm,ispsm,nspsm,ib,ib2)
c initializing matrices
      do 10 i=1,ir
      do 10 j=1,ir
      uar(i,j)=0.0d0
      uai(i,j)=0.0d0
10    continue
      do 11 i=1,ir
      yer(i)=0.0d0
      yei(i)=0.0d0
      do 11 j=1,ir
      znr(i,j)=0.0d0
11    zni(i,j)=0.0d0
      do 12 i=1,ir
12    uar(i,i)=1.0d0
c Hamiltonian (2.1)
      time=0.0d0
      do 700 it=0,nt
      dt=+2.0d0*pi/nt
      time=time+dt
      if(it.eq.1) time=0.5d0*dt
      if(it.eq.0) time=0.0d0
      do 20 i=1,ir
      do 20 j=1,ir
      a(i,j)=0.0d0
      ur(i,j)=0.0d0
20    ui(i,j)=0.0d0
c
      do 200 i=1,ir
200   a(i,i)=+delta*eszsz(i)+w*eszsz2(i)
      do 100 i=1,ir
      do 100 j=1,nspsm(i)
      if(i.gt.ispsm(i,j)) then
      a(i,ispsm(i,j))=a(i,ispsm(i,j))+dcos(time)*gspsm*espsm(i,j)
      a(ispsm(i,j),i)=a(ispsm(i,j),i)+dsin(time)*gspsm*espsm(i,j)
      endif
      if(i.lt.ispsm(i,j)) then
      a(i,ispsm(i,j))=a(i,ispsm(i,j))-dsin(time)*gspsm*espsm(i,j)
      a(ispsm(i,j),i)=a(ispsm(i,j),i)+dcos(time)*gspsm*espsm(i,j)
      endif
100   continue
c
c diagonalize Hamiltonian
      do 30 i=1,ir
      do 30 j=1,ir
      zr(i,j)=0.0d0
30    zi(i,j)=0.0d0
      do 40 i=1,ir
40    zr(i,i)=1.0d0
      call htrid3(ir,ir,a,d,e,e2,tau)
      call tql2(ir,ir,d,e,zr,ierr)
      call htrib3(ir,ir,a,tau,ir,zr,zi)
      if(ierr.gt.0) write(6,*) 'ierr=',ierr
      if(it.eq.0) e0=d(1)
      do 777 i=1,ir
777   write(10,*) time/(2*pi/n),d(i)-e0
c
      if(it.eq.0) then
c current
      do 290 i=1,ir
      do 290 j=1,ir
      t(i,j)=0.0d0
290   c(i,j)=0.0d0
      do 300 i=1,ir
      do 300 j=1,nspsm(i)
      if(i.gt.ispsm(i,j))
     1t(i,ispsm(i,j))=t(i,ispsm(i,j))+gspsm*espsm(i,j)
      if(i.lt.ispsm(i,j))
     1t(ispsm(i,j),i)=t(ispsm(i,j),i)+gspsm*espsm(i,j)
300   continue
      do 310 i=1,ir
      do 310 j=1,ir
310   c(i,j)=t(i,j)
      do 320 i=1,ir
      do 320 j=1,ir
      if(abs(t(i,j)).gt.1.d-6) t(j,i)=+t(i,j)
320   if(abs(c(i,j)).gt.1.d-6) c(j,i)=-c(i,j)
c initial state
      do 44 i=1,ir
      yr0(i)=zr(i,1)
      yi0(i)=zi(i,1)
      yr(i)=zr(i,1)
      yi(i)=zi(i,1)
      do 44 j=1,ir
      znr(i,j)=zr(i,j)
44    zni(i,j)=zi(i,j)
      call current(ir,time,t,c,yr,yi,xcr,xci,sumr,sumi)
      write(8,'(5f15.6)') time/(2*pi/n),xcr,xci,sumr,sumi
      goto 700
      endif
c evolution operator
      do 50 i=1,ir
      do 50 j=1,ir
      ur(i,j)=0.0d0
50    ui(i,j)=0.0d0
      do 60 i=1,ir
      do 60 j=1,ir
      do 60 k=1,ir
      er=dcos(d(k)/field*dt)
      ei=-dsin(d(k)/field*dt)
      ur(i,j)=ur(i,j)+(zr(i,k)*zr(j,k)+zi(i,k)*zi(j,k))*er
     1               -(zi(i,k)*zr(j,k)-zr(i,k)*zi(j,k))*ei
      ui(i,j)=ui(i,j)+(zr(i,k)*zr(j,k)+zi(i,k)*zi(j,k))*ei
     1               +(zi(i,k)*zr(j,k)-zr(i,k)*zi(j,k))*er
60    continue
      call evolve(ir,ur,ui,yr,yi,yr2,yi2)
      call current(ir,time,t,c,yr,yi,xcr,xci,sumr,sumi)
      write(8,'(5f15.6)') time/(2*pi/n),xcr,xci,sumr,sumi
      call operprod(ir,ur,ui,uar,uai,utr,uti)
      call evolve2(ir,utr,uti,yr0,yi0,yr3,yi3)
      call current(ir,time,t,c,yr3,yi3,xcr,xci,sumr,sumi)
      write(12,'(5f15.6)') time/(2*pi/n),xcr,xci,sumr,sumi
c U(2pi)*U(2pi)=U(4pi) check
      if (it.eq.nt) then
      do 490 i=1,ir
      do 490 j=1,ir
      ut11r(i,j)=utr(i,j)
      ut11i(i,j)=uti(i,j)
      ut1r(i,j)=utr(i,j)
      ut1i(i,j)=uti(i,j)
490   continue
      endif
700   continue
c diagonalize evolution operator
      call cg(ir,ir,ut1r,ut1i,wr,wi,matz,zur,zui,fv1,fv2,fv3,ierr2)
      print*,"ierr=", ierr2
c |<k|0>|^2
      sum=0.0D0
      do k=1,ir
         ov(k)=0.0D0
      do i=1,ir
         ov(k)=ov(k)+(zur(i,k)*znr(i,1)+zui(i,k)*zni(i,1))**2
     &              +(zur(i,k)*zni(i,1)-zui(i,k)*znr(i,1))**2
      enddo
      write(31,*) k, ov(k)
      sum=sum+ov(k)
      enddo
      print*,sum
      open(14,file='eigenvalues.dat')
      do 495 i=1,ir
      theta(i)=datan(wi(i)/wr(i))
      if(wr(i).lt.0) then
      if(wi(i).ge.0) then
      theta(i)=theta(i)+pi
      else
      theta(i)=theta(i)-pi
      endif
      endif
      write(14,*) wr(i),wi(i),theta(i),wr(i)**2+wi(i)**2
c      write(30,*) field,theta(i)
495   continue
c900   continue
      close(14)
      close(30)
c current j_m (2.14)
      do 850 m=0,100
      time=2.0d0*pi*m
      do 799 i=1,ir
      yr4(i)=0.0d0
      yi4(i)=0.0d0
      do 799 j=1,ir
      do 799 k=1,ir
      yr4(i)=yr4(i)
     &+yr0(j)*(zur(i,k)*zur(j,k)+zui(i,k)*zui(j,k))*dcos(theta(k)*m)
     &-yr0(j)*(zui(i,k)*zur(j,k)-zur(i,k)*zui(j,k))*dsin(theta(k)*m)
      yi4(i)=yi4(i)
     &+yr0(j)*(zur(i,k)*zur(j,k)+zui(i,k)*zui(j,k))*dsin(theta(k)*m)
     &+yr0(j)*(zui(i,k)*zur(j,k)-zur(i,k)*zui(j,k))*dcos(theta(k)*m)
799   continue
      call current(ir,time,t,c,yr4,yi4,xcr,xci,sumr,sumi)
      write(25,'(5f15.6)') real(m),xcr,xci,sumr,sumi
      if (m.eq.1) then
      do 801 k=1,ir
      do 805 i=1,ir
      zuar(i)=zur(i,k)
805   zuai(i)=zui(i,k)
      call current(ir,time,t,c,zuar,zuai,xcr,xci,sumr,sumi)
801   write(29,'(5f25.16)') real(k),xcr,xci,sumr,sumi
      call current2(ir,time,t,c,zur,zui,cr,ci,sumr,sumi)
      do i=1,ir
```

```
         do j=1,ir
            pc(i,j)=cr(i,j)**2+ci(i,j)**2
            write(74,*) i,j, pc(i,j)
            write(75,*) theta(i), theta(j),pc(i,j)
         enddo
         write(74,*)
      enddo
      do i=1,ir
         write(71,'(5f25.16)') (cr(i,j),j=1,ir)
         write(72,'(5f25.16)') (ci(i,j),j=1,ir)
      enddo
c      print*,sumr,sumi
      endif
850   continue
      close(8)
      stop
      end
c
      subroutine current(ir,time,t,c,yr,yi,xcr,xci,sumr,sumi)
c calculates the particle current expectation value
      implicit double precision(a-h,o-z)
      dimension yr(ir),yi(ir),t(ir,ir),c(ir,ir)
      xcr=0.0d0
      xci=0.0d0
      do 100 i=1,ir
      do 100 j=1,ir
      xcr=xcr-dsin(time)*(+yr(i)*yr(j)+yi(i)*yi(j))*t(i,j)
     1       +dcos(time)*(+yi(i)*yr(j)-yr(i)*yi(j))*c(i,j)
      xci=xci+dsin(time)*(-yi(i)*yr(j)+yr(i)*yi(j))*t(i,j)
     1       +dcos(time)*(+yr(i)*yr(j)+yi(i)*yi(j))*c(i,j)
100   continue
c
      sumr=0.0d0
      sumi=0.0d0
      do 200 i=1,ir
      sumr=sumr+(yr(i)*yr(i)+yi(i)*yi(i))
200   sumi=sumi+(yr(i)*yi(i)-yi(i)*yr(i))
      return
      end
c
```

```
      subroutine current2(ir,time,t,c,zur,zui,cr,ci,sumr,sumi)
c calculates the current matrix elements
      implicit double precision(a-h,o-z)
      dimension zur(ir,ir),zui(ir,ir),t(ir,ir),c(ir,ir)
      dimension cr(ir,ir),ci(ir,ir)
      do 100 k=1,ir
      do 100 kp=1,ir
      cr(k,kp)=0.0d0
      ci(k,kp)=0.0d0
      do 100 i=1,ir
      do 100 j=1,ir
      cr(k,kp)=cr(k,kp)
     1       -dsin(time)*(+zur(i,k)*zur(j,kp)+zui(i,k)*zui(j,kp))*t(i,j)
     1       +dcos(time)*(+zui(i,k)*zur(j,kp)-zur(i,k)*zui(j,kp))*c(i,j)
      ci(k,kp)=ci(k,kp)
     1       +dsin(time)*(-zui(i,k)*zur(j,kp)+zur(i,k)*zui(j,kp))*t(i,j)
     1       +dcos(time)*(+zur(i,k)*zur(j,kp)+zui(i,k)*zui(j,kp))*c(i,j)
100   continue
      sumr=0.0d0
      sumi=0.0d0
      do 200 i=1,ir
      do 200 j=1,ir
      sumr=sumr+(zur(i,j)*zur(i,j)+zui(i,j)*zui(i,j))
200   sumi=sumi+(zur(i,j)*zui(i,j)-zui(i,j)*zur(i,j))
      return
      end
c
      subroutine diagcheck(ir,zur,zui,ut1r,ut1i,dur,dui)
c checks the diagonalization of the evolution operator through
c the eigenvalue problem
      implicit double precision(a-h,o-z)
      dimension zur(ir,ir),zui(ir,ir)
      dimension dur(ir,ir),dui(ir,ir),ut1r(ir,ir),ut1i(ir,ir)
      do 480 i=1,ir
      do 480 j=1,ir
      dur(i,j)=0.0d0
      dui(i,j)=0.0d0
      do 480 k=1,ir
      dur(i,j)=dur(i,j)+ut1r(i,k)*zur(k,j)-ut1i(i,k)*zui(k,j)
      dui(i,j)=dui(i,j)+ut1r(i,k)*zui(k,j)+ut1i(i,k)*zur(k,j)
```

```
480   continue
      return
      end
c
      subroutine evolve(ir,ur,ui,yr,yi,yr2,yi2)
c time evolution acting evolution operator increments on previous
c state
      implicit double precision(a-h,o-z)
      dimension ur(ir,ir),ui(ir,ir)
      dimension yr(ir),yi(ir),yr2(ir),yi2(ir)
      do 10 i=1,ir
      yr2(i)=0.0d0
10    yi2(i)=0.0d0
      do 100 i=1,ir
      do 100 j=1,ir
      yr2(i)=yr2(i)+(ur(i,j)*yr(j)-ui(i,j)*yi(j))
100   yi2(i)=yi2(i)+(ui(i,j)*yr(j)+ur(i,j)*yi(j))
      do 20 i=1,ir
      yr(i)=yr2(i)
20    yi(i)=yi2(i)
      return
      end
c
      subroutine evolve2(ir,utr,uti,yr0,yi0,yr3,yi3)
c time evolution acting evolution operator on initial state
      implicit double precision(a-h,o-z)
      dimension utr(ir,ir),uti(ir,ir)
      dimension yr0(ir),yi0(ir),yr3(ir),yi3(ir)
      do 10 i=1,ir
      yr3(i)=0.0d0
10    yi3(i)=0.0d0
      do 100 i=1,ir
      do 100 j=1,ir
      yr3(i)=yr3(i)+(utr(i,j)*yr0(j)-uti(i,j)*yi0(j))
100   yi3(i)=yi3(i)+(uti(i,j)*yr0(j)+utr(i,j)*yi0(j))
      return
      end
c
      subroutine usq(ir,utr,uti,usr,usi)
c calculates the square of a complex matrix
```

```
      implicit double precision(a-h,o-z)
      dimension usr(ir,ir),usi(ir,ir)
      dimension utr(ir,ir),uti(ir,ir)
      do 450 i=1,ir
      do 450 j=1,ir
      usr(i,j)=0.d0
      usi(i,j)=0.d0
      do 450 k=1,ir
      usr(i,j)=usr(i,j)+utr(i,k)*utr(k,j)-uti(i,k)*uti(k,j)
      usi(i,j)=usi(i,j)+uti(i,k)*utr(k,j)+utr(i,k)*uti(k,j)
450   continue
      return
      end
c
      subroutine operprod(ir,ur,ui,uar,uai,utr,uti)
c calculates the time ordered product
      implicit double precision(a-h,o-z)
      dimension uar(ir,ir),uai(ir,ir),ur(ir,ir),ui(ir,ir)
      dimension utr(ir,ir),uti(ir,ir)
      do 450 i=1,ir
      do 450 j=1,ir
      utr(i,j)=0.d0
      uti(i,j)=0.d0
      do 450 k=1,ir
      utr(i,j)=utr(i,j)+ur(i,k)*uar(k,j)-ui(i,k)*uai(k,j)
      uti(i,j)=uti(i,j)+ui(i,k)*uar(k,j)+ur(i,k)*uai(k,j)
450   continue
      do 452 i=1,ir
      do 452 j=1,ir
      uar(i,j)=utr(i,j)
452   uai(i,j)=uti(i,j)
      return
      end
c
      subroutine unitarity(ir,utr,uti,uudr,uudi,udr,udi)
c checks the unitarity of a matrix
      implicit double precision(a-h,o-z)
      dimension utr(ir,ir),uti(ir,ir),udr(ir,ir),udi(ir,ir)
      dimension uudr(ir,ir),uudi(ir,ir)
      do 460 i=1,ir
      do 460 j=1,ir
      udr(i,j)=utr(j,i)
460   udi(i,j)=-uti(j,i)
      do 480 i=1,ir
      do 480 j=1,ir
      uudr(i,j)=0.
      uudi(i,j)=0.
      do 480 k=1,ir
      uudr(i,j)=uudr(i,j)+utr(i,k)*udr(k,j)-uti(i,k)*udi(k,j)
480   uudi(i,j)=uudi(i,j)+uti(i,k)*udr(k,j)+utr(i,k)*udi(k,j)
      return
      end
c
      subroutine sub(n,ibase,sz,id,ib,sz0,mbs,ir,msb)
c creation of the state space
      implicit double precision(a-h,o-z)
      dimension sz(0:ibase-1),mbs(id),msb(ir)
      dimension ib(0:n-1)
      ids=0
      do 100 is=0,id-1
      call sb(ibase,n,is,ib)
      sum=0.0
      do 200 i=0,n-1
200   sum=sum+sz(ib(i))
      if(abs(sum-sz0).lt.(1.e-6)) then
      ids=ids+1
      mbs(is)=ids
      msb(ids)=is
      endif
100   continue
      if(ids.ne.ir) then
      write(6,*) 'error ids.ne.ir'
      endif
      return
      end
c
      subroutine szsz(n,ibase,sz,id,mbs,ir,msb,eszsz,eszsz2,ib)
c calculation of the diagonal elements of the Hamiltonian
      implicit double precision(a-h,o-z)
      dimension sz(0:ibase-1),eszsz(ir),eszsz2(ir)
      dimension mbs(id),msb(ir)
      dimension ib(0:n-1)
      do 100 is=1,ir
      call sb(ibase,n,msb(is),ib)
      eszsz(is)=0.0
      eszsz2(is)=0.0
      do 200 i=0,n-2
200   eszsz(is)=eszsz(is)+sz(ib(i))*sz(ib(i+1))
c p.b.c.
      eszsz(is)=eszsz(is)+sz(ib(0))*sz(ib(n-1))
c
      eszsz2(is)=sz(ib(0))
100   continue
      return
      end
c
      subroutine spsm(n,ibase,cjp,cjm,id,mbs,ir,msb,
     1mspsm,espsm,ispsm,nspsm,ib,ib2)
c calculation of the non-diagonal elements of the Hamiltonian
      implicit double precision(a-h,o-z)
      dimension cjp(0:ibase-1),cjm(0:ibase-1)
      dimension mbs(id),msb(ir)
      dimension espsm(ir,mspsm),ispsm(ir,mspsm),nspsm(ir)
      dimension ib(0:n-1),ib2(0:n-1)
      do 100 is=1,ir
      nspsm(is)=0
      call sb(ibase,n,msb(is),ib)
      do 200 i=0,n-2
      do 202 ii=0,n-1
202   ib2(ii)=ib(ii)
      ib2(i)=ib(i)+1
      if(ib2(i).eq.ibase) goto 200
      ib2(i+1)=ib(i+1)-1
      if(ib2(i+1).eq.-1) goto 200
      nspsm(is)=nspsm(is)+1
      if(nspsm(is).gt.mspsm) then
      write(6,*) 'nspsm.gt.mspsm'
      stop
      endif
      call bs(ibase,n,ib2,isnew)
      ispsm(is,nspsm(is))=mbs(isnew)
      espsm(is,nspsm(is))=cjp(ib(i))*cjm(ib(i+1))
200   continue
c p.b.c.
      do 203 ii=0,n-1
203   ib2(ii)=ib(ii)
      ib2(n-1)=ib(n-1)+1
      if(ib2(n-1).eq.ibase) goto 100
      ib2(0)=ib(0)-1
      if(ib2(0).eq.-1) goto 100
      nspsm(is)=nspsm(is)+1
      call bs(ibase,n,ib2,isnew)
      ispsm(is,nspsm(is))=mbs(isnew)
      espsm(is,nspsm(is))=cjp(ib(n-1))*cjm(ib(0))
100   continue
      return
      end
c
      subroutine oper(s,ibase,sz,cjp,cjm)
c creation spin operators
      implicit double precision(a-h,o-z)
      dimension sz(0:ibase-1)
      dimension cjp(0:ibase-1),cjm(0:ibase-1)
      do 100 i=0,ibase-1
100   sz(i)=-s+i
      do 200 i=0,ibase-1
200   cjp(i)=sqrt(s*(s+1)-sz(i)*(sz(i)+1))
      do 300 i=0,ibase-1
300   cjm(i)=sqrt(s*(s+1)-sz(i)*(sz(i)-1))
      return
      end
c
      subroutine sb(ibase,n,is,ib)
c converts a decimal to a binary
      implicit double precision(a-h,o-z)
      dimension ib(0:n-1)
      do 10 j=0,n-1
10    ib(j)=0
      ii=0
      is2=is
```

```
1       is1=is2
        if(is1.ge.ibase) then
        is2=is1/ibase
        ib(ii)=mod(is1,ibase)
        ii=ii+1
        goto 1
        endif
        ib(ii)=is2
c check
        isum=0
        do 100 j=0,n-1
100     isum=isum+ib(j)*(ibase**j)
        if(isum.ne.is) then
        write(6,*) 'error in conversion sb'
        stop
        endif
        return
        end
c
        subroutine bs(ibase,n,ib,is)
c converts a binary to a decimal
        implicit double precision(a-h,o-z)
        dimension ib(0:n-1)
        is=0
        do 100 j=0,n-1
100     is=is+ib(j)*ibase**j
        return
        end
```

Bibliography

[1] M. Raizen, C. Salomon, Q. Niu, Phys. Today **50**, 30 (1997).

[2] F. Bloch, Z. Phys. **52**, 555 (1929).

[3] S. R. Wilkinson, C. F. Bharucha, K. W. Madison, Q. Niu, M. G. Raizen, Phys. Rev. Lett. **76**, 4512 (1996).

[4] M. B. Dahan, E. Peik, J. Reichel, Y. Castin, C. Salomon, Phys. Rev. Lett. **76**, 4508 (1996).

[5] C. Waschke, H. G. Roskos, R. Schwedler, K. Leo, H. Kurz, K. Kohler, Phys. Rev. Lett. **70**, 3319 (1993).

[6] G. Ferrari, N. Poli, F. Sorrentino, G. M. Tino, Phys. Rev. Lett. **97**, 060402 (2006).

[7] O. Morsch, J. H. Muller, M. Cristiani, D. Ciampini, E. Arimondo, Phys. Rev. Lett. **87**, 140402 (2001).

[8] A. R. Kolovsky, Phys. Rev. Lett. **90**, 213002 (2003).

[9] M. Mierzejewski, P. Prelovšek, Phys. Rev. Lett. **105**, 186405 (2010).

[10] M. Mierzejewski, J. Bonča, P. Prelovšek, Phys. Rev. Lett. **107**, 126601 (2011).

[11] T. Oka, R. Arita, H. Aoki, Phys. Rev. Lett. **91**, 0664006 (2003).

[12] E. Lieb, T. Schultz, D. Mattis, Ann. Phys. **16**, 407 (1961).

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, "Numerical Recipes in Fortran 77: The Art of Scientific Computing, Volume 1", Second edition, Cambridge University Press, 1992