Effect of Dynamic Torsional Disorder on Exciton Migration in Linear Conjugated Polymer chains

A Dissertation as a course requirement for

Master of Science (Chemistry)

M S Vijay Bhaskar

(Regd.No.: 17108)

SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING
(Deemed to be University)

Department of Chemistry
Prasanthi Nilayam Campus

March 2019

DEPARTMENT OF CHEMISTRY
Sri Sathya Sai Institute of Higher Learning
(Deemed to be University)
Prashanthi Nilayam Campus
Andhra Pradesh - 515134

Certificate

This is to certify that this Dissertation titled **Effect of Dynamic Torsional Disorder on Exciton Transport rate in linear Conjugated Polymers** is a bona fide record of the original work done under my supervision as a Course requirement for the Degree of Master of Science (Chemistry).

Dr. B Siva Kumar
Dissertation Supervisor

Place: Prashanthi Nilayam
Date: 15^{th} March, 2019

Dr. C N Sundaresan
Head of the Department

Declaration

The Dissertation titled **Effect of Dynamic Torsional Disorder on Exciton Transport rate in linear Conjugated Polymers** was carried out by me under the supervision of **Dr. B. Siva Kumar**, Department of Chemistry, Prasanthi Nilayam Campus, as a Course requirement for the Degree of Master of Science (Chemistry) and has not formed the basis for the award of any degree, diploma or any other such title by this or any other University .

Place: Prashanthi Nilayam
Date: 15^{th} March, 2019

M S Vijay Bhaskar
Regd. No. 17108
II M. Sc. Chemistry
Prashanti Nilayam Campus

ACKNOWLEDGMENTS

I offer my gratitude to Bhagawan Sri Sathya Sai Baba, founder Chancellor of our University, for His divine guidance. I express my heartfelt sense of love and gratitude to my parents for having molded me into what I am.

I am indebted to the key administrators of our university, the revered Vice-Chancellor **Prof. K B R Varma**, the Registrar **Dr. B Sai Giridhar** and the Controller of Examinations **Sri Sanjay Sahni** for having been instrumental in my stay as student of this hallowed institution. I would like to specially thank **Prof. G Nageswara Rao**, Dean, Faculty of Sciences and **Dr. C N Sundaresan**, Head of Department of Chemistry for their constant support.

I would like to express my gratitude to my supervisor **Dr. B. Siva Kumar**, for his guidance throughout this thesis work. Without his help and care, this thesis work could not have been accomplished. I am thankful to **Dr. N Uday Kiran**, Department of Mathematics and Computer Science for his valuable inputs and all the insightful discussions during the course of this project.

I am grateful to **Prof. Chelli Janardhana** for nurturing my interests in Theoretical Chemistry and **Dr.R Sai Sathish** for having mentored me in the initial days of research as a master's student. I am indebted to my teachers from undergraduate days, **Dr. J K Kiran**, for initiating me into programming with Python and **Sri Darshan Gera**, for patiently teaching me the fundamental aspects of Linear Algebra and Differential Equations.

I am grateful to all my teachers: Dr.S Jagadeeswara Rao, Dr.D Rajesh Babu, Dr. S Prathap Chandran and Dr. V N Ravi Kishore Vutukuri for their guidance.

I would like to specially thank brother Hari Kumar Yadalam for having initiated me into this wonderful world of 'Quantum Dynamics' and for all the valuable inputs he has given me. I would like to thank the research scholars of my research group, Nishant Kolli and Chelli Sai Manohar for their support. I thank all my classmates and the research scholars of the Department of Chemistry for having accompanied me in my journey as a master's student. I also take this opportunity to thank everybody who has helped me accomplish this dissertation work.

ABSTRACT

We theoretically study the dynamics photo-induced exciton migration in linear conjugated polymeric chains. Considering coupling between the local torsional modes and electronic degrees of freedom at each polymer subunit (rotors), we propose a model to understand the effect of conjugation breaks caused by the torsional motion within the backbone of the polymer chain. As a first step, we study the diffusion of the exciton in ordered chains and derive an analytic expression for its time-averaged transition probability as function of chain length of the polymer. Further, we have investigated ordered chains with a single impurity to understand its impact on end-to-end exciton migration. Using configurational averaging over a large sample space, we calculate the end-to-end transition probability for systems with static torsional disorder. We propose a conjecture that the time-averaged transition probability for end-to-end exciton migration is always greater in cased of ordered systems than in case of any disordered system. Finally, we propose a scheme based on exact diagonalisation method to numerically solve the complete Hamiltonian proposed in our model, to understand the exciton migration dynamics of polymer chains with dynamic torsional disorder.

Keywords: *Exciton migration, Torsional disorder, Linear Conjugated Polymers, Exact Diagonalisation, Configurational averaging*

Contents

List of Figures

"Nobody ever figures out what life is all about, and it doesn't matter. Explore the world. Nearly everything is really interesting if you go into it deeply enough."
-Richard Feynman

1 Introduction

An Exciton is a bound state of an excited electron that is electrostatically held by a hole. The term 'Exciton' was first coined by Yakov Frenkel to describe a quasiparticle, an excitation wave packet formed by the superposition of excitation waves [1]. An important property of these excitons is their ability to move *i.e.* the transfer of excitation energy from one site to another [2]. Exciton dynamics or the dynamics of the excitation energy transfer plays a crucial role in the photoresponse of molecular systems like the light-harvesting complexes in photosynthetic species, organic semiconductors, and molecular aggregates. The performance of organic solar cells, OLEDs (organic light emitting diodes) and sensors is dependent on the efficiency of exciton transport. Many aspects of this process of energy transfer or transport mechanism remain elusive and have been the subject of many studies.

Conventionally, the transfer of excitation energy from a donor to an acceptor has been described perturbatively in terms of Förster's resonance energy transfer (FRET) mechanism [3]. While such a theoretical framework disregards the role of quantum coherence, experimental investigations in the last

decade have provided evidence for a wave-like energy transfer mechanism that operates in ultrashort timescales [4]. These quantum effects were first observed in the case of FMO (Fenna-Mathew-Olson) complex, a photosynthetic light harvesting complex found in the green sulfur bacteria. The near quantum efficiency of energy conversion has been attributed to the existence of long-lived coherence in these molecules. Interestingly, similar results were reported in case of conjugated polymers, where the intramolecular exciton migration was found to occur in a coherent fashion [5]. These findings that excitons could surf along the chain of a conjugated polymer preserving coherence at ambient temperature indicate the presence of certain structural features that preserve these quantum effects. Certainly, a detailed study to identify these aspects of structure and bonding would not only help us in improving the efficiency of organic semiconductor devices but also design novel functional materials. In the following sections, we delve on some of the ideas about excitation energy transfer to get insights to understand the origin of long lived coherence in molecular systems.

1.1 Regimes of Excitation Energy transfer

The actual mechanism of excitation energy transfer in molecular systems is determined by the relative time scales at which intramolecular vibrational relaxation (τ_{rel}) and transfer of excitation(τ_{trans}) occur. τ_{rel} is the time taken for the intramolecular vibrations to completely relax to thermal equilibrium after the electronic transition and τ_{trans} is the time taken for movement of excitation energy from the donor molecule to the acceptor molecule [6]. These factors distinguish the excitation energy transfer into three regimes. This classification and their brief description in the following sections

(I) Incoherent Transfer (FRET) where $\tau_{rel} \ll \tau_{trans}$

(II) Coherent Transfer, where $\tau_{rel} \gg \tau_{trans}$

(III) Intermediate Coupling Regime (Partial Coherent exciton transfer).

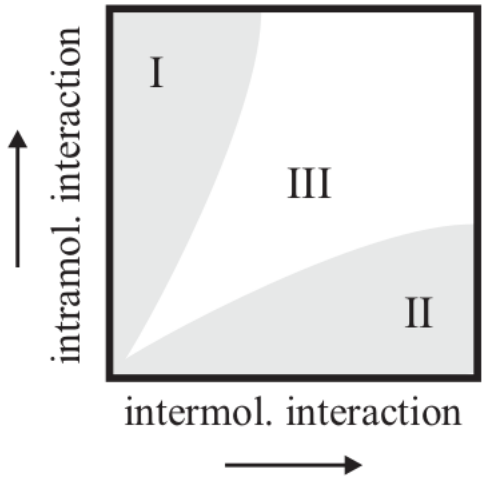


Figure 1.1: Different Regimes of Excitation Energy Transfer(EET)
Image taken from: May, V., and Kühn, O. (2008)

Förster Resonance Energy Transfer

The classical theory of Förster Resonance Energy Transfer (FRET) considers weak coupling between electronic transitions of donor-acceptor chromophores. These interactions are treated perturbatively and the spectral overlap serves as a necessary condition to preserve the conservation of energy. Accordingly, the rate constant of energy transfer (k) is inversely proportional to sixth power of the inter-chromophore separation (R).

$$k = \frac{1}{\tau_D}\left(\frac{R_o}{R}\right)^6 \tag{1.1}$$

Where τ_D is the lifetime of the donors' excited state [1] and R_o is the Förster radius. This distance dependence was experimentally studied by Stryer and Haugland in macromolecules, who proposed that under suitable conditions the energy transfer process could be used as an optical ruler to measure the distance between chromophores that are separated by 10 - 60Å[7].

[1] Here τ_D is same as the relaxation time(τ_{rel})

Over the decades, following the initial work done by Förster, the theory underpinning the mechanisms of energy transfer have been subject to many changes. The deviations from the distance dependence of $\frac{1}{R^6}$ are seen, as the dipole approximation fails at very close and very large distances of separation. Inherently, Förster theory needs complete relaxation of the donor for the transfer of excitation energy to occur *i.e.* the intra-molecular relaxation time (τ_{rel}) is much less than transition time (τ_{trans}). The rapid intra-molecular relaxation of the donor species gives rise to dephasing and leads to *incoherent transfer* of excitation energy and migration is occurs in a random walk fashion. Even as quantum chemical validations of this model have failed, it remains to be most suitable as a first approximation method. So far, the widely accepted notion has been that exciton migration in conjugated polymers occurs through a hopping mechanism across a collection of spectroscopic units (chromophores).

Coherent Transport

When $\tau_{rel} \gg \tau_{trans}$, the movement of the exciton would be governed by the time-dependent Schrödinger equation with the corresponding Hamiltonian describing the system.

$$i\hbar\frac{\partial\Psi(x,t)}{\partial t} = \hat{H}_S\Psi(x,t) \tag{1.2}$$

The exciton travels through the molecular system as wave packet and such motion would require a fixed phase relationship between excited state wave function of different molecular species (Chromophores)[2] participating in the transport process. Therefore, it is called a *coherent transfer* and it is the typical type of motion observed in closed quantum system where there are no environmental factors that can causing dephasing of the system. The density matrix is either formulated in local site basis (say, $\{|\alpha_i\rangle's\}$) or in the

[2]Chromophores or the spectroscopic sub-units mean the same in our discussion here, but the definition of the same in case of conjugated polymers is vague and could depend various factors such as conformational dynamics that introduce conjugation breaks.

excitonic basis (say, $\{|n_i\rangle's\}$) to theoretically describe exciton dynamics of such systems.

Partial Coherent Transport

When $\tau_{rel} \approx \tau_{trans}$, it is not possible to strictly comment on the mechanism of exciton transport. As there could be a possibility of strong coupling between excitonic states within the molecule (intra molecular) while it could still occur in an incoherent fashion across molecules. In general, it is possible to have transport occurring in diverse ensembles through different types of motion.Therefore the transport occuring in this regime is considered as, '*Partially Coherent*'. In all these cases the exact definition of a spectroscopic unit or a chromophore remains vague and depend various other factors. But, both in the case of coherent and partial coherent transport, 'coherence' enhances the process. In the following we discuss about the different types of coherence and also present a brief argument to distinguish them with each other.

1.2 What is Quantum coherence?

Coherence is a correlation between any two random variables, say ($\mathbf{X}\,\mathbf{Y}$). It is given by:

$$\rho_{XY} = \frac{F_{XY}}{\sqrt{F_{XX}F_{YY}}} \tag{1.3}$$

Where F_{XX} , F_{YY} are the expectation values of the probability densities of X and Y. $F_{XY} = \int xy^* f_{XY}(x,y)dx\ dy$ is the expectation value of the joint probability density f_{XY} [8]. Quantum coherence could be defined as a correlation of waves or wave-like entities at the mesoscopic level. Coherence in this regime could also be classical in nature. It is important note that in chemical dynamics some of the coherence effects could actually be described accurately using classical mechanics. Experimentally it is quite difficult to distinguish

between classical and quantum coherences. Particularly, this is the case with electronic processes that involve certain non-adiabaticity. An elegant argument was presented by Miller to distinguish such cases using semiclassical methods in reference [9]. Further, these effects could be associated with the states of the system or the process itself.

State Coherence

The quantum state is said to be pure if it can be represented by a density matrix, $\rho = |\psi\rangle\langle\psi|$. $Tr(\rho^2)$ is a measure of purity of a state, it tells us on how close is a given state to a pure state. It is independent of the basis $\{|\psi_i\rangle\}$. Coherences appear as off-diagonal elements in the density matrix and are dependent on the basis chosen to construct the density matrix(ρ). For the case excitonic systems that are of interest to us, two bases are quite useful in describing the system. They are the localised site basis and the excitonic basis or the energy basis which is the eigen basis of the given Hamiltonian. As the coupling terms that appear as off-diagonal elements for the density matrices in these two basis do not coincide, the coherence in each case are different. This kind of coherence is described as *'state coherence'* [10].

Process Coherence

Based on the extent to which an open system evolves unitarily with time, a process could be described as coherent process or an incoherent process [6, 10]. Unitary evolution dominates if the individual sites in a multichromophoric system are strongly coupled to each other when compared to the environment. In Förster's mechanism the individual donor and acceptor molecules are strongly coupled to their own dissipative environments than to each other. The randomness introduced by the environment or any other mode leads to loss of coherence, and dephases the system. Disorder in various forms is the dominant origin for inhomogeneous spectral broadening in molecular systems and could also lead to quantum mechanical motion in

materials. Therefore, it is necessary to understand disorder effects to define the nature of transport in molecular systems.

1.3 Excitons Surf along Conjugated Polymer Chains

In an article with quite a rhetoric title, J.L. Brédas and Robert Silbey shared their perspective on the implications of excitons being able to preserve their phase over a long-range. As outlined by the authors, these results have followed a series of investigations that were dedicated to the study multi-chromphoric systems. It appears that chromophores in such systems communicate with each other through long range coulombic interactions and also through a bath of fluctuating nuclear motions within their molecular architecture and the environment. The evidence for long-lived electronic quantum coherence of excitation energy transfer in light harvesting complexes like the Fenna-Mathew-Olson (FMO) complex have attracted a lot of attention. This is due to the fact that coherence has been found to be preserved in a highly disordered medium. Greg Scholes *et al* have reported similar quantum coherent dynamics in the case of intrachain exciton migration in conjugated polymers at room temperature[11]. Using two-time anisotropy decay (TTAD) experiments, MEH-PPV systems have been studied in two regimes: **(i)** In solution phase, dissolved in good solvent and **(ii)** Aqueous suspensions in water, where these polymers could aggregate to form nanoparticles. It was observed that coherent transport was observed only in the case of solution samples of the polymer, where the polymer molecules exist more freely as open chains.

The striking difference in the photoresponse of the signals could be attributed to excitation energy transfer being the dominant mechanism. Further, 2D photon echo experiments confirmed the long-lived coherence in extended chains of polymers rather than in polymer aggregates [11]. This indicates

the presence of some structural features of these open polymeric chains that induce correlations in the fluctuations of energy gaps. While it was surprising that such features could induce coherences in highly disordered systems like polymers, the true nature of such coherences if understood could have a wide range of implications from building solar cells with higher quantum efficiency to better quantum information storage devices. Therefore, it would be interesting to study the exact nature of intra-chain exciton migration in linear conjugated molecules as a function of their geometry to elucidate the mystery behind the coherent excitation energy transfer.

"We can only see a short distance ahead, but we can see plenty there that needs to be done."
Sir Alan Turing

2

Literature Survey

In organic semiconductors, the delocalisation of π electrons is the key feature that defines their optical and electrical properties. There exists a strong relationship between their electronic structure and molecular geometry. For example, conformational fluctuations and molecular vibrations modulate the delocalisation of the π electrons. In condensed phase, these factors significantly affect the properties of conjugates polymers. Recently, there has been a lot interest to understand the role of torsional motion in exciton transport in many conjugated polymers such as polyfluorenes, P3HT [1], MEH-PPV[2] and other phenylene type polymers. [12–14]. Excitons in these systems are low lying excited states of a spectroscopic unit in the polymer. The definition of a chromophore or a spectroscopic unit in these kind of molecular systems with extended conjugation has remained elusive [15].

[1] Poly(3-hexylthiophene-2,5-diyl)
[2] Poly[2-methoxy-5-(2-ethylhexyloxy)-1,4-phenylenevinylene]

Definition of Chromophores

Spectroscopic units or chromophores are generally regarded as the irreducible parts of a polymer, that can absorb and emit light [16, 17]. More often, they have been defined arbitrarily, based on the minimum threshold in the overlap of the π molecular orbitals. Computational studies have suggested that there could alos be a possibility for super exchange occurring in exciton transport along with the dipole-dipole coupling [18]. It was shown that the chromophore distributions obtained from the assumption of random conjugation breaks can not explain the spectral properties of polymers in condensed phase [14]. Therefore, the criterion of conjugation break does not sufficient to define a chromophore or a spectral unit.

An alternative proposed by Beenken *et al* defined a chromophore as the spatial extent of an exciton-polaron formed by the coupling of nuclear coordinates with electronic degrees of freedom [15]. This was rejected by Barford and Tozer in their work on the fully quantized Frenkel-Holstein model. They claim that that the self localised solutions of Born-Oppenheimer Hamiltonians do not occur in the case of conjugated polymers. There has been an attempt to provide a rigorous definition for the same by Barford *et al* in their trilogy of papers on theory of linear optical transitions in conjugated polymers [19–21]. They have defined a spectroscopic unit to be the average spatial extent to which the exciton could get delocalised. As it is evident, the definition of the fundamental unit that participates in the exciton transfer process is crucial to explain the underlying mechanism and further studies will have to provide a clear distinction of their definition and its validation. In our study we neglect the possibility of long range dipole-dipole coupling through space and thus conjugation would play a crucial role. This defines a clear distinction between chromophores in our case. In the following section we present a brief discussion of various theoretical models that have been used to study exciton transport.

Theoretical Models for Exciton Transport

In case of conjugated polymers, the movement of exciton from one site to another could occur through super exchange via the chemical bonds and through space by dipole-dipole coupling of any two chromophores. Many models have been proposed to understand the optical properties of conjugated polymers under various limits. Within the Born-Oppenheimer approximation, the Hamiltonian could separated into nuclear part and the electronic part. In these models Born-Oppenheimer Hamiltonians parameterized for particular nuclear coordinates, describing the state π-electrons are used. These π-electron models include Hückel model, Su-Schrieffer-Heeger (SSH) model, Peierls model and the Pariser-Parr-Pople(P-P-P) model. The Hückel model describes a system of non-interacting π-electrons with a static geometry and the PPP model describes a system interacting electrons

$$\hat{H} = \underbrace{-\sum_{n=1,\sigma} t_n \left(\hat{a}^{\dagger}_{n,\sigma}\hat{a}_{n+1,\sigma} + \hat{a}^{\dagger}_{n+1,\sigma}\hat{a}_{n,\sigma}\right)}_{H\ddot{u}ckel\ \ Hamiltonian} + U\sum_{n}(N_{n,\uparrow} - \frac{1}{2})(N_{n,\downarrow} - \frac{1}{2}) + \frac{1}{2}\sum_{n\neq m} V_{n,m}(N_n - 1)(N_m - 1) \tag{2.1}$$

Where $-t_n \left(\hat{a}^{\dagger}_{n,\sigma}\hat{a}_{n+1,\sigma} + \hat{a}^{\dagger}_{n+1,\sigma}\hat{a}_{n,\sigma}\right)$ represents the transfer between spin orbitals $\chi_n(r,\sigma)$ and $\chi_{n+1}(r,\sigma)$ of energy $-t_n$. Here the $UN_{n,\uparrow}N_{n,\downarrow}$ and $V_{n,m}N_nN_m$ represent the coulombic interaction between two electrons of the same spatial orbital and the orbitals ϕ_n, ϕ_m respectively. The Schrieffer-Heeger (SSH) model describes the dynamics of a system of non-interacting electrons. We have the SSH hamiltonian to be defined as,

$$\hat{H}_{SSH} = \hat{H}_e + \hat{H}_{n-n} + \hat{H}_{e-n} \tag{2.2}$$

where $\hat{H}_e$ is the Hückel Hamiltonian,

$$\hat{H}_{n-n} = \sum_n \left(\frac{P_{n,x}^2}{2M} + \frac{K_x}{2}(u_{n+1,x} - u_{n,x})^2 + K_x \delta r (u_{n+1,x} - u_{n,x}) \right) \tag{2.3}$$

and

$$\hat{H}_{e-n} = -\sum_{n,\sigma} \alpha_x \left(u_{n+1,x} - u_{n,x}\right) \left(\hat{a}_{n,\sigma}^\dagger \hat{a}_{n+1,\sigma} + \hat{a}_{n+1,\sigma}^\dagger \hat{a}_{n,\sigma} \right) \tag{2.4}$$

Where $u_{n,x}$, K_x and α_x are the projection parameters [3]. In the static limit the SSH model reduces to give the Peierls model where we have only the contributions from the kinetic and elastic parts of the Hamiltonian.

$$\hat{H}_{Peierls} = \hat{H}_{kinetic} + H_{elastic}, \tag{2.5}$$

$$\hat{H}_{kinetic} = -2 \sum_n t_n \hat{T}_n \tag{2.6}$$

$$\text{where } \hat{T}_n = \frac{1}{2} \sum_\sigma \left(\hat{a}_{n,\sigma}^\dagger \hat{a}_{n+1,\sigma} + \hat{a}_{n+1,\sigma}^\dagger \hat{a}_{n,\sigma} \right) \tag{2.7}$$

$$\hat{H}_{elastic} = \frac{1}{4\pi t \lambda} \sum_n \Delta_n^2 + \Gamma \sum_n \Delta_n \tag{2.8}$$

Here t_n is the hybridization integral, $t_n = t + \frac{\Delta_n}{2}$ where $\Delta_n = -2\alpha(u_{n+1} - u_n)$. α is the electron-phonon coupling parameter. The positive and negative values of Δ_n corresponds to the reduction and extension of bond lengths.

Of all these the Frenkel-Holstein model has been widely used for to describe the optical properties of conjugated polymers. As mentioned earlier, the frenkel excitons could possibly delocalise over many monomers and it could be described by the Frenkel model. The Holstein model gives the description of the coupling between individual sites. Together the Frenkel-Holstein model

[3] u_i is the displacement of the i^{th} ion from its reference position and K is the spring constant of the oscillator

explains the motion of a delocalised exciton coupled to a normal mode [22]. The Frenkel-Holstein Hamiltonian is defined as:

$$\hat{H}_{FH} = \underbrace{\sum_n \varepsilon_n \hat{a}_n^\dagger \hat{a}_n + \sum_{i,j} V_{ij}(\hat{a}_i^\dagger \hat{a}_j)}_{Frenkel\ \ Hamiltonian} \underbrace{- t \sum_n Q_n \hat{a}_n^\dagger \hat{a}_n + \frac{K}{2}\sum_n Q_n^2 + \frac{M}{2}\sum_n P_n^2}_{Holstein\ \ Hamiltonian}$$

Where ε_n is the excitation energy of the Frenkel exciton on the unit n and $\hat{a}_n, \hat{a}_n^\dagger$ are the fermionic creation, annihilation operators written in second quantisation[4]. V_{ij} is the exciton transfer integral, it accounts for both super exchange and dipole-dipole coupling. t is the coupling parameter that quantifies the strength of coupling between the electronic degrees of freedom and nuclear co-ordinates (*i.e.* the exciton and the normal mode at a particular site). The terms P_n^2 and Q_n^2 terms represents the kinetic energy and elastic energy of the harmonic oscillators (associated with the local normal modes) respectively. K is the spring constant and M is the mass associated with the oscillator. Barford *et al* have used the same model within the Born-Oppenheimer limit for their study of linear conjugated polymers [19]. In these studies the local normal modes where considered to be coupled to the excitons formed at the site. In a similar way, local torsional modes being coupled to the electronics degrees of freedom could be a possible area for exploring theoretical models that could explain disorder in these systems.

Torsional dynamics

Torsional relaxation could lead to formation of relaxed excitons with a binding energy of about $0.5eV$ [5]. This high binding energy compared to inorganic semiconductors(few milli eV) is due to two factors: **(i)** Weak screening of charges or higher effective nuclear charge on the outermost electrons. **(ii)** The presence of geometrical relaxation factors that yield stable excitons. Experimental findings have confirmed the possibility of inter-

[4]For a pedagogic discussion on second quantization we refer the reader to the arguments presented in Chapter 1 of Mahan, G. D. (2013); Many-particle physics.

conversion of electronic excitation energy into kinetic energy that lead to planarization of conjugated molecules [12].This torsional relaxation process was found to occur at a sub-100 fs timescale due to efficient redistribution of the electronic excitation energy into torsional modes through non-adiabatic transitions. These relaxation effects have been conjectured to interfere in the exciton transport process. Yet, the efficient interconversion leading to dumping energy into momentum states could dominate over the dampening effects of the environment leading to coherent exciton transport. Beenken *et al* have shown that torsional dynamics within the polymer backbone has a significant influence on its spectral properties of phenylene type polymers. Torsional broadening has a major contribution to the total inhomogeneous broadening of the $1^1A \rightarrow 1^1B$ transition in biphenyl and bithiophene molecule [23, 24]. Torsional disorder could be the reason for such broadening possibly due to the defects that break the conjugated chain into different spectroscopic units and the dependency of transition energy on torsional angles.

As discussed earlier, though conjugation breaks can not completely decided the extent of localisation of the exciton, they do play an important role. While conventional ensemble measurements have been confined to single-molecule spectroscopy and transient electronic absorption spectroscopy, recent ultrafast spectroscopic techniques have been able to probe molecular events occurring on femtosecond timescales. These studies have outlined the importance of conformational motion within the polymer chain and it has been conjectured that torsional motion on such timescales could guide coherent exciton transport with the polymeric chain in condensed phase systems [12, 13]. While the existing theories have to reconcile with many aspects relating to exitonic delocalisation, exciton trapping, exciton-phonon correlations, and exciton-exciton annihilation, we find that a key aspect of future studies should be centered around torsional effects and their undeniable link with excitonic motion.

In this regard, we propose that a quantum dynamical study incorporating the aspects of coupling between local torsional modes and exctions could provide valuable insights to guide further experimental studies. While emerging evidence from experimental studies proves a definite role of torsional effects in conjugated polymers, we find that there has been a lot of speculation on the theoretical models that have been proposed to explain them. Against this background, we find the need for a unified model that can treat both static and dynamic torsional disorder effects on an equal footing. In the following, chapters of this dissertation we present a new theoretical model and provide some of our initial results.

"Our imagination is stretched to the utmost, not, as in fiction, to imagine things which are not really there, but just to comprehend those things which 'are' there."
Richard Feynman

3

Theoretical Framework

In this chapter we describe the basic theoretical frame work that has been adopted for studying the torsional effects on exciton migration rates in linear conjugated polymer chains. To model these effects, we consider the coupling between the electronic and torsional degrees of freedom of the polymer subunits. For the proposed model, we derive the various realizations of the Hamiltonian under different limits.

3.1 The Model

We consider a polymer chain with $\mathcal{N}(\mathcal{N} \geq 2)$ identical subunits that are optically active two-level systems. It is assumed that there is a significant torsional motion about the bonds that connect these subunits. To describe the same, we use a symbolic example of Poly(p-phenylene) system. Each benzene ring is considered as a subunit of the polymer chain or as a site of the 1-Dimensional lattice formed by the polymer chain. It is assumed that the benzene ring to be a rotor that can rotate about its molecular axis. As mentioned earlier, we neglect the possibility of exciton transport through space and only consider super exchange that occurs through bonds.

Thus conjugation break in our case to a great extent defines the spatial delocalisation of the exciton. The Hamiltonian of the system is considered to be an operator in the Hilbert space $\mathcal{H}_S = \mathcal{H}_\varepsilon \otimes \mathcal{H}_{\theta_1} \otimes \mathcal{H}_{\theta_2} \otimes .....\otimes \mathcal{H}_{\theta_N}$, a tensor product space of the localised exciton space and the 'N' rotor spaces.For simplicity, we assumed that exciton energies(ε) on all sites are same and moment of inertia(I) of all rotors are same.

$$\hat{H}_S = \sum_{i=1}^{N} \epsilon \sigma_+(i)\sigma_-(i) - \frac{\hbar^2}{2I}\sum_{i=1}^{N}\frac{\partial^2}{\partial\theta_i^2} + \sum_{i,j=1}^{N} U_{ij} + \sum_{i,j=1}^{N} V_{ij}\sigma_+(i)\sigma_-(j) \quad (3.1)$$

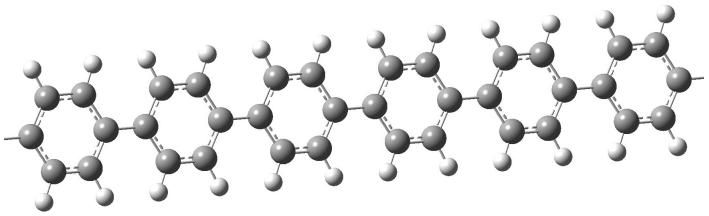


Figure 3.1: Symbolic example of Poly(para-phenylene) systems

Analogous to the Heisenberg spin chain system, we define our Hamiltonian using the operators $\sigma_+(i)$ and $\sigma_-(i)$ as the ladder operators that operate on the i^{th} site($i = 1, 2, \cdots, N$), written in the form of pauli spin operators σ^x, σ^y and σ^z. Here $|\uparrow\rangle$ and $|\downarrow\rangle$ indicate the excited state and ground state of the two level system respectively.[1]

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \text{ and } \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\sigma^+|\downarrow\rangle = |\uparrow\rangle \text{ and } \sigma^-|\uparrow\rangle = |\downarrow\rangle$$

[1] $|\uparrow\rangle$ and $|\downarrow\rangle$ are used in an analogous fashion and are not be confused with the nature of spin of the excited electron. In our discussion we do not describe the nature of the exciton that is formed *i.e.* whether it is a singlet or triplet exciton.

where $\sigma^+ = \frac{1}{2}\left[\sigma^x + i\sigma^y\right] = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ and $\sigma^- = \frac{1}{2}\left[\sigma^x - i\sigma^y\right] = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$

The first term corresponds to the Hamiltonian for localized excitons, the second term corresponds to Hamiltonian for free rotors, third term corresponds to coupling between rotors and the last term corresponds to the coupling between localized excitons and the rotors (This is the term responsible for exciton transfer). The coupling between any two sites is assumed to depend only on the difference of rotor angles coupling these sites, further it is assumed of the form $V_{ij}(\theta_i - \theta_j) = 2v_{ij}\cos(\theta_i - \theta_j)$ and similarly $U_{ij}(\theta_i - \theta_j) = 2v_{ij}\cos(\theta_i - \theta_j)$.

$$\begin{aligned}\hat{H}_S =& \hbar\omega_\varepsilon \sum_{i=1}^{N} \sigma_i^+\sigma_i^- - \hbar\omega_\theta \sum_{i=1}^{N} \frac{\partial^2}{\partial\theta_i^2} + \sum_{\substack{i=1 \\ i>j}}^{N}\sum_{j=1}^{N} 2u_{ij}\cos(2(\theta_i - \theta_j)) \\ &+ \sum_{\substack{i=1 \\ i\neq j}}^{N}\sum_{j=1}^{N} 2v_{ij}\cos(\theta_i - \theta_j)\sigma_i^+\sigma_j^- \end{aligned} \tag{3.2}$$

where $\omega_\theta = \frac{\hbar}{2I}$ and $\hbar\omega_\varepsilon = \epsilon$.
As excitation number operator $\hat{N}_{ex} = \sum_{i=1}^{N} \sigma_+(i)\sigma_-(i)$ commutes with the Hamiltonian given in equation (3.2), excitation number is a good quantum number. Additionally, we find that the Hamiltonian also commutes with the total momentum operator $\hat{P}_{tot} = -i\hbar\sum_i^N \frac{\partial}{\partial\theta_i}$ [2].Thus, we restrict ourselves to single excitation manifold. The orthonormal basis set spanning this space is given by:

$$\{|i\rangle := \underset{\uparrow \text{ at } i^{\text{th}} \text{ position of a polymer chain of size } N}{|\downarrow, \cdots, \uparrow, \cdots, \downarrow\rangle} \quad | \, i \in \{1, \cdots, N\}\}$$

[2]We present these commutation relations in the Appendix-C

An arbitrary wave function in this composite system $\mathcal{H}_S$ at time t can be written as,

$$
\begin{aligned}
|\Psi(\theta_1, \theta_2, ...\theta_N, t)\rangle &= \sum_{n=1}^{N} \psi_n(\theta_1, \theta_2, ...\theta_N, t)|n\rangle \\
|\Psi[\Theta; t]\rangle &= \sum_{n=1}^{N} \psi_n[\Theta; t]|n\rangle
\end{aligned} \tag{3.3}
$$

where $\Theta = (\theta_1, \cdots, \theta_N) \in [0, 2\pi]^N$ and all ψ_i's are periodic w.r.t each θ_i with the period 2π. Plugging in $|\Psi[\Theta, t]\rangle$ and $\hat{H}_S$ from equation 3.2, we into the Time-dependent Schrödinger equation as

$$
\begin{aligned}
i\hbar\frac{\partial\Psi[\Theta,t]\rangle}{\partial t} =& -\ \hbar\omega_\theta \sum_{i=1}^{N} \frac{\partial^2}{\partial\theta_i^2} \sum_{n=1}^{N} \psi_n[\Theta,t]|n\rangle + \sum_{\substack{i,j=1 \\ i>j}}^{N} U_{ij} \sum_{n=1}^{N} \psi_n[\Theta,t]|n\rangle \\
&+ \hbar\omega_\varepsilon \sum_{i=1}^{N} \sigma_i^+\sigma_i^- \sum_{n=1}^{N} \psi_n[\Theta,t]|n\rangle + \sum_{\substack{i,j=1 \\ i\neq j}}^{N} V_{ij}\sigma_i^+\sigma_j^- \sum_{n=1}^{N} \psi_n[\Theta,t]|n\rangle
\end{aligned}
$$

$$
\begin{aligned}
i\hbar\frac{\partial\Psi[\Theta,t]\rangle}{\partial t} =& -\hbar\omega_\theta \sum_{i,n=1}^{N} \frac{\partial^2\psi_n[\Theta,t]}{\partial\theta_i^2}|n\rangle + \sum_{\substack{i,j=1 \\ i>j}}^{N} \sum_{n=1}^{N} U_{ij}\psi_n[\Theta,t]|n\rangle \\
&+ \hbar\omega_\varepsilon \sum_{n=1}^{N} \psi_n[\Theta,t] \sum_{i=1}^{N} \sigma_i^+\sigma_i^-|n\rangle + \sum_{n=1}^{N} \psi_n[\Theta,t] \sum_{\substack{i,j=1 \\ i\neq j}}^{N} V_{ij}\sigma_i^+\sigma_j^-|n\rangle
\end{aligned}
$$

Therefore, the Time-dependent Schrödinger equation(TDSE) takes the form,

$$
\begin{aligned}
i\hbar\frac{\partial|\Psi\rangle}{\partial t} =& -\hbar\omega_\theta \sum_{i,n=1}^{N} \frac{\partial^2\psi_n[\Theta,t]}{\partial\theta_i^2}|n\rangle + \sum_{\substack{i,j=1 \\ i>j}}^{N} \sum_{n=1}^{N} U_{ij}\psi_n[\Theta,t]|n\rangle + \hbar\omega_\varepsilon \sum_{n=1}^{N} \psi_n[\Theta,t]|n\rangle \\
&+ \sum_{\substack{n=1 \\ i\neq n}}^{N} \sum_{i=1}^{N} V_{in}\psi_i[\Theta,t]|n\rangle
\end{aligned} \tag{3.4}
$$

Considering equation 3.4 , the matrix form of the TDSE would be,

$$i\hbar\frac{\partial}{\partial t}\Psi_{N\times 1} = \left[-\hbar\omega_\theta \sum_{i=1}^{N}\frac{\partial^2}{\partial\theta_i^2} + \sum_{\substack{i,j=1\\ i>j}}^{N} U_{ij} + \hbar\omega_\varepsilon\right]\mathbb{I}_{N\times N}\Psi_{N\times 1} + \mathbb{V}^{in}_{N\times N}\Psi_{N\times 1} \tag{3.5}$$

where

$$\mathbb{V}^{in}_{N\times N} = \begin{bmatrix} 0 & V_{12} & \cdots & V_{1N} \\ V_{21} & 0 & \cdots & V_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ V_{N1} & V_{N2} & \cdots & 0 \end{bmatrix} \text{ and } \Psi_{N\times 1} = \begin{bmatrix} \psi_1[\Theta,t] \\ \psi_2[\Theta,t] \\ \vdots \\ \psi_n[\Theta,t] \end{bmatrix}$$

Therefore, the Hamiltonian matrix is given by:

$$H_S = \left[-\hbar\omega_\theta \sum_{i=1}^{N}\frac{\partial^2}{\partial\theta_i^2} + \sum_{\substack{i,j=1\\ i>j}}^{N} U_{ij} + \hbar\omega_\varepsilon\right]\mathbb{I}_{N\times N} + \mathbb{V}^{in}_{N\times N}. \tag{3.6}$$

3.2 Various realizations of $\hat{H}_S$

Different forms of the potential matrix V_{in} represent model different physical situations. First is a linear chain, where there are nearest neighbor interactions. Second case is a nanohoop with circular geometry with a matrix similar to that of a Huckel type matrix. Finally, a dense matrix that represents a closely packed geometry that could model molecular aggregate type of interactions.The matrix $\mathbb{V}^{in}_{N\times N}$ for the different arrangements of the polymer chain such as linear, cyclic and closely packed geometries are illustrated in the following table.

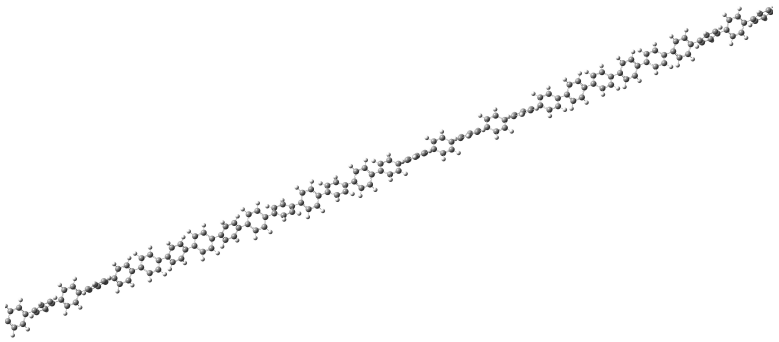


Figure 3.2: Linear Geometry

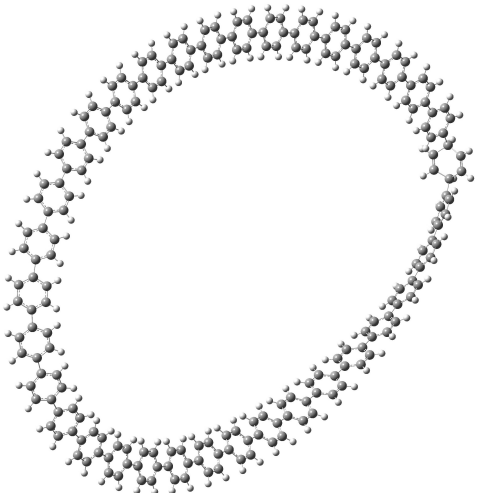


Figure 3.3: Cyclic Geometry (Nano-hoop)

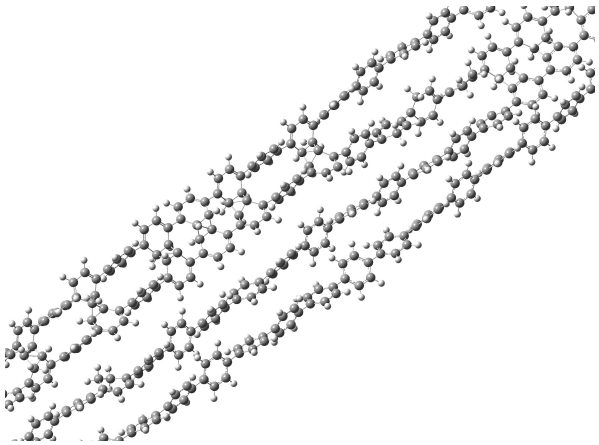


Figure 3.4: Close-packed case

Models of Arrangements

Model	**Matrix** $\mathbb{V}^{in}_{N\times N}$
Linear	$\begin{bmatrix} 0 & V_{12} & 0 & \cdots & \cdots & 0 \\ V_{21} & 0 & V_{23} & \cdots & \cdots & 0 \\ 0 & V_{32} & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & V_{(N-1)N} \\ 0 & 0 & \cdots & \cdots & V_{N(N-1)} & 0 \end{bmatrix}$
Cyclic	$\begin{bmatrix} 0 & V_{12} & 0 & \cdots & \cdots & V_{1N} \\ V_{21} & 0 & V_{23} & \cdots & \cdots & 0 \\ 0 & V_{32} & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & V_{N(N-1)} \\ V_{N1} & 0 & \cdots & \cdots & V_{N(N-1)} & 0 \end{bmatrix}$
Close Packing	$\begin{bmatrix} 0 & V_{12} & V_{13} & \cdots & V_{1N} \\ V_{21} & 0 & V_{23} & \cdots & V_{2N} \\ V_{31} & V_{32} & 0 & \cdots & V_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ V_{N1} & V_{N2} & V_{N3} & \cdots & 0 \end{bmatrix}$

3.3 Fourier Transform of the Hamiltonian $\hat{H}_S$

The general form of the Hamiltonian presented in the earlier section generates a set of coupled parabolic differential equations that do not seem to have an exact solution.

$$i\hbar\frac{\partial}{\partial t}\sum_{n=1}^{N}\psi_n[\Theta,t]|n\rangle = \underbrace{-\hbar\omega_\theta\sum_{i=1}^{N}\sum_{n=1}^{N}\frac{\partial^2\psi_n[\Theta,t]}{\partial\theta_i^2}\,|n\rangle}_{\mathbf{I}} + \underbrace{\sum_{\substack{i,j=1\\ i>j}}^{N}\sum_{n=1}^{N}U_{ij}\psi_n[\Theta,t]\,|n\rangle}_{\mathbf{II}}$$
$$+\underbrace{\sum_{\substack{n=1\\ i\neq n}}^{N}\sum_{i=1}^{N}V_{in}\psi_i[\Theta,t]\,|n\rangle}_{\mathbf{III}} + \underbrace{\hbar\omega_\varepsilon\sum_{n=1}^{N}\psi_n[\Theta,t]\,|n\rangle}_{\mathbf{IV}} \tag{3.7}$$

Now using the fact that $\psi_n[\Theta;t]$'s are periodic functions with respect to each θ_i period 2π, we can use Fourier series representation for writing ψ_n's as :

$$\psi_n[\Theta;t] = \sum_{K\in\mathbb{Z}^N}\tilde{\psi}_n[K;t]e^{iK\cdot\Theta}, \text{where } K=(k_1,\cdots,k_N)\in\mathbb{Z}^N \tag{3.8}$$

The time-dependent Schrödinger on in the momentum space is as follows:

$$\begin{aligned}i\hbar\frac{\partial}{\partial t}\sum_{n=1}^{N}\tilde{\psi}_n[K^{'},t]|n\rangle &= -\hbar\omega_\theta\sum_{n=1}^{N}\sum_{i=1}^{N}k_i^{'2}\tilde{\psi}_n[K^{'},t]|n\rangle + \hbar\omega_\varepsilon\sum_{n=1}^{N}\tilde{\psi}_n[K^{'},t]|n\rangle\\ &+ u_{ij}\sum_{\substack{i,j=1\\ i>j}}^{N}\sum_{n=1}^{N}\Big(\tilde{\psi}_n[K^{'}(j^-,i^+),t]+\tilde{\psi}_n[K^{'}(j^+,i^-),t]\Big)\,|n\rangle\\ &+ v_{ij}\sum_{\substack{i=1\\ i\neq n}}^{N}\sum_{n=1}^{N}\Big(\tilde{\psi}_i[K^{'}(n^-,i^+),t]+\tilde{\psi}_i[K^{'}(n^+,i^-),t]\Big)\,|n\rangle\end{aligned} \tag{3.9}$$

where all $k_i^{'}s\in\mathbb{Z}$ i.e. $k_i^{'}\in(-\infty,\infty)$ and $K^{'}(x^+,y^-)=(k_1^{'},k_2^{'},...k_x^{+^{'}}...k_y^{+^{'}}...k_N^{'})$

"The laws of nature are constructed in such a way as to make the universe as interesting as possible."
Freeman Dyson

4

Effect of Torsional Disorder on Exciton Migration

In this chapter we derive various realizations of the general Hamiltonian ($\hat{H}_S$) of the model proposed in the last chapter. At different limits of the parameters : $\omega_\theta, \omega_\varepsilon, V_{ij}$ and U_{ij}, the resulting Hamiltonian would correspond to a different physical picture.

Static Picture

When the rotor frequency tends to zero ($\omega_\theta \to 0$) *i.e.* the rotors become rigid as their moment of inertia tends to infinity($I \to \infty$), the rotors remain static. Therefore, as there is no kinetic motion of the rotors we neglect the term U_{ij} that represents coupling between any two rotors. Such a Hamiltonian would describe a static system with no torsional motion of rotors about their molecular axis. This would be physical resemblance to the case of polymer thin films. Additionally we impose the condition of nearest neighbor coupling to understand the effects on exciton transport in linear polymeric chains. In this limit the potential matrix $\mathbb{V}_{N\times N}$ would contain only the elements from

the superdiagonal and subdiagonal as the Hamiltonian $\hat{H}_S$ from equation (3.1) takes the form:

$$H = \hbar\omega_\varepsilon \, \mathbb{I}_{N\times N} + \mathbb{V}_{N\times N}. \tag{4.1 a}$$
$$\text{where } \langle i|\mathbb{V}_{N\times N}|j\rangle = V_{ij}\left[\delta_{i,j+1} + \delta_{i+1,j}\right] \qquad i,j > 0$$

Let $|\Psi\rangle$ be any arbitrary wave function of this system. This could be written in the excitonic basis as $|\Psi\rangle = \sum_n^N \langle n|\Psi\rangle|n\rangle$ and say $\langle\Psi|n\rangle = \psi_n$ then we have $|\Psi\rangle = \sum_n^N \psi_n|n\rangle$. The same could be written as a column matrix $\Psi_{N\times 1}$. Now the dynamics of such a system, whose description is given by the Hamiltonian ($\hat{H}$) is described by the time-dependent Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}\Psi_{N\times 1} = \left[\hbar\omega_\varepsilon \, \mathbb{I}_{N\times N} + \mathbb{V}_{N\times N}\right]\Psi_{N\times 1} \tag{4.1 b}$$

Dynamic Picture

In the complete form, our general Hamiltonian models the case of dynamic torsional disorder when the rotors have a finite frequencyω_θ and there finite coupling between the local torsional modes and the excitonic states. In our current work, we are interested in understanding the dynamics of exciton transport in linear conjugated molecules. Therefore we fix the potential matrix to be

$$\mathbb{V}_{in} = \begin{bmatrix} 0 & V_{12} & 0 & \cdots & \cdots & 0 \\ V_{21} & 0 & V_{23} & \cdots & \cdots & 0 \\ 0 & V_{32} & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & V_{(N-1)N} \\ 0 & 0 & \cdots & \cdots & V_{N(N-1)} & 0 \\ & & & & & \end{bmatrix}$$

In section(4.3) we present some of our initial results for the dynamic disordered case in linear conjugated polymers.

4.1 Ordered Chains

As a first step, in our attempt to understand the effect of torsional disorder on exciton transport, we study the case of ordered chains where all the polymer subunits or chromophores are completely aligned *i.e.* $\theta_1 = \cdots \theta_N = \theta$. Further we assume all the coupling factors $v'_{ij}s = v_o$. In this limit the Hamiltonian matrix would be a symmetric tridiagonal matrix.

$$H = \hbar\omega_\varepsilon \mathbb{I}_{N\times N} + \mathbb{V} \tag{4.2}$$

In order to find the time-dependent wave function for this system, we first find the eigenvalues and eigenvectors of this system and then carry out time propagation by operating with the unitary time evolution operator.

Eigenvalues and Eigenvectors

The eigenvectors of the matrices H and $\mathbb{V}$ are the same and their eigenvalues are related to each other by the relation $\alpha = \hbar\omega_\theta + v_o\lambda$, where α and λ are the eigenvalues of the H and $\mathbb{V}$ matrices respectively. Therefore, it is enough to find the eigenvalues and eigenvectors of $\mathbb{V}$ to obtain the wave function of the system. The matrix $\mathbb{V}$ is a symmetric tridiagonal matrix that 0 along the diagonal. Such matrices in different variations have been well studied in the literature and have an analytic solution for the eigenvalue equation $\mathbb{V}\Psi_{N\times 1} = \lambda\Psi_{N\times 1}$.

Considering $\mathbb{V} - \lambda \mathbb{I}_{_{N\times N}} = 0$ we get,

$$\begin{bmatrix} -\lambda & 1 & & & & & \\ 1 & -\lambda & 1 & & & & \\ & 1 & -\lambda & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -\lambda & 1 \\ & & & & & 1 & -\lambda \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_k \\ \vdots \\ \phi_N \end{bmatrix} = 0$$

Considering $\phi_o = \phi_{N+1} = 0$

$$\begin{bmatrix} \phi_o - \lambda\phi_1 + \phi_2 \\ \phi_1 - \lambda\phi_2 + \phi_3 \\ \vdots \\ \phi_{k-1} - \lambda\phi_k + \phi_{k+1} \\ \vdots \\ \phi_{N-1} - \lambda\phi_N + \phi_{N+1} \end{bmatrix} = 0 \tag{4.3}$$

Thus we have system of second order linear difference equations of the form $\phi_{k-1} - \lambda\phi_k + \phi_{k+1}$, with boundary conditions $\phi_o = \phi_{N+1} = 0$. The solution to such a system of equations is of the form $\phi_k = x^k$, substituting this in the equation of the second order linear difference equations we get,

$$\phi_{k-1} - \lambda\phi_k + \phi_{k+1} = 0 \tag{4.4}$$

$$x^{k-1} - \lambda x^k + x^{k+1} = 0$$

$$x^k \left[\frac{1}{x} - \lambda + x\right] = 0 \tag{4.5}$$

The equation(4.5) has a trivial solution $x^k = 0$ and the non-trivial solution for $x^2 - \lambda x + 1 = 0$. The roots for which are given by

$$x_{\pm} = \frac{\lambda \pm \sqrt{\lambda^2 - 4}}{2} \tag{4.6}$$

We get $x_+ x_- = 1$ where, $x_+ = x$ and $x_- = \frac{1}{x}$. The general solution is given by $\phi_k = Ax_+^k + Bx_-^k$. Considering the boundary conditions $v_o = 0$, we get

$$\phi_k = A(x^k - x^{-k}) \text{ where } k = 0, 1, 2, 3 \cdots N + 1 \tag{4.7}$$

For a non-trivial solution we need $A \neq 0$ and by applying the second boundary condition $\phi_{N+1} = 0$, we get

$$x^{N+1} - x^{-(N+1)} = 0 \quad i.e. \quad x^{2(N+1)} = 1.$$

Then we have, $|x| = 1$. Using the relations between x_+ and x_-, we get

$$|\lambda| \leq |x| + |x|^{-1} = 2 \quad i.e. \quad |\lambda| \leq 2 \tag{4.8}$$

Actually, equation(4.6) has two possible solutions $\lambda = \pm 1$ and $\lambda \neq \pm 1$. For the case $\lambda = \pm 1$ we do not have a nontrivial solution. In the later case, $\lambda \neq \pm 1$ we found that $|\lambda| \leq 2$ and $|x| = 1$. Thus, we could consider $x = e^{i\theta}$ and we have $x^{2(n+1)} = e^{2i(N+1)\theta} = 1$.

$$\cos(2(N+1)\theta) = 1$$
$$\theta = \frac{m\pi}{(N+1)} \qquad 1 \leq m \leq N$$

Substituting $x = e^{i\theta}$ in equations(4.5) and (4.6) we get $\lambda = 2\cos(\theta)$.

$$\lambda_m = 2\cos(\frac{m\pi}{N+1}) \qquad \text{where } 1 \leq m \leq N \tag{4.9}$$

Therefore, the eigenvalues of the Hamiltonian are

$$\alpha_m = \hbar\omega_\varepsilon + 2v_o \cos(\frac{m\pi}{N+1}) \tag{4.10}$$

$$\phi_k^{(m)} = 2iA \sin(\frac{km\pi}{N+1}) \qquad \text{where } m = 1, 2, ...N$$

The eigen vectors of $\mathbb{V}$ are $|\phi_k\rangle = 2iA\ \left[\sin(\frac{\pi}{N+1}), \sin(\frac{2k\pi}{N+1}), ... \sin(\frac{Nk\pi}{N+1})\right]$. Considering the norm of $|\phi_k\rangle$

$$\begin{aligned}
\|\phi_k\|^2 =& \langle\phi_k|\phi_k\rangle = 1 \\
\|\phi_k\|^2 =& -4A^2 \sum_n^N \sin^2(\frac{nk\pi}{N+1}) \\
\therefore A^2 =& \frac{1}{4(-1)\sum_n^N \sin^2(\frac{nk\pi}{N+1})}.
\end{aligned}$$

Let $\theta = \frac{k\pi}{N+1}$, then

$$\begin{aligned}
\sum_m^N \sin^2(\frac{mk\pi}{N+1}) =& \sum_m^N \frac{[1-\cos 2m\theta]}{2} \\
=& \frac{1}{2}\left[N - \sum_m^N \cos 2m\theta\right] \qquad (4.11\ \text{a})
\end{aligned}$$

Using the euler formula, we get a summation of two *geometric progressions*

$$\begin{aligned}
\sum_m^N \cos(2m\theta) =& \sum_m^N \frac{e^{2im\theta} + e^{-2im\theta}}{2} \\
=& \frac{1}{2}\left[\frac{e^{2i\theta}(1-e^{2iN\theta})}{1-e^{2i\theta}} + \frac{e^{-2i\theta}(1-e^{-2iN\theta})}{1-e^{-2i\theta}}\right] \\
=& \frac{1}{2}\left[\frac{(e^{2i\theta}-1)(1-e^{2iN\theta}) + (e^{-2i\theta}-1)(1-e^{-2iN\theta})}{1-e^{2i\theta}-e^{-2i\theta}+1}\right] \\
\therefore \sum_m^N \cos(2m\theta) =& \frac{1}{2}\left[\frac{\cos 2N\theta + \cos 2\theta - \cos(2(N+1)\theta) - 1}{1-\cos 2\theta}\right] \qquad (4.11\ \text{b})
\end{aligned}$$

Substituting equation(4.11(b)) in (4.11(a)), we simplify the summation $\sum_m^N \sin^2(\frac{mk\pi}{N+1})$.

$$\begin{aligned}
\sum_m^N \sin^2(\frac{mk\pi}{N+1}) =& \frac{1}{2}\left[N - \frac{1}{2}\left(\frac{\cos 2N\theta - \cos(2(N+1)\theta)}{1-\cos 2\theta} - 1\right)\right] \\
=& \frac{N}{2} + \frac{1}{4} + \frac{\cos(2(N+1)\theta) - \cos 2N\theta}{4(1-\cos 2\theta)} \\
=& \frac{N}{2} + \frac{1}{4} + \frac{\cos(2mk\pi) - \cos\frac{2mNk\pi}{N+1}}{4(1-\cos\frac{2mNk\pi}{N+1})} \\
=& \frac{N}{2} + \frac{1}{4} + \frac{1-\cos\frac{2mNk\pi}{N+1}}{4(1-\cos\frac{2mNk\pi}{N+1})} \\
=& \frac{N}{2} + \frac{1}{4} + \frac{1}{4} \therefore \sum_m^N \\
\sin^2(\frac{mk\pi}{N+1}) =& \frac{N+1}{2} \qquad (4.12) \\
\text{and } A = & \frac{1}{i\sqrt{2(N+1)}}
\end{aligned}$$

Therefore the normalized eigenvectors are

$$|\phi_k\rangle = \sqrt{\frac{2}{N+1}} \begin{bmatrix} \sin(\frac{k\pi}{N+1}) \\ \sin(\frac{2k\pi}{N+1}) \\ \vdots \\ \sin(\frac{Nk\pi}{N+1}) \end{bmatrix} \qquad (4.13)$$

Time Propagation

Considering that our Hamiltonian is independent of time. The time evolution operator $\mathcal{U}(t,t_o)$ maps the initial state of the system at t_o to the state of the system at t. It works as $|\Psi(t)\rangle = \mathcal{U}(t,t_o)|\Psi(t_o)\rangle$. If $\hat{H}$ is hermitian, then $\mathcal{U}(t,t_o)$ is a unitary operator and $\mathcal{U}$ could be defined as

$$\begin{aligned}
\mathcal{U}(t,t_o) &= e^{-i\frac{\hat{H}(t-t_o)}{\hbar}} \qquad (4.14) \\
|\Psi(t)\rangle &= e^{-i\frac{\hat{H}(t-t_o)}{\hbar}}|\Psi(t_o)\rangle
\end{aligned}$$

The initial state of the wave function $|\Psi(0)\rangle$ could be written as

$$|\Psi(0)\rangle = \sum_{k}^{N} \langle \phi_k | \Psi(0) \rangle | \phi_k \rangle \tag{4.15}$$

Therefore, solution to the time-dependent Schrodinger equation as a coherent superposition could be written as

$$\begin{aligned} |\Psi(t)\rangle &= \sum_{k}^{N} e^{-i\frac{\hat{H}}{\hbar}t} \langle \phi_k | \Psi(0) \rangle \ |\phi_k\rangle \\ &= \sum_{k}^{N} \langle \phi_k | \Psi(0) \rangle \ e^{-\frac{1}{\hbar} i \alpha_k t} |\phi_k\rangle \end{aligned}$$

The eigenket $|\phi_k\rangle$ could be expanded in terms of site basis $\{|n\rangle's\}$ as

$$|\phi_k\rangle = \sqrt{\frac{2}{N+1}} \sum_{n}^{N} \sin(\frac{nk\pi}{N+1}) |n\rangle \tag{4.16}$$

Transition Probability $|\mathcal{U}_{N,1}(\tau)|^2$

We wish to compute $|1\rangle$ to $|N\rangle$ transition probability to understand the dynamics of exciton migration in the ordered polymeric chains where all the chromophores are completely aligned. For this we consider that a frank condon type excitation generates the exciton at site $|1\rangle$. The transition probability $|\mathcal{U}_{N,1}(t)|^2$ tell us how the probability of finding exciton at $|N\rangle$ evolves with time given that initial excitation was created at site $|1\rangle$.

$$|\mathcal{U}_{N,1}(\tau)|^2 = |\langle N | e^{-i\hat{H}\tau} | 1 \rangle|^2$$

Here we consider t as dimensionless time, $\tau = \frac{Vt}{\hbar}$. Both $|N\rangle$ and $|1\rangle$ can be expanded in the eigen basis as follows:

$$
\begin{aligned}
|N\rangle &= \sum_{m}^{N} \langle \phi_m | N \rangle \ |\phi_i\rangle \\
|1\rangle &= \sum_{n}^{N} \langle \phi_n | 1 \rangle \ |\phi_j\rangle \langle N | e^{-i\hat{H}\tau} | 1 \rangle \\
&= \sum_{m}^{N} \sum_{n}^{N} \langle \phi_m | N \rangle^* \langle \phi_n | 1 \rangle \langle \phi_m | e^{-i\hat{H}\tau} | \phi_n \rangle \\
&= \sum_{m}^{N} \sum_{n}^{N} \langle \phi_m | N \rangle^* \langle \phi_n | 1 \rangle \langle \phi_m | e^{-iE_n\tau} | \phi_n \rangle \\
&= \sum_{m}^{N} \sum_{n}^{N} \sin(\frac{m\pi N}{N+1}) \sin(\frac{n\pi}{N+1}) e^{-iE_n\tau} \ \delta_{m,n} \\
\therefore \langle N | e^{-i\hat{H}\tau} | 1 \rangle &= \sum_{m}^{N} \sin(\frac{m\pi N}{N+1}) \sin(\frac{m\pi}{N+1}) e^{-iE_m\tau} \qquad (4.17)
\end{aligned}
$$

Let $a_m = \frac{2}{N+1} \sin(\frac{m\pi}{N+1}) \sin(\frac{mN\pi}{N+1})$ then, $\mathcal{U}_{N,1}(\tau) = \sum_m^N a_m e^{-iE_m\tau}$. The transition probability would be:

$$
\begin{aligned}
|\mathcal{U}_{N,1}(\tau)|^2 &= |\sum_{m}^{N} a_m e^{-iE_m\tau}|^2 \\
&= \sum_{i}^{N} |a_i|^2 + \sum_{i \neq j}^{X} a_m {a_n}^* \ e^{-i \ (E_m - E_n)\tau} \\
&= \sum_{m}^{N} {a_m}^2 + 2 \sum_{m>n}^{N} a_m {a_n}^* \cos((E_m - E_n)\tau) \\
\therefore |\mathcal{U}_{N,1}(\tau)|^2 &= \sum_{m}^{N} {a_m}^2 + 2 \sum_{m>n}^{N} a_m a_n \cos(E_m - E_n)t) \qquad (4.18)
\end{aligned}
$$

Using the expression (4.18), we find the transition probability for different sizes of the system to evolve with time as follows[1] [2] :

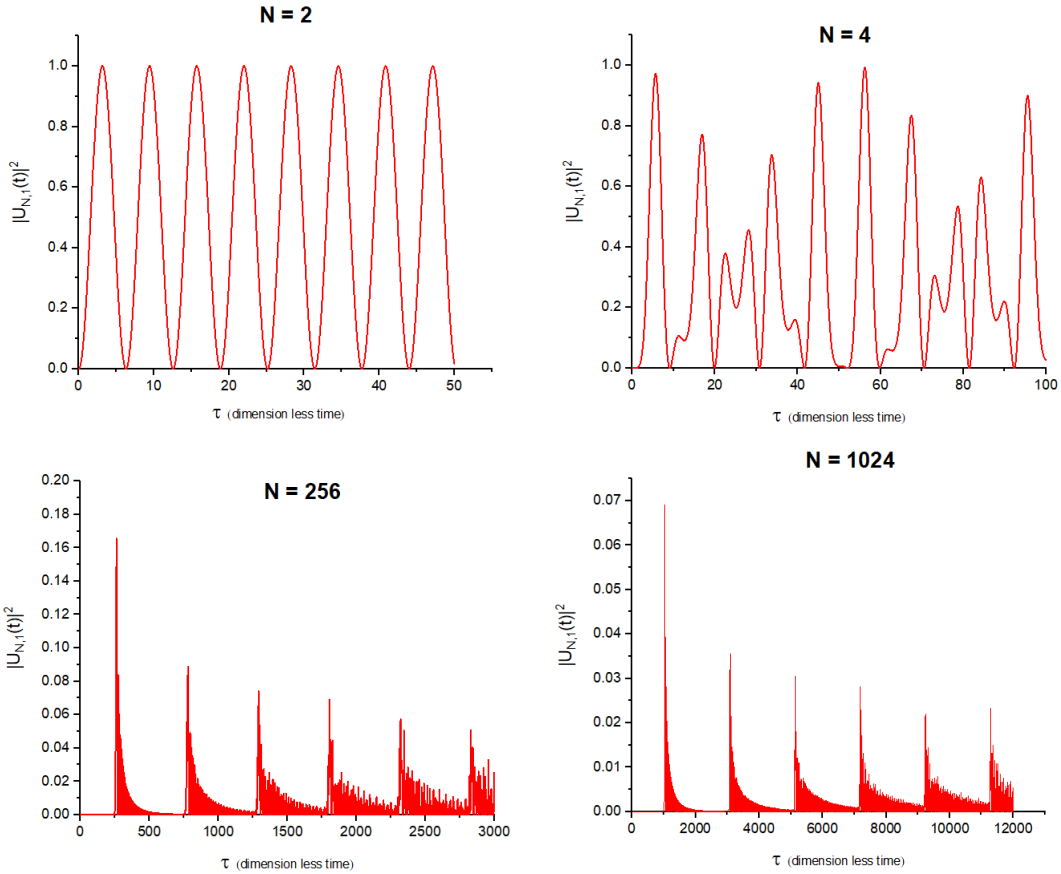


Figure 4.1: Transition Probability plots in case of Ordered Chains.

In these representative plots of end-to-end transition probability, we clearly observe an oscillatory behavior and it is wilder in case of higher values of N. As N increases the contributions from various eigenvalues to phase factor also increase and this could be the reason for wilder oscillations in case of longer ordered chains.

[1] To perform these calculations and plot the data, we have used the numpy and matplotlib libraries in python.

[2] We only present representative examples here and provide the python codes employed for this in the appendix-C.

Time-averaged Transition Probability

The time average transition probability would be

$$
\begin{aligned}
P_N &= \lim_{T\to\infty}\frac{1}{T}\int_0^T |\mathcal{U}_{N,1}(\tau)|^2 d\tau \\
&= \lim_{T\to\infty}\frac{1}{T}\left[\sum_m^N a_m^2 T + 2\sum_{m>n}^N a_m a_n \frac{\sin[(E_m-E_n)T]}{(E_m-E_n)}\right] \\
&= \lim_{T\to\infty}\left[\sum_m^N a_m^2 + 2\sum_{m>n}^N a_m a_n \frac{\sin[(E_m-E_n)T]}{(E_m-E_n)T}\right] \\
&= \sum_m^N a_m^2 + 2\sum_{m>n}^N a_m a_n \lim_{T\to\infty}\frac{\sin[(E_m-E_n)T]}{(E_m-E_n)T} \qquad \text{as } \lim_{x\to\infty}\frac{\sin(kx)}{kx}=0 \\
&= \frac{4}{(N+1)^2}\sum_m^N \sin^2\!\left(\frac{m\pi}{N+1}\right)\sin^2\!\left(\frac{mN\pi}{N+1}\right) \qquad \text{let } \theta=\frac{\pi}{N+1} \\
&= \frac{4}{(N+1)^2}\sum_m^N \frac{[1-\cos 2m\theta]}{2}\frac{[1-\cos 2mN\theta]}{2} \\
&= \frac{1}{(N+1)^2}\sum_m^N [1-\cos 2mN\theta-\cos 2m\theta+\cos 2m\theta\cos 2mN\theta] \\
&= \frac{1}{(N+1)^2}\left[\sum_m^N 1-\sum_m^N \cos 2mN\theta-\sum_m^N \cos 2m\theta+\sum_m^N \cos 2m\theta\cos 2mN\theta\right] \\
&\quad \text{as } \sum_k^N \cos(kx) = -1 \text{ when } x=\frac{\pi}{N+1} \\
&= \frac{1}{(N+1)^2}\left[N-(-1)-(-1)+\sum_m^N \cos 2m\theta\cos 2mN\theta\right] \\
&\quad \text{using the identity } \cos(x)\cos(y)=\frac{1}{2}[\cos(x-y)+\cos(x+y)] \\
&= \frac{1}{(N+1)^2}\left[N+2+\frac{1}{2}\sum_m^N [\cos(2m(N-1)\theta)+\cos(2m(N+1)\theta)]\right] \\
&= \frac{1}{(N+1)^2}\left[N+2+\frac{1}{2}\sum_m^N \cos(2m(N-1)\theta)+\frac{1}{2}\sum_m^N \cos(2m(N+1)\theta)\right] \\
&= \frac{1}{(N+1)^2}\left[N+\frac{3}{2}+\frac{1}{2}\sum_m^N \cos\left(2m(N+1)\frac{\pi}{(N+1)}\right)\right]
\end{aligned}
$$

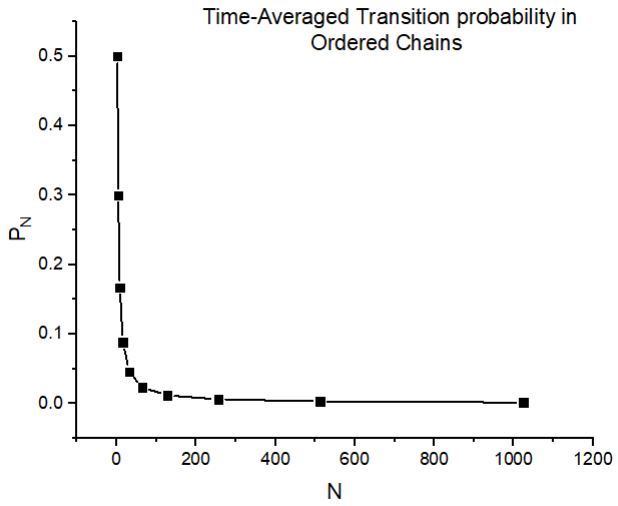


Figure 4.2: Time-Averaged Transition Probability as a function of chain length of the polymer.

$$\begin{aligned} P_N =& \frac{1}{(N+1)^2}\left[N+\frac{3}{2}+\frac{1}{2}\sum_m^N \cos\left(2m\pi\right)\right] \\ =& \frac{1}{(N+1)^2}\left[N+\frac{3}{2}+\frac{N}{2}\right] \\ =& \frac{1}{(N+1)^2}\left[\frac{3}{2}(N+1)\right] \\ \therefore P_N =& \frac{3}{2(N+1)} \end{aligned} \tag{4.19}$$

Thus, the time average transition probability $\lim_{T\to\infty}\frac{1}{T}\int_0^T |\mathcal{U}_{N,1}(\tau)|^2 d\tau$ is given by $P_N = \frac{3}{2(N+1)}$.

Defects in ordered chains

The Hamiltonian presented in this section describe the over simplified cases of single impurities or defects being present in a perfectly ordered chain. In spite of their simplicity, these Hamiltonians provide an qualitative understanding on the impact of a single disordered site on exciton transport. We treat these defects as a first order perturbation in an ordered chain and employ

numerical calculations to find the transition probability and time averaged transition probability. The general form of the Hamiltonian would be:

$$\begin{aligned} H =& H_o + H_1 \\ H_o =& \hbar\omega_\varepsilon \mathbb{I}_{N\times N} + \mathbb{V} \end{aligned}$$

Where H_1 is the first order perturbation in the system. Here we could two different types, chemical and torsional. A chemical defect in this could be defined as a site with a higher site energy within the order chain and torsional defect could be defined as misalignment of the rotor or at between any two rotors within the chain. While chemical defects present a case of diagonal disorder, torsional defects are cases of off-diagonal disorder.

Chemical Defects

Here we consider a case where the chromophore at some m^{th} has higher site energy than the rest of the chromophores, say $\varepsilon_n = \hbar\omega_\varepsilon + \epsilon$. This is analogous to a situation when a polymer chain has chemical impurity due to the synthetic process that is adopted. In this case the perturbation H_1 is given by:

$$H_1 = \begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & \epsilon & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} \tag{4.20}$$

We present representative example of transition probability plot against dimensionless time τ at the end of this section. [3].

[3]For this we randomly sample ϵ over a uniform distribution ranging from 1 to 10 times the site energy elsewhere in the chain. Further we carry out configurational averaging as discussed in section(4.2).

Torsional Defects

In case of torsional defects, there are two possibilities: **(i)**One of the rotors in the ordered chains in misaligned and rest of the entire chain remains intact in the same plain.**(ii)**The torsional defect could also manifest in the form of a conjugation break between any two sites separating it into two ordered chains.

Case(i):

When the $(m-1)^{th}$ and $(m+1)^{th}$ chromophores are not in the same plane as the m^{th} chromophore. (i.e.)

$\theta_1 = \theta_2 ..... = \theta_{m-1} = \alpha$ and $\theta_{m+1} = \theta_{m+2} ..... = \theta_N = \beta$.

In the case the H_1 turns out be as follows:

$$H_1 = \begin{bmatrix} 0 & & & & & & \\ & \ddots & & & & & \\ & & 0 & 0 & -1+\cos(\alpha-\theta_m) & & \\ & & -1+\cos(\alpha-\theta_m) & 0 & -1+\cos(\beta-\theta_m) & & \\ & & -1+\cos(\beta-\theta_m) & 0 & 0 & & \\ & & & & \ddots & & \\ & & & & & & 0 \end{bmatrix} \tag{4.21}$$

In this case the transition probability and time-averaged transition probability are as follows:

(graphs and table)

Case(ii):

When the m^{th} and $(m+1)^{th}$ chromophores are not in the same plane. (i.e.)

$\theta_1 = \theta_2 ..... = \theta_m = \alpha$ and $\theta_{m+1} = \theta_{m+2} ..... = \theta_N = \beta$.

In the case the H_1 turns out be as follows:

$$H_1 = \begin{bmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & 0 & -1+\cos(\alpha-\beta) & \\ & & -1+\cos(\alpha-\beta) & 0 & 0 & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} \tag{4.22}$$

In this case the transition probability and time-averaged transition probability are as follows:

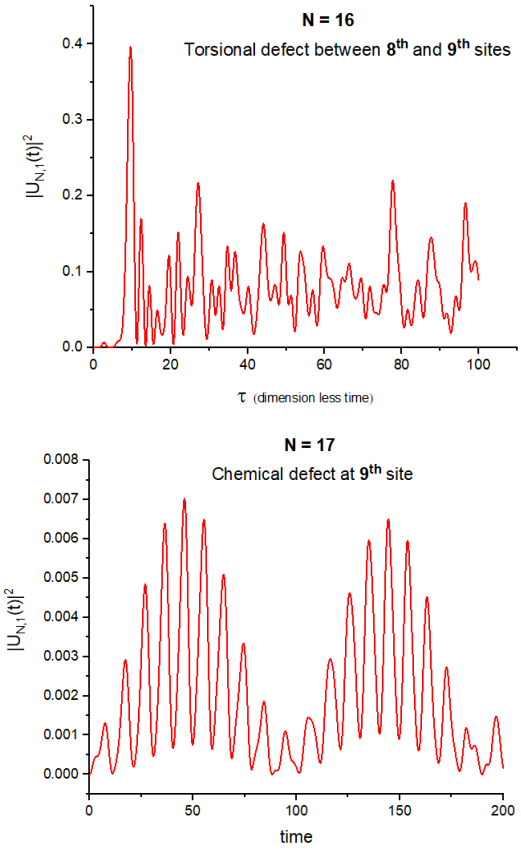


Figure 4.3: Transition Probability plots of ordered chains with defects.

4.2 Static Disorder

In a realistic setting an important factor that determines the optical properties of conjugated polymers is static disorder. The origin of static disorder could be both energetic and structural in nature. As discussed in the previous section, chemical defects are energetic defects that arise due to the synthetic scheme. Torsional defects arise due to strain in the molecule and are structural in nature. These ideas of disorder have been of great interest in condensed matter systems, as disorder beyond a threshold value could induce interesting phase transitions in material. In the following section we briefly discuss these ideas of disorder and configurational averaging.

Disorder Averaging

In his seminal paper, "Absence of Diffusion in Certain Random Lattices", P.W. Anderson suggested that the localization of charge carriers due to increasing disorder in the system was responsible for a phase transition from a conductor to an insulator. This sudden fall in the conductivity of a material as the disorder increases beyond a certain threshold has been attributed to the localization of charge carriers. It is now understood to be an interference phenomena, where the wave amplitudes associated with various tunneling paths cancel out each other [25]. A similar picture could be used to describe transport in the case of molecular systems, where the charge carriers or excitons could get scattered due to presence of defects. Here the resultant inhomogeneous broadening could described as a *configurational average*, an energy spectrum averaged over different random arrangements of the system [6]. For a sample volume $\mathcal{V}$ that contains different possible arrangements of system (say V number of sample arrangements) parameterised by a structural parameter θ_i that varies with disorder, the configurational average for

any property $f(\omega, \theta)$ is given by

$$F(\omega) = \frac{1}{V} \sum_{A \in \mathcal{V}} f(\omega, \theta_i) \tag{4.23}$$

For a large number of realisations of the system, the summation in the configurational average could be replaced by an integrand .

$$F(\omega) = \frac{1}{V} \int_{\mathcal{V}} \mathcal{F}(\theta) f(\omega, \theta_i) \partial\theta \tag{4.24 a}$$

where $\mathcal{F}(\theta)$ is the normalized distribution function, given by

$$\mathcal{F}(\theta) = \frac{1}{V} \sum_{A \in \mathcal{V}} \prod_{j} \delta(\theta_A - \theta_j) \tag{4.24 b}$$

The extent of inhomogeneous broadening of the spectrum is dependent on the distribution of microscopic parameters. In case of disordered systems it becomes necessary to do a configurational averaging to account for various effects introduced due to the ergodicity of the microscopic parameters. It provides a more realistic picture in the description of molecular systems that are inherently disordered. In the following sections we proceed to calculate the time-averaged transition probability using disorder averaging.

Configurational Averaging

We consider $\hat{H}_S$ from equation 4.1(a) as function of $\Theta = \{\theta_1, \theta_2 \cdots \theta_N\}$.

$$\hat{H}_S[\Theta] = V \sum_{i}^{N-1} \cos(\theta_i - \theta_{i+1}) \, [\ |i\rangle\langle i+1| + |i+1\rangle\langle i| \ ] \tag{4.25}$$

Then consider that the Hamiltonian $\hat{H}_S$ has the energy spectrum

$$\hat{H}_S[\Theta] \ |\phi_n[\Theta]\rangle = \varepsilon_n[\Theta] \ |\phi_n[\Theta]\rangle \qquad \text{where } n = 1, 2, 3...N$$

We consider that the eigenvectors $\{|\phi_n[\Theta]\rangle's\}$ and eigenvalues $\{\varepsilon_n[\Theta]'s\}$ are functions of Θ and expand $\langle N|$ and $|1\rangle$ in terms of eigenvectors. Then the transition probability [4] is given by

$$\begin{aligned}|\mathcal{U}_{N,1}(\tau)|^2 =&|\langle N|e^{-i\hat{H}[\Theta]\tau}|1\rangle|^2 \\ =&\sum_m^N\sum_n^N\langle 1|\phi_m[\Theta]\rangle\langle\phi_m[\Theta]|N\rangle\langle N|\phi_n[\Theta]\rangle\langle\phi_n[\Theta]|1\rangle e^{-i(\varepsilon_m[\Theta]-\varepsilon_n[\Theta])\tau}\end{aligned}$$

Therefore the time average transition probability($P_N[\Theta]$) is given by

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T|\mathcal{U}_{N,1}[\Theta,\tau]|^2d\tau=\lim_{T\to\infty}\frac{1}{T}\int_0^T\sum_m^N\sum_n^N\langle 1|\phi_m[\Theta]\rangle\langle\phi_m[\Theta]|N\rangle \langle N|\phi_n[\Theta]\rangle\langle\phi_n[\Theta]|1\rangle e^{-i(\varepsilon_m[\Theta]-\varepsilon_n[\Theta])\tau}$$

$$P_N=\sum_m^N\sum_n^N\langle 1|\phi_m[\Theta]\rangle\langle\phi_m[\Theta]|N\rangle\langle N|\phi_n[\Theta]\rangle\langle\phi_n[\Theta]|1\rangle \lim_{T\to\infty}\frac{1}{T}\int_0^T e^{-i(\varepsilon_m[\Theta]-\varepsilon_n[\Theta])\tau}d\tau$$

$$\therefore P_N=\sum_{\substack{m\\ \varepsilon_m[\Theta]=\varepsilon_n[\Theta]}}^N|\langle 1|\phi_m[\Theta]\rangle|^2|\langle\phi_m[\Theta]|N\rangle|^2 \tag{4.26}$$

From equation 4.24 (a), we know that disorder averaged time-averaged transition probability is given by

$$\langle\ P_N[\Theta]\ \rangle_{average}=\int_{\Theta\in[0,2\pi]^N}P[\Theta]\left[\lim_{T\to\infty}\frac{1}{T}\int_0^T\ |\ \mathcal{U}_{N,1}[\Theta,\tau]\ |^2d\tau\right]\ d^N\Theta \tag{4.27}$$

[4] Where τ is the dimensionaless time that we have previously defined.

Similarly, the disorder averaged transition probability is given by

$$\langle\, \mathcal{U}_{N,1}[\Theta,\tau]\ |^2\ \rangle_{average} = \int_{\Theta\in[0,2\pi]^N} P[\Theta]\left[|\langle N|e^{-i\hat{H}[\Theta]\tau}|1\rangle|^2\right]\ d^N\Theta \qquad (4.28)$$

Using the expressions 4.27 and 4.28 we have numerically calculated[5] the

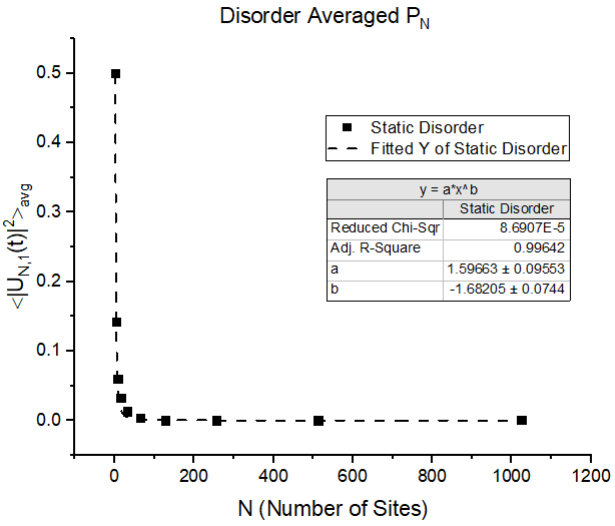


Figure 4.4: Time-averaged transition probability plotted as a function of N for polymer chains with static torsional disorder.

disorder averages of transition probability and time-averaged transition probability. Here we also present the scaling behavior of the time-averaged transition probability with respect to growing chain size. We find that the time averaged transition probability for any given value of N is higher in case of the ordered chains when compared to the static disordered case. We also observe a trend that in case of single defects in ordered chains, the impact of case(ii) torsional defects is less, when compared to case(i) torsional defects. Though chemical defects preserve coherence, they drastically reduce the time-averaged transition probability value for an exciton moving from $|1\rangle$ to $|N\rangle$.We find that time averaged transition probability decays as a power law with an exponent of -1.68 *i.e.* ie decays faster than the completely disor-

[5]We have performed configurational averaging over a sample space with 1,00,000 samples generated by drawing randomly from a uniform distribution.

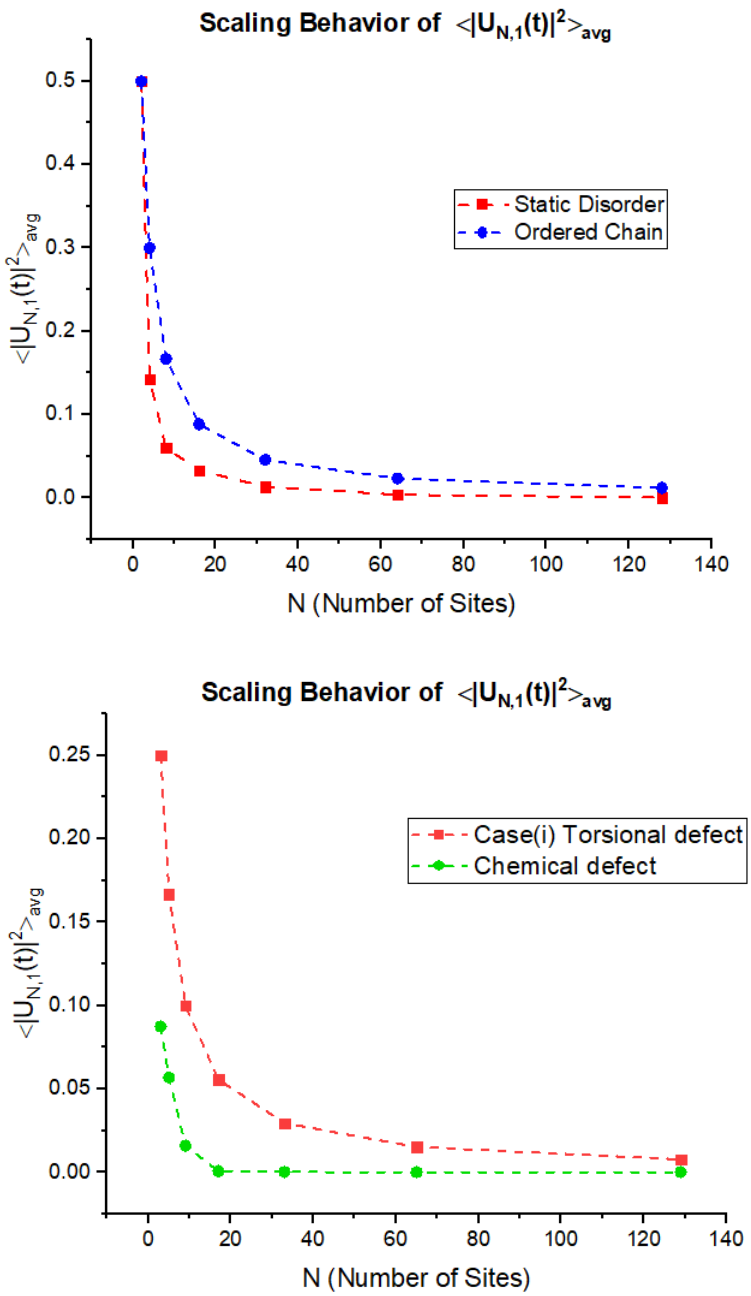


Figure 4.5: Scaling behavior of Time-averaged transition probability

dered case. In the following section, we consider the case of dynamic torsional disorder.

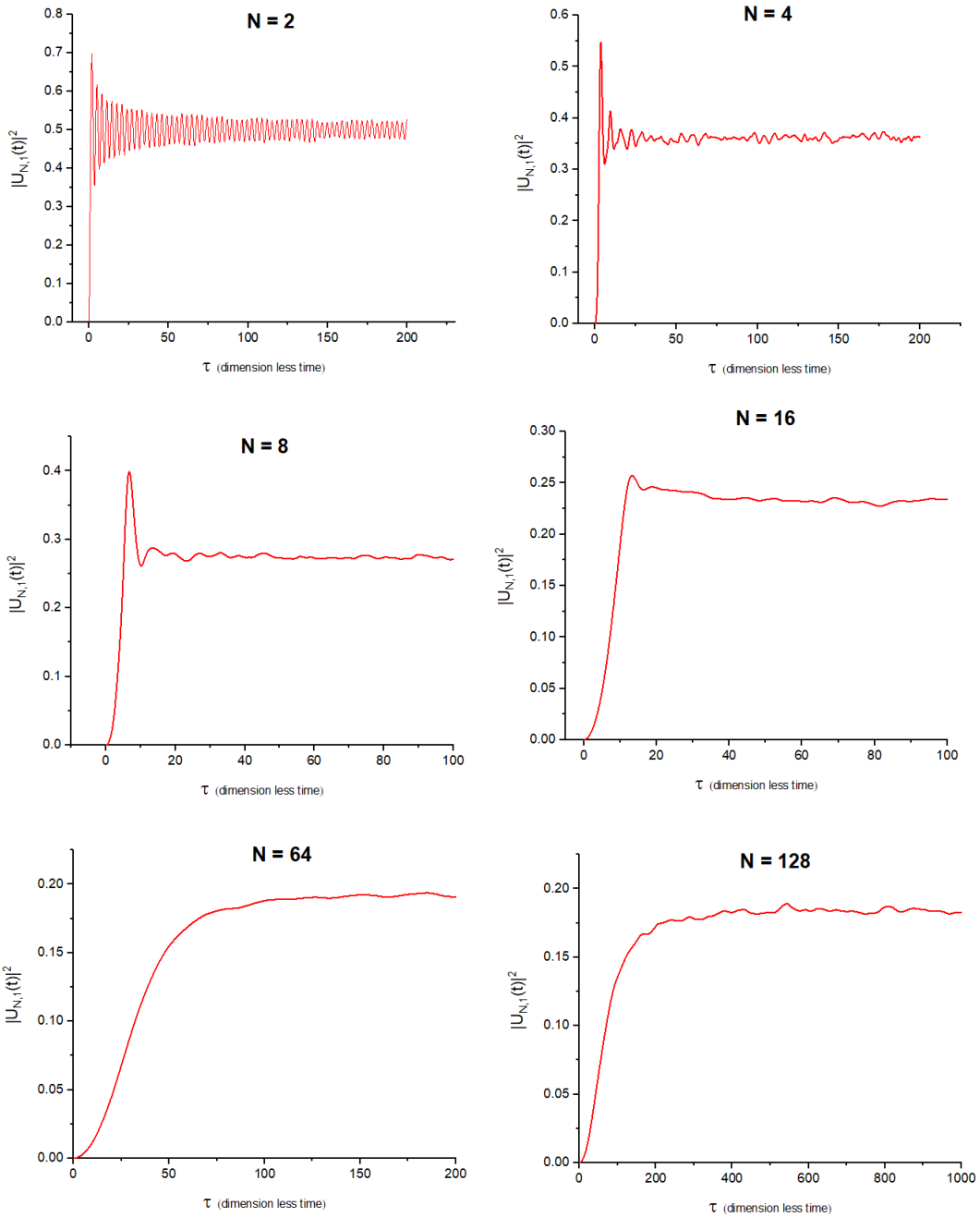


Figure 4.6: Representative examples of Transition probability as function of dimensionless time τ for polymer chains with static torsional disorder

4.3 Dynamic Torsional Disorder

To model the case dynamic disorder we begin by considering the complete Hamiltonian of our model. We found it difficult to find an analytic solution for the system defined by equation 3.5, therefore we have resorted to numerical methods to solve the same. As a first step, we have Fourier transformed the time-dependent Schrödinger wave equation and obtained the equation (3.9). In the equation (3.9), we find that there infinitely many momentum states and to make the problem tractable, we truncate the momentum states K from $\mathbb{Z}^N$ to a finite set $[-M, M]^N$. By truncating the momentum space we are indirectly discretising the torsional angles ($\theta_i \in [0, 2\pi]$) to $2M+1$ equally spaced states. Even though we have restricted the number of momentum states to a finite value the dimensionality of truncated Hamiltonian matrix grows rapidly as an exponential $N(2M+1)^N$. This follows from the fact that the entanglement of such one-dimensional systems grows rapidly. As mentioned in the initial discussion of this chapter we fix the potential matrix to model linear chains with nearest neighbor coupling.

We chose to employ the exact diagonalisation using Lanczos algorithm to solve the problem numerically. These methods come with their own limitations, as the maximum size of a non-singular matrix that can be diagonalized using the commonly available quad core CPUs is limited to the order of a few thousands. While the exact dimensionality of the matrix that could be diagonalised depends on its sparseness, we could achieve diagonalisation of reasonably sparse matrices upto the size (15,000×15,000) [6]. The implementation of the numerical scheme in python is presented in appendix-C of this thesis. In the following section we outline the time propagation scheme that we have considered for numerical solution.

[6] All the computations for these calculations where run on a Desktop with Intel Core i5-4590 CPU with 7.7 GB of available memory.

Time propagation

Suppose on diagonalising the $X = N \times (2M+1)^N$ dimensional Hamiltonian matrix in the fourier space, we get X eigen vectors, say $\{|\phi_i\rangle's\}$. The expressions for the initial state in the site basis before and after fourier transform are given by

$$\begin{aligned} |\psi(0)\rangle &= \sum_{n}^{N} \psi_n |n\rangle \\ |\tilde{\psi}(0)\rangle &= \sum_{n}^{N} \sum_{-M}^{+M} \overset{N}{\cdots} \sum_{-M}^{+M} \tilde{\psi}_n[K] \, |n\rangle \otimes |k_1\rangle \otimes |k_2\rangle ... \otimes |k_N\rangle \end{aligned} \tag{4.29}$$

where K $= k_1, k_2, ..k_N$

The an arbitrary eigen vector $|\phi_i\rangle$ in the fourier transformed eigen basis is given by:

$$\begin{aligned} |\phi_i\rangle &= \sum_{n}^{N} \sum_{-M}^{+M} \overset{N}{\cdots} \sum_{-M}^{+M} c^{(i)}_{n,k_1,k_2...k_n} |n\rangle \otimes |k_1\rangle \otimes |k_2\rangle ... \otimes |k_N\rangle \\ \langle\phi_i| &= \sum_{m}^{N} \sum_{-M}^{+M} \overset{N}{\cdots} \sum_{-M}^{+M} c^{(i)*}_{m,k_1',k_2'...k_N'} \langle m| \otimes \langle k_1'| \otimes \langle k_2'| ... \otimes \langle k_N'| \end{aligned} \tag{4.30}$$

The initial state as a coherent superposition of eigenkets is given as :

$$\begin{aligned} |\tilde{\psi}(0)\rangle &= \sum_{i}^{N} \langle\phi_i|\tilde{\psi}(0)\rangle \, |\phi_i\rangle \\ &= \sum_{i}^{X} \sum_{n}^{N} \sum_{-M}^{+M} \overset{N}{\cdots} \sum_{-M}^{+M} c^{(i)*}_{n,k_1,k_2...k_n} \, \tilde{\psi}_n[K] \, |\phi_i\rangle \end{aligned}$$

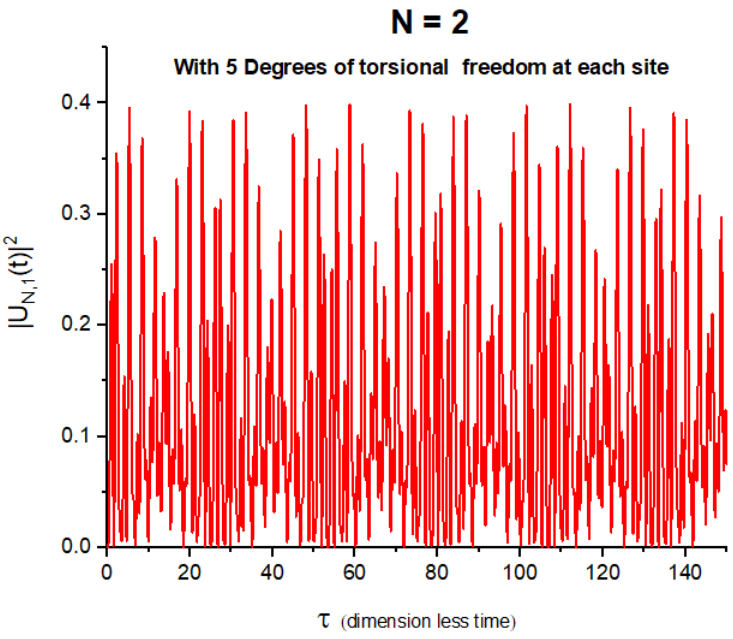
The wave function at any time t $|\tilde{\psi}(t)\rangle$ is given by

$$|\tilde{\psi}(t)\rangle = \sum_{i}^{X} \sum_{n}^{N} \sum_{-M}^{+M} \overset{N}{\cdots} \sum_{-M}^{+M} c^{(i)*}_{n,k_1,k_2...k_n} \, \tilde{\psi}_n[K] \, e^{-iE_i t} \, |\phi_i\rangle \tag{4.31}$$

We consider that initially all the rotors are in their zero momentum state $i.e$ $|k_1 = 0\rangle \otimes .....|k_N = 0\rangle$. Further, we create a Franck-Condon type excitation at site $|1\rangle$. Then given that such an excitation is created at $|1\rangle$ we wish to find the transition probability $|\mathcal{U}_{N,1}(\tau)|^2$.

$$|\mathcal{U}_{N,1}(\tau)|^2 = \sum_{N summations} ..... \sum \langle N| \otimes \langle k_1| \otimes .....\langle k_N| \; e^{iHt} \; |1\rangle \otimes |k_1 = 0\rangle \otimes .....|k_N = 0\rangle \tag{4.32}$$

Here we present a few representative examples from the exact diagonalisation studies of the complete Hamiltonian. From our initial results, we hypothesis that exciton migration occurs in a coherent fashion in the presence of dynamic torsional disorder in an isolated molecule.



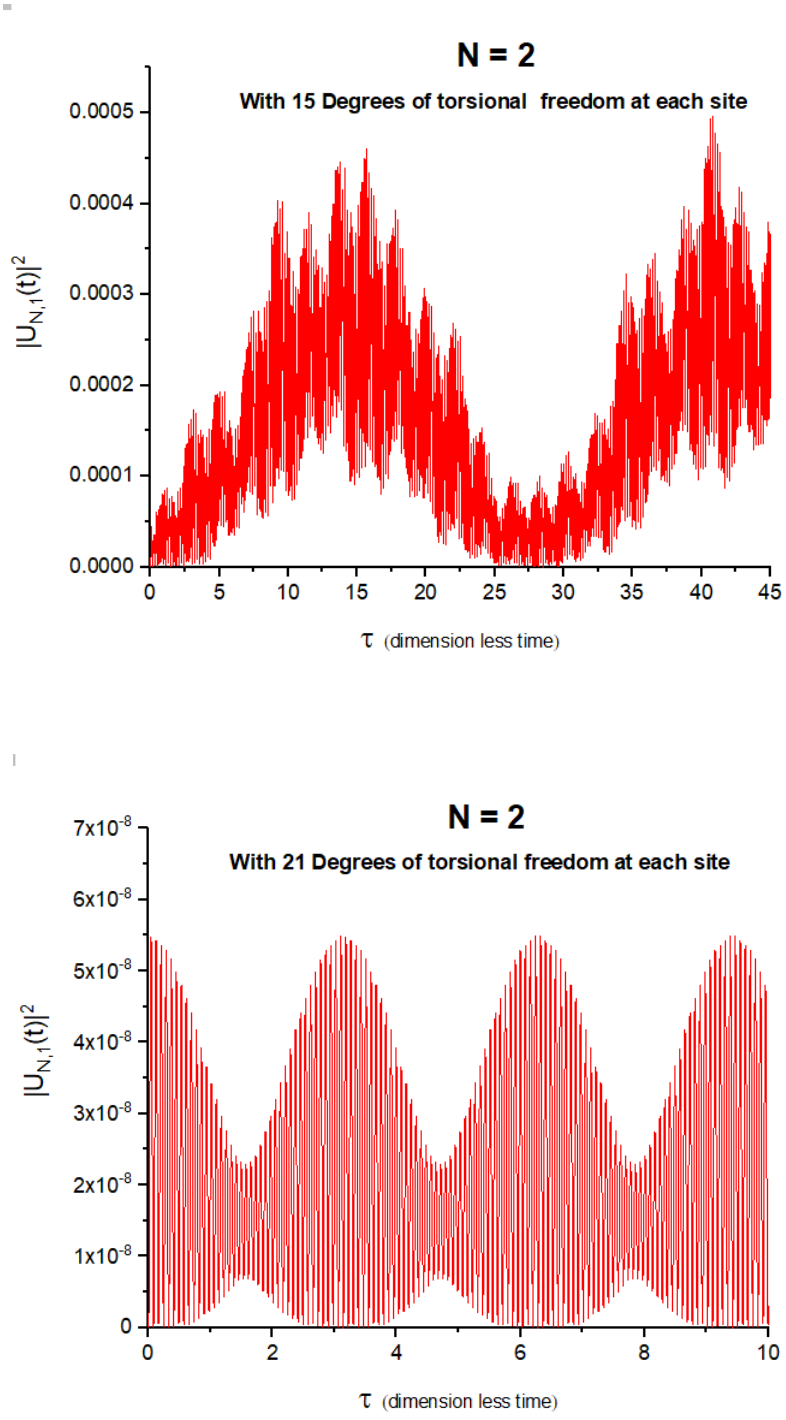


Figure 4.7: Transition probability plots for N=2 with different degrees of freedom at each site.

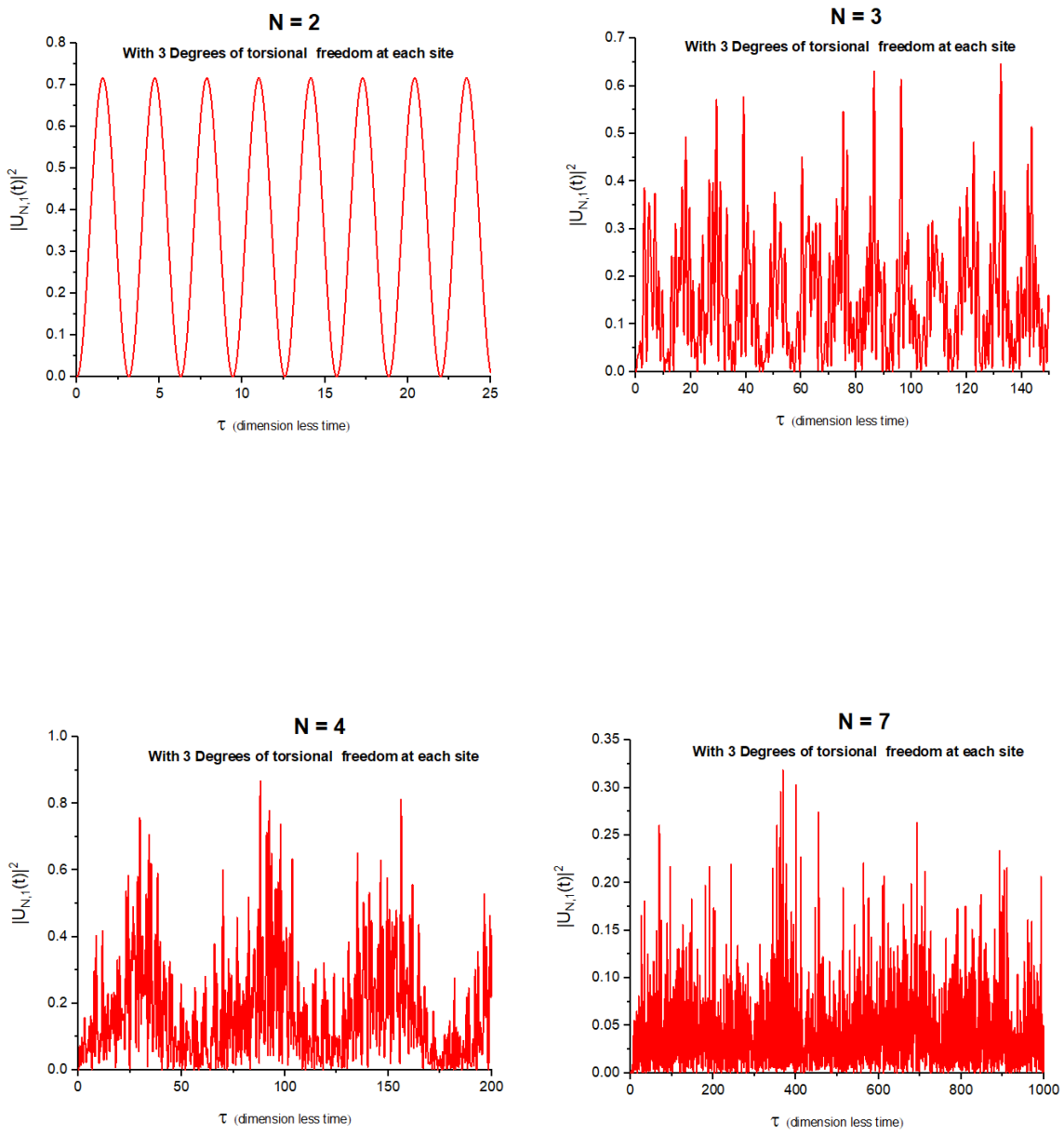


Figure 4.8: Representative examples of Transition probability as function of dimensionless time τ for polymer chains with dynamic torsional disorder.

"Somewhere, something incredible is waiting to be known."
Carl Sagan

5

Conclusions and Future Work

As a part of this dissertation work, we have proposed a general model to study the torsional effects on exciton transport in π-conjugated polymers. Using the model, we have studied the specific case of linear conjugated polymers with nearest neighbour coupling in various limits. At various limits of the parameters: $\omega_\theta, \omega_\varepsilon$ and V_{ij}, we have computed the end-to-end transition probability and its time-averaged value as a means to study the dynamics of the system. From our results, we find a trend in the scaling behavior time-averaged transition probability with respect to the size of the system across various limits. Based on these studies, we propose a conjecture that, the time-averaged transition probability for end-to-end exciton migration is always higher in ordered chains when compared to disordered chains. Comparing the different type of defects, it is evident that the nature and extent of disorder determine the type of transport in the isolated molecules. Finally, we have numerically solved the complete Hamiltonian proposed our model using exact diagonalisation method.

5.1 Future Work

In our work so far, we have limited ourselves to the simple case of nearest neighbor interactions. There are a wide range other possibilities that arise from the general Hamiltonian proposed in chapter 3. From cyclic systems to molecular aggregates many other systems could be studied using the proposed model. Many cases such as frustrated interactions and the cases modeling selective rotor coupling are yet to be explored. These present an opportunity to further extend this study on dynamics of exciton migration in conjugated polymers using this model.

While we have resorted to numerical methods to a great extent in our work and we speculate that there could be a possibile analytic solution for the case of static torsional disorder. An analytic solution, if derived could provide a more rigorous proof to the conjecture that we have proposed on time averaged transition probability for the ordered chains being greater than any case with disorder. The time-dependent Schrödinger equation with the complete Hamiltonian of our model generates a set of coupled partial differential equations that present a challenging task to find a solution in the closed form and one would have to check for the integrability of the model to initiate such a study. In another direction, we have also initiated the analytic calculations for the case of defects in ordered chains using Green's functions methods [26] and we have achieved considerable amount of success in the process. Even, the numerical methods that we have used suffers from exponential growth of entanglement in our system. In literature, the class of Renormalization group methods [27] have been proposed to find a tractable solution for studying larger sizes of entangled systems, similar to the one that we encounter in our model. Of all these renormalization group methods, density matrix renormalization group(DMRG) [28] methods have been quite successful for systems with linear growth of Von Neumann entropy [29]. At the time of submission of this dissertation, we are in the process of devel-

oping a time-dependent DMRG scheme for our model. The application of the DMRG algorithm using the Matrix Product states ansatz has been our primary interest all through the last leg of this dissertation work and we hope to arrive to a conclusion on these line very soon. We would like to continue this study further using these numerical methods to provide insights on effects of dynamic torsional disorder in linear conjugated polymers.

A

Fourier Transform of the Hamiltonian

Considering the time-dependent Schrödinger equation, we find the Fourier transform of this equation.

$$i\hbar\frac{\partial}{\partial t}\sum_{n=1}^{N}\psi_n[\Theta,t]|n\rangle = \underbrace{-\hbar\omega_\theta\sum_{i=1}^{N}\sum_{n=1}^{N}\frac{\partial^2\psi_n[\Theta,t]}{\partial\theta_i^2}\,|n\rangle}_{\mathbf{I}} + \underbrace{\sum_{\substack{i,j=1\\ i>j}}^{N}\sum_{n=1}^{N}U_{ij}\psi_n[\Theta,t]\,|n\rangle}_{\mathbf{II}}$$
$$+\underbrace{\sum_{\substack{n=1\\ i\neq n}}^{N}\sum_{i=1}^{N}V_{in}\psi_i[\Theta,t]\,|n\rangle}_{\mathbf{III}} + \underbrace{\hbar\omega_\varepsilon\sum_{n=1}^{N}\psi_n[\Theta,t]\,|n\rangle}_{\mathbf{IV}} \qquad (3.7)$$

Term I

Fourier transform of $-\hbar\omega_\theta \sum_{i,n}^{N} \frac{\partial^2 \psi_n[\Theta,t]}{\partial\theta_i^2}$ is given by

$$
\begin{aligned}
&-\frac{\hbar\omega_\theta}{(2\pi)^N}\int_0^{2\pi} d\Theta e^{-i\sum_{m=1}^{N} k_m^{'}\theta_m} \sum_{i,n}^{N}\frac{\partial^2}{\partial\theta_i^2}\left[\sum_{-\infty\ N\ sums}^{+\infty} \tilde{\psi}_n[K,t]e^{i\sum_{m=1}^{N} k_m\theta_m}\right] \\
=&-\frac{\hbar\omega_\theta}{(2\pi)^N}\sum_{n,i}^{N}\int_0^{2\pi} d\Theta e^{-i\sum_{m=1}^{N} k_m^{'}\theta_m}\left[\sum_{-\infty\ N\ sums}^{+\infty} \tilde{\psi}_n[K,t]\frac{\partial^2}{\partial\theta_i^2}e^{i\sum_{m=1}^{N} k_m\theta_m}\right] \\
=&-\frac{\hbar\omega_\theta}{(2\pi)^N}\sum_{n,i}^{N}\int_0^{2\pi} d\Theta e^{-i\sum_{m=1}^{N} k_m^{'}\theta_m}\left[\sum_{-\infty\ N\ sums}^{+\infty} \tilde{\psi}_n[K,t]k_i^2 e^{i\sum_{m=1}^{N} k_m\theta_m}\right] \\
=&-\hbar\omega_\theta\sum_{n,i}^{N}\left[\sum_{-\infty\ N\ sums}^{+\infty} k_i^2\tilde{\psi}_n[K,t]\frac{1}{(2\pi)^N}\int_0^{2\pi} d\Theta e^{i\sum_{m=1}^{N}(k_m-k_m^{'})\theta_m}\right]
\end{aligned}
$$

We know that $\delta(x-x^{'}) = \frac{1}{2\pi}\int_0^{2\pi} dt e^{i(x-x^{'})t}$

$$
= -\hbar\omega_\theta\sum_{n,i}^{N}\left[\sum_{-\infty\ N\ sums}^{+\infty} k_i^2\tilde{\psi}_n[K,t]\delta(k_1-k_1^{'})\cdots\delta(k_N-k_N^{'})\right]
$$

We get the Fourier transform of **Term I** to be

$$
-\hbar\omega_\theta\sum_{n=1}^{N}\sum_{i=1}^{N} k_i^{'2} \tag{A 1}
$$

Term II

Fourier transform of $\sum_{\substack{i,j=1\\i>j}}^{N}\sum_{n=1}^{N} U_{ij}\psi_n[\Theta,t]$ is given by

$$\frac{1}{(2\pi)^N}\int_0^{2\pi} d\Theta e^{-i\sum_{m=1}^{N}k_m'\theta_m}\sum_{\substack{i,j=1\\i>j}}^{N}\sum_{n=1}^{N}U_{ij}\left(\sum_{-\infty\ sums}^{+\infty}\tilde{\psi}_n[K,t]e^{i\sum_{m=1}^{N}k_m\theta_m}\right)$$

$$=\frac{1}{(2\pi)^N}\int_0^{2\pi} d\Theta e^{-i\sum_{m=1}^{N}k_m'\theta_m}$$

$$\sum_{\substack{i,j=1\\i>j}}^{N}\sum_{n=1}^{N}u_{ij}\left[e^{i(\theta_i-\theta_j)}+e^{-i(\theta_i-\theta_j)}\right]\left(\sum_{-\infty\ sums}^{+\infty}\tilde{\psi}_n[K,t]e^{i\sum_{m=1}^{N}k_m\theta_m}\right)$$

$$=\sum_{\substack{i,j=1\\i>j}}^{N}\sum_{n=1}^{N}\frac{u_{ij}}{(2\pi)^N}\int_0^{2\pi} d\Theta e^{-i\sum_{m=1}^{N}k_m'\theta_m}\left[e^{i(\theta_i-\theta_j)}+e^{-i(\theta_i-\theta_j)}\right]$$

$$\left(\sum_{-\infty\ sums}^{+\infty}\tilde{\psi}_n[K,t]e^{i\sum_{m=1}^{N}k_m\theta_m}\right)$$

$$=\sum_{\substack{i,j=1\\i>j}}^{N}\sum_{n=1}^{N}\frac{u_{ij}}{(2\pi)^N}\int_0^{2\pi} d\Theta$$

$$\left(\sum_{-\infty\ sums}^{+\infty}\tilde{\psi}_n[K,t]e^{-i\sum_{m=1}^{N}k_m'\theta_m}\left[e^{i(\theta_i-\theta_j)}+e^{-i(\theta_i-\theta_j)}\right]e^{i\sum_{m=1}^{N}k_m\theta_m}\right)$$

$$=\sum_{\substack{i,j=1\\i>j}}^{N}\sum_{n=1}^{N}\frac{u_{ij}}{(2\pi)^N}\int_0^{2\pi} d\Theta$$

$$\left(\sum_{-\infty\ sums}^{+\infty}\tilde{\psi}_n[K,t]e^{i\sum_{m=1}^{N}(k_m-k_m')\theta_m}\left[e^{i(\theta_i-\theta_j)}+e^{-i(\theta_i-\theta_j)}\right]\right)$$

$$=\sum_{\substack{i,j=1\\i>j}}^{N}\sum_{n=1}^{N}\frac{u_{ij}}{(2\pi)^N}\int_0^{2\pi} d\Theta\left(\sum_{-\infty\ sums}^{+\infty}\tilde{\psi}_n[K,t]\right.$$

$$\left.\left[e^{i\left(\sum_{m=1}^{N}(k_m-k_m')\theta_m\right)+(\theta_i-\theta_j)}+e^{i\left(\sum_{m=1}^{N}(k_m-k_m')\theta_m\right)-(\theta_i-\theta_j)}\right]\right)$$

$$=u_{ij}\sum_{\substack{i,j=1\\ i>j}}^{N}\sum_{n=1}^{N}\left(\sum_{\substack{-\infty\\ N\ sums}}^{+\infty}\tilde{\psi}_n[K,t]\frac{1}{(2\pi)^N}\int_0^{2\pi}d\Theta\right.$$
$$\left.\left[e^{i(k_1-k_1^{'})\theta_1}\cdots e^{i(k_N-k_N^{'})\theta_N}e^{i(\theta_i-\theta_j)}+e^{i(k_1-k_1^{'})\theta_1}\cdots e^{i(k_N-k_N^{'})\theta_N}e^{-i(\theta_i-\theta_j)}\right]\right)$$

Let $\delta_i^+=\dfrac{1}{2\pi}\displaystyle\int_0^{2\pi}d\theta_i e^{i\theta_i}e^{i(k_i-k_i^{'})\theta_i}$ and $\delta_j^-=\frac{1}{2\pi}\int_0^{2\pi}d\theta_j e^{-i\theta_j}e^{i(k_j-k_j^{'})\theta_j}$.
Also we condsider $K^{'}(x^+,y^-)=(k_1^{'},k_2^{'},...k_x^{+^{'}}...k_y^{+^{'}}...k_N^{'})$.

$$=u_{ij}\sum_{\substack{i,j=1\\ i>j}}^{N}\sum_{n=1}^{N}\sum_{\substack{-\infty\\ N\ sums}}^{+\infty}\tilde{\psi}_n[K,t]\left[\delta_1...\delta_i^+...\delta_j^-....\delta_N+\delta_1....\delta_i^-...\delta_j^+...\delta_N\right]$$

We get the Fourier transform of the **Term II**

$$u_{ij}\sum_{\substack{i,j=1\\ i>j}}^{N}\sum_{n=1}^{N}\left[\tilde{\psi}_n[K^{'}(i^+,j^-),t]+\tilde{\psi}_n[K^{'}(j^+,i^-),t]\right] \qquad \text{(A 2)}$$

Term III

$$=\frac{1}{(2\pi)^N}\int_0^{2\pi}d\Theta e^{-i\sum_{m=1}^{N}k_m^{'}\theta_m}$$
$$\sum_{\substack{i=1\\ i\neq n}}^{N}\sum_{n=1}^{N}V_{ij}\left(\sum_{\substack{-\infty\\ N\ sums}}^{+\infty}\tilde{\psi}_n[K,t]e^{i\sum_{m=1}^{N}k_m\theta_m}\right)$$
$$=\frac{1}{(2\pi)^N}\int_0^{2\pi}d\Theta e^{-i\sum_{m=1}^{N}k_m^{'}\theta_m}$$
$$\sum_{\substack{i=1\\ i\neq n}}^{N}\sum_{n=1}^{N}\frac{V_o}{2}\left[e^{i(\theta_i-\theta_n)}+e^{-i(\theta_i-\theta_n)}\right]\left(\sum_{\substack{-\infty\\ N\ sums}}^{+\infty}\tilde{\psi}_n[K,t]e^{i\sum_{m=1}^{N}k_m\theta_m}\right)$$

Similar to the simplification of in the previous term, we get the Fourier transform of the **term III**

$$u_{ij}\sum_{\substack{i=1\\ i\neq n}}^{N}\sum_{n=1}^{N}\left[\tilde{\psi}_n[K^{'}(i^+,n^-),t]+\tilde{\psi}_n[K^{'}(n^+,i^-),t]\right] \qquad \text{(A 3)}$$

Term IV

$$\begin{aligned}
=&\frac{1}{(2\pi)^N}\int_0^{2\pi}d\Theta e^{-i\sum_{m=1}^{N}k_m^{'}\theta_m}\hbar\omega_\varepsilon\sum_{n=1}^{N}\psi_n[\Theta,t]\\
=&\frac{\hbar\omega_\varepsilon}{(2\pi)^N}\int_0^{2\pi}d\Theta e^{-i\sum_{m=1}^{N}k_m^{'}\theta_m}\sum_{n=1}^{N}\underbrace{\sum_{-\infty}^{+\infty}}_{N\ sums}\tilde{\psi}_n[K,t]e^{i\sum_{m=1}^{N}k_m\theta_m}\\
=&\frac{\hbar\omega_\varepsilon}{(2\pi)^N}\sum_{n=1}^{N}\underbrace{\sum_{-\infty}^{+\infty}}_{N\ sums}\int_0^{2\pi}d\Theta\tilde{\psi}_n[K,t]e^{i\sum_{m=1}^{N}(k_m-k_m^{'})\theta_m}\\
=&\hbar\omega_\varepsilon\sum_{n=1}^{N}\underbrace{\sum_{-\infty}^{+\infty}}_{N\ sums}\tilde{\psi}_n[K,t]\left(\frac{1}{2\pi}\int_0^{2\pi}d\theta_1 e^{i(k_1-k_1^{'})\theta_1}\cdots\frac{1}{2\pi}\int_0^{2\pi}d\Theta e^{i(k_N-k_N^{'})\theta_N}\right)\\
=&\hbar\omega_\varepsilon\sum_{n=1}^{N}\underbrace{\sum_{-\infty}^{+\infty}}_{N\ sums}\tilde{\psi}_n[K,t]\delta_1\delta_2.....\delta_N
\end{aligned}$$

We get the Fourier transform of **Term IV**

$$\hbar\omega_\varepsilon\sum_{n=1}^{N}\tilde{\psi}_n[K^{'},t]$$

Therefore, the time-dependent Schrödinger on in the momentum space is as follows:

$$
\begin{aligned}
i\hbar\frac{\partial}{\partial t}\sum_{n=1}^{N}\tilde{\psi}_n[K^{'},t]|n\rangle = &-\hbar\omega_\theta\sum_{n=1}^{N}\sum_{i=1}^{N}\left(k_i^{'2}\tilde{\psi}_n[K^{'},t]\right)|n\rangle \\
&+u_{ij}\sum_{\substack{i,j=1\\ i>j}}^{N}\sum_{n=1}^{N}\left[\tilde{\psi}_n[K^{'}(i^+,j^-),t]+\tilde{\psi}_n[K^{'}(j^+,i^-),t]\right]|n\rangle \\
&+v_{ij}\sum_{\substack{i=1\\ i\neq n}}^{N}\sum_{n=1}^{N}\left[\tilde{\psi}_i(K^{'}[i^+,n^-),t]+\tilde{\psi}_i[K^{'}(n^+,i^-),t]\right]|n\rangle \\
&+\hbar\omega_\varepsilon\sum_{n=1}^{N}\tilde{\psi}_n[K^{'},t]|n\rangle
\end{aligned}
$$

where all $k_i^{'}s \in \mathbb{Z}$ i.e. $k_i^{'} \in (-\infty,\infty)$

B

Python Codes

In our work we have extensively used Python as a tool to carry out our numerical calculations. Programming in python presents itself as an ideal tool to any theoretician with its versatile data structures and extensively supported in-built library function from modules such as numpy, scipy, pandas and matplotlib. As a part of the appendix chapter we present various functions and programs that have been written in Python for this dissertation work.

B.1 Ordered_chain_calculation.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 24 22:06:18 2019

@author: Vijay
"""

import numpy as np
```

```
10 import matplotlib.pyplot as plt
11 import pandas as pd
12 import time as timer
13 def range1(n):
14     return range(1,n+1)
15 def E(k,se,V,n):
16     return se*(n-2) + V*np.cos((k*np.pi)/(n+1))
17 def c_t(t,se,V,n):
18     sites = range1(n+1)
19     c = 0
20     for k in sites:
21         c += ((2/(n+1))*np.sin((k*np.pi)/(n+1))
22               * np.sin((k*n*np.pi)/(n+1))
23               * np.exp(-1j*E(k,se,V,n)*t))
24     return c
25 def s_t(t,se,V,n):
26     sites = range1(n+1)
27     s = 0
28     for k in sites:
29         s += ((2/(n+1))*(np.sin((k*np.pi)/(n+1)))**2
30                   * np.exp(-1j*E(k,se,V,n)*t))
31     return s
32 def calculation(time_range,time_steps,n,V,se):
33     start = timer.time()
34     temp_dict1= dict()
35     temp_dict2= dict()
36     time= np.linspace(0,time_range,time_steps)
37     for t in time:
38         val1 = abs(c_t(t,se,V,n))**2
39         temp_dict1[t]=val1
40     for t in time:
41         val2 = abs(s_t(t,se,V,n))**2
42         temp_dict2[t]=val2
43 
44     lists1 = sorted(temp_dict1.items())
45     # sorted by key, return a list of tuple
46     x1, y1 = zip(*lists1)
47     # unpack a list of pairs into two tuples
```

```
48      lists2 = sorted(temp_dict2.items())
49      x2, y2 = zip(*lists2)
50
51      trans_prob = dict()
52      trans_prob['time'] = x1
53      trans_prob['trans_prob'+str(n)+'sites'] = y1
54      file_name1 = ('NDis_'+str(n)+'sites'
55                    +str(time_steps)+'_ts'
56                    +str(time_range)+'time_range')
57
58      survival_prob = dict()
59      survival_prob['time'] = x2
60      survival_prob['survival_prob'+str(n)+'sites'] = y2
61      file_name2 = ('NDis_'+str(n)+'survival_prob'+
62                     'sites'+str(time_steps)
63                    +'_ts'+str(time_range)+'time_range')
64
65      df1 = pd.DataFrame.from_dict(trans_prob)
66      df1.to_csv(file_name1 +
67                 '.dat', sep='\t',
68                 encoding='utf-8')
69
70      df2 = pd.DataFrame.from_dict(survival_prob)
71      df2.to_csv(file_name2 +
72                 '.dat', sep='\t',
73                 encoding='utf-8')
74
75      plt.figure()
76      plt.plot(x1, y1,color='red')
77      plt.ylabel(r"$|U_{N,1}(t)|^{2}$",
78                 fontsize=16, color='black')
79      plt.title(str(n)+ r" sites",
80                fontsize=16, color='black')
81      plt.xlabel(r"Time (a.u.)",
82                 fontsize=16, color='black')
83      plt.savefig(file_name1+
84                  '.png', dpi=1000)
85
```

```
86
87      plt.figure()
88      plt.plot(x2, y2,color='red')
89      plt.ylabel(r"$|S_1(t)|^{2}$",
90                 fontsize=16, color='black')
91      plt.title(str(n)+ r" sites",
92                fontsize=16, color='black')
93      plt.xlabel(r"Time (a.u.)",
94                 fontsize=16, color='black')
95      plt.savefig(file_name2+'.png', dpi=1000)
96      end = timer.time()
97      print(file_name1,file_name2)
98      print(start-end)
99
100
101 calculation(500,1000,32,V=1,se=1)
```

B.2 Defects_calculation.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 24 21:56:49 2019

@author: rimm
"""
import numpy as np
import itertools as it
import matplotlib.pyplot as plt
import pandas as pd
def point_defect(N):
    matrix = np.zeros((N,N))
    angles = np.random.uniform(0,2*np.pi,size=N)
    for (i,j) in it.product(range(N),range(N)):
        if i==j+1 or i==j-1:
                if i== N/2-1 and j== N/2  :
                    matrix[i][j] = np.cos(angles[i]-angles[j])
                elif j== N/2-1 and i== N/2  :
                    matrix[i][j] = np.cos(angles[i]-angles[j])
                else:
                    matrix[i][j] = 1
    return(matrix)
def line_defect(N):
    matrix = np.zeros((N,N))
    angles = np.random.uniform(0,2*np.pi,size=N)
    for (i,j) in it.product(range(N),range(N)):
        if i==j+1 or i==j-1:
                if i== int(N/2):
                    matrix[i][j] = np.cos(angles[i]-angles[j])
                elif i== int(N/2) or j== int(N/2):
                    matrix[i][j] = np.cos(angles[i]-angles[j])
                else:
                    matrix[i][j] = 1
    return(matrix)
```

```
36
37 def chemical_defect(N, defect):
38     matrix = np.zeros((N,N))
39     for (i,j) in it.product(range(N),range(N)):
40         if i==j+1 or i==j-1:
41                     matrix[i][j] = 1
42         elif i== int(N/2) and j==int(N/2):
43             matrix[i][j] = defect
44     return(matrix)
45 def transition_probability_defects(N,
46                                    vals,vecs,
47                                    time_step,
48                                    end_time,se = 1,V = 1):
49     dim = N
50     time = np.linspace(0,end_time,time_step)
51     ts = lambda t: sum([(vec[0]*vec[dim-1]
52                          *np.exp(-1j*((-se*(dim-2)/2)
53                             + V*val)*t) ) for vec,val in zip(vecs,vals)])
54
55     trans_prob = [(abs(ts(i)))**2 for i in time]
56     #lists = sorted(ts_dict.items())
57     #time, trans_prob = zip(*lists)
58     return(time, trans_prob)
59 def survival_probability_defects(N,vals,vecs,
60                                  time_step,
61                                  end_time,
62                                  se = 1,V = 1):
63     dim = N
64     time = np.linspace(0,end_time,time_step)
65     ss = lambda t: sum([(vec[0]**2
66                          *np.exp(-1j*((-se*(dim-2)/2)
67                            + V*val)*t) )
68                         for vec,val in zip(vecs,vals)])
69
70     survival_prob = [(abs(ss(i)))**2 for i in time]
71     #lists = sorted(ts_dict.items())
72     #time, trans_prob = zip(*lists)
73     return(time, survival_prob)
```

```
74 def chemical_defect_calculation(N,
75                                  sample_range,
76                                  time_step,end_time,
77                                  se = 1,V = 1):
78     sampling = list()
79     random_value = np.random.uniform(0,10,size=1)
80     a = np.zeros(time_step)
81     c = np.zeros(time_step)
82     for i in range(1,sample_range+1) :
83         matrix = chemical_defect(N,random_value[0])
84         vals, vecs = np.linalg.eigh(matrix)
85         z = 0
86         for (i,j) in it.product(range(N),range(N)):
87                 if vals[i] == vals[j] :
88                     z += vecs[i][1]*vecs[i][N-1]*
89                     np.transpose(vecs[j][1])
90                     * np.transpose(vecs[j][N-1])
91         sampling.append(z)
92         time, trans_prob = transition_probability_defects(N
93                                                           ,vals,vecs,
94                                                           time_step,
95                                                           end_time)
96         time, survival_prob = survival_probability_defects(N,
97                                                            vals,vecs,
98                                                            time_step,
99                                                            end_time)
100        a += trans_prob
101        c += survival_prob
102    average_t  = a/sample_range
103    average_s  = c/sample_range
104    trans_prob=dict()
105    t_avg = sum(sampling)/sample_range
106    survival_prob =dict()
107    x  = time
108    y1 = average_t
109    y2 = average_s
110
111    trans_prob = dict()
```

```
112     trans_prob['time'] = x
113     trans_prob['trans_prob'+str(N)+'sites'] = y1
114     trans_prob['survival_prob'+str(1)+'sites'] = y2
115     file_name1 = ('chemical_defect_'+str(N)+'sites'+str(1000)+'_ts'
116                   +str(sample_range)+'sample_range_'
117                   +str(time_step)+'timestep'
118                   +'_survival_at_'+str(1))
119     df = pd.DataFrame.from_dict(trans_prob)
120     df.to_csv(file_name1 +'.dat', sep='\t', encoding='utf-8')
121     plt.figure()
122     plt.plot(x, y1,color='red')
123     plt.ylabel(r"$|\mathcal{U}_{N,1}(t)|^{2}$",
124                fontsize=16, color='black')
125     plt.title(str(N)+ r" sites with Chemical defect",
126               fontsize=16, color='black')
127     plt.xlabel(r"Time (a.u.)", fontsize=16, color='black')
128     plt.savefig(file_name1+'.png', dpi=1000)
129     plt.figure()
130     print(t_avg)
131 def point_defect_calculation(N,sample_range,time_step,se = 1,V = 1):
132     sampling = list()
133     a = np.zeros(time_step)
134     c = np.zeros(time_step)
135     for i in range(1,sample_range+1) :
136         matrix = point_defect(N)
137         vals, vecs = np.linalg.eigh(matrix)
138         z = 0
139         for (i,j) in it.product(range(N),range(N)):
140                 if vals[i] == vals[j] :
141                     z += vecs[i][1]*vecs[i][N-1]
142                     * np.transpose(vecs[j][1])
143                     * np.transpose(vecs[j][N-1])
144         sampling.append(z)
145         time, trans_prob = transition_probability_defects(N,
146                                                           vals,vecs,
147                                                           time_step,100)
148
149
```

```
150
151         time, survival_prob = survival_probability_defects(N,
152                                                            vals,vecs,
153                                                            time_step,100)
154         a += trans_prob
155         c += survival_prob
156     average_t  = a/sample_range
157     average_s  = c/sample_range
158     trans_prob=dict()
159     t_avg = sum(sampling)/sample_range
160     survival_prob =dict()
161     x  = time
162     y1 = average_t
163     y2 = average_s
164
165     trans_prob = dict()
166     trans_prob['time'] = x
167     trans_prob['trans_prob'+str(N)+'sites'] = y1
168     trans_prob['survival_prob'+str(1)+'sites'] = y2
169     file_name1 = ('Defect_Dis_V'+str(N)+'sites'+str(1000)+'_ts'
170                   +str(sample_range)+'sample_range_'
171                   +str(time_step)+'timestep'
172                   +'_survival_at_'+str(1))
173     df = pd.DataFrame.from_dict(trans_prob)
174     df.to_csv(file_name1 +'.dat', sep='\t', encoding='utf-8')
175     plt.figure()
176     plt.plot(x, y1,color='red')
177     plt.ylabel(r"$|\mathcal{U}_{N,1}(t)|^{2}$",
178                fontsize=16, color='black')
179     plt.title(str(N)+ r" sites with Point defect",
180               fontsize=16, color='black')
181     plt.xlabel(r"Time (a.u.)", fontsize=16, color='black')
182     plt.savefig(file_name1+'.png', dpi=1000)
183     plt.figure()
184     print(t_avg)
185
186 def line_defect_calculation(N,sample_range,time_step,se = 1,V = 1):
187     sampling = list()
```

```
188     a = np.zeros(time_step)
189     c = np.zeros(time_step)
190     for i in range(1,sample_range+1) :
191         matrix = point_defect(N)
192         vals, vecs = np.linalg.eigh(matrix)
193         z = 0
194         for (i,j) in it.product(range(N),range(N)):
195                 if vals[i] == vals[j] :
196                     z += vecs[i][1]*vecs[i][N-1]
197                     * np.transpose(vecs[j][1])
198                     * np.transpose(vecs[j][N-1])
199         sampling.append(z)
200         time, trans_prob = transition_probability_defects(N,
201                                                           vals,vecs,
202                                                           time_step,200)
203         time, survival_prob = survival_probability_defects(N,
204                                                            vals,vecs,
205                                                            time_step,200)
206         a += trans_prob
207         c += survival_prob
208     average_t  = a/sample_range
209     average_s  = c/sample_range
210     trans_prob=dict()
211     t_avg = sum(sampling)/sample_range
212     survival_prob =dict()
213     x  = time
214     y1 = average_t
215     y2 = average_s
216
217     trans_prob = dict()
218     trans_prob['time'] = x
219     trans_prob['trans_prob'+str(N)+'sites'] = y1
220     trans_prob['survival_prob'+str(1)+'sites'] = y2
221     file_name = ('Defect_V'+str(N)+'sites'+str(1000)+'_ts'
222                  +str(sample_range)+'sample_range_'
223                  +str(time_step)+'timestep')
224     df = pd.DataFrame.from_dict(trans_prob)
225     df.to_csv(file_name +'.dat', sep='\t', encoding='utf-8')
```

```
226     plt.figure()
227     plt.plot(x, y1,color='red')
228     plt.ylabel(r"$|\mathcal{}_{N,1}(t)|^{2}$",
229                fontsize=16, color='black')
230     plt.title(str(N)+ r" sites with Point defect",
231               fontsize=16, color='black')
232     plt.xlabel(r"Time (a.u.)", fontsize=16, color='black')
233     plt.savefig(file_name+'.png', dpi=1000)
234     print(t_avg)
235 #-------------------------------------------------------------------#
236
237 line_defect_calculation(N,sample_range,time_step)
238 chemical_defect_calculation(N,sample_range,time_step
239                                 ,end_time,se = 1,V = 1)
240 line_defect_calculation(N,sample_range,end_time)
```

B.3 Static_disorder.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 24 21:45:05 2019

@author: Vijay
"""
import numpy as np
import itertools as it
import matplotlib.pyplot as plt
import time
import pandas as pd

def potential_matrix(dim,angles,geometry=None):
    """ Generates a nxn square matrix with diagonal elements
    to be zero and off-diagonal elements as Cosine functions
    of values (indexed) from an array angles.

    The parameter dim is the dimensionality of
    the square matrix, angles is an array that contains
    same number of elements as the dimesion of the matrix
    and the argument geometry specifies the arrangment of
    the system. By default it is set to None and generates
    a null matrix if no argument is specified for geometry.
    """
    rows=[]
    for i in range(1,dim+1):
        row = []
        for j in range(1,dim+1):
            if i==j:
                row.append(0)
            elif geometry == 'linear':
                if i==j+1 or j==i+1:
                    row.append(np.cos(angles[i-1]-angles[j-1]))
                else:
```

```
36                      row.append(0)
37              elif geometry == 'cyclic':
38                  if i==j+1 or j==i+1:
39                      row.append(np.cos(angles[i-1]-angles[j-1]))
40                  elif (i==dim and j==1 ) or (i==1 and j==dim ) :
41                      row.append(np.cos(angles[i-1]-angles[j-1]))
42                  else:
43                      row.append(0)
44              elif geometry == 'close packed':
45                  row.append(np.cos(angles[i-1]-angles[j-1]))
46              else:
47                  row.append(0)
48          rows.append(row)
49      return np.asanyarray(rows)
50  def fast_pmatrix(N):
51      matrix = np.zeros((N,N))
52      angles = np.random.uniform(0,2*np.pi,size=N)
53      for (i,j) in it.product(range(N),range(N)):
54          if i==j+1 or i==j-1:
55              matrix[i][j] = np.cos(angles[i]-angles[j])
56
57      return(matrix)
58  def gen_sample_space(N, sample_range, geometry =None,
59                       matrices=False,angles =False):
60      """ Generates a sample space(list) of NxN matrices
61         sampled over a set of randomly generated angles
62         for the cosine functions in the potential matirx.
63        sample_range(dtype= int()) specifies the number
64        times the potential matrix is sampled over randomly
65        generated angles. The arguments geometry is same as
66        that for the function potential_matrix
67        matrices (by default=False) arg** is specified
68        if the matrices generates are (=True)to be printed.
69        angles (by default=False) arg** is specified to
70        print the set of arrays of randomly generated angles
71        for each matrix in the sample space. he arrays are
72        by default stored as a list angles_sample.  """
73
```

```
74      sample_set = range(1,sample_range+1)
75      sample_space= list()
76      angles_sample = list()
77      for j in sample_set:
78          random_angles = np.random.uniform(0,2*np.pi,size=N)
79          angles_sample.append(random_angles)
80          matrix = potential_matrix(N,
81                                    random_angles ,geometry)
82          sample_space.append(matrix)
83          if matrices == True:
84              print(j)
85              print(matrix)
86      return(sample_space)
87
88  def tavg_transition_probability(N,sample_range,
89                                  se = 1,V = 1):
90      sampling = list()
91      for i in range(1,sample_range+1) :
92          matrix = fast_pmatrix(N)
93          vals, vecs = np.linalg.eigh(matrix)
94          z = 0
95          for (i,j) in it.product(range(N),range(N)):
96                  if vals[i] == vals[j] :
97                      z += vecs[i][1]*vecs[i][N-1]*
98                      np.transpose(vecs[j][1]) *
99                      np.transpose(vecs[j][N-1])
100         sampling.append(z)
101     t_avg = sum(sampling)/sample_range
102     return(t_avg)
103
104
105 def transition_probability(N,vecs,vals,
106                            time_step,
107                            end_time,se = 1,V = 1):
108
109     dim = N
110     time = np.linspace(0,end_time,time_step)
111     ts = lambda t: sum([(vec[0]*vec[dim-1]
```

```
112                     *np.exp(-1j*((-se*(dim-2)/2)
113                                  + V*val)*t) )
114                      for vec,val in zip(vecs,vals)])
115
116     trans_prob = [(abs(ts(i)))**2 for i in time]
117     #lists = sorted(ts_dict.items())
118     #time, trans_prob = zip(*lists)
119     return(time, trans_prob)
120 def survival_probability(site,
121                              potential_matrix,
122                              time_step,end_time
123                              ,se = 1,V = 1):
124     vals, vecs = np.linalg.eigh(potential_matrix)
125
126
127     dim = potential_matrix.shape[0]
128     index= range(1,dim+1)
129     time = np.linspace(0,end_time,time_step)
130     ss = lambda t: sum([(vec[site]**2 *
131                              np.exp(-1j*((-se*(dim-2)/2)
132                                + V*val)*t) )
133                         for vec,val in zip(vecs,vals)])
134
135     survival_prob = [(abs(ss(i)))**2 for i in time]
136     #lists = sorted(ts_dict.items())
137     #time, trans_prob = zip(*lists)
138     return(time, survival_prob)
139 def calculation(N,sample_range,
140                     time_step,
141                     end_time,
142                     survival_site):
143     start = time.time()
144
145     my_space = gen_sample_space(N,sample_range,
146                                         geometry='linear')
147     a = np.zeros(time_step)
148     c = np.zeros(time_step)
149     intd = time.time()
```

```
150     print(intd-start)
151     for m in my_space:
152         t ,transprob = transition_probability(m,
153                                               time_step,
154                                               end_time)
155         t,survivalprob = survival_probability(survival_site,
156                                               m,time_step,
157                                               end_time)
158         a+= transprob
159         c+= survivalprob
160     average_t  = a/len(my_space)
161     average_s  = c/len(my_space)
162     trans_prob=dict()
163     survival_prob =dict()
164     x  = t
165     y1 = average_t
166     y2 = average_s
167
168     end = time.time()
169     print(end-start)
170     trans_prob['time'] = x
171     trans_prob['trans_prob'+str(N)+'sites'] = y1
172     trans_prob['survival_prob'+str(survival_site)+'sites'] = y2
173     file_name = ('St_Dis_V'+str(N)
174                  +'sites'+str(1000)+'_ts'
175                  +str(sample_range)+'sample_range_'
176                  +str(time_step)+'timestep')
177     file_name1 = ('St_Dis_V'+str(N)+'sites'
178                   +str(1000)+'_ts'
179                  +str(sample_range)
180                  +'sample_range_'
181                  +str(time_step)
182                  +'timestep'+'_survival_at_'
183                  +str(survival_site))
184     df = pd.DataFrame.from_dict(trans_prob)
185     df.to_csv(file_name +'.dat', sep='\t', encoding='utf-8')
186     plt.figure()
187     plt.plot(x, y2,color='red')
```

```
188     plt.ylabel(r"$|S_n(t)|^{2}$", fontsize=16, color='black')
189     plt.title(str(N)+ r" sites", fontsize=16, color='black')
190     plt.xlabel(r"Time (a.u.)", fontsize=16, color='black')
191     plt.savefig(file_name1+'.png', dpi=1000)
192 
193     plt.figure()
194     plt.plot(x, y1,color='red')
195     plt.ylabel(r"$|\mathcal{U}_{N,1}(t)|^{2}$",
196                fontsize=16, color='black')
197     plt.title(str(N)+ r" sites",
198               fontsize=16, color='black')
199     plt.xlabel(r"Time (a.u.)",
200                fontsize=16, color='black')
201     plt.savefig(file_name+'.png', dpi=1000)
202     return(my_space,t)
```

B.4 Dynamic_disorder.py

```
import itertools as it
import pandas as pd
import numpy as np
import time
import matplotlib.pyplot as plt

class Dynamic_disorder:
    def __init__(self,N,M):
        self.N = N
        self.M = M
    def inverse(self,tup,dictionary):
        # Takes the basis state to return its key which is the index
        try:
            key_value = (list(dictionary.keys())
                         [list(dictionary.values()).index(tup)])
            return(key_value)
        except ValueError:
            pass

    def op1(self,tup):
        # It is corresponds to the first term  of the Hamiltonian i.e
        # kinetic term of the rotor in fourier space
        sum_k = sum([tup[i]**2 for i in self.sites])
        return(sum_k)

    def op2(self,tup):
        # op2 is the rotor coupling operator
        # returns list of tuples generated by
        # the rotor coupling term u(\theta)
        rc_1 = list() # rotor coupling term 1
        rc_2 = list() # rotor coupling term 2
        tup_list = list(tup)
        for i in self.sites:
            for j in self.sites:
                if j>i:
```

```
36                      r1 = tup_list.copy() # rotor term 1
37                      r2 = tup_list.copy() # rotor term 2
38                      #print(i,j)
39                      r1[i]+=1
40          # raises k_i momentum state by +1 changes in \psi_{tilda}
41                      r1[j]-=1
42          # lowers k_j momentum state by +1 changes in \psi_{tilda}
43
44                      r2[j]+=1
45                      r2[i]-=1
46                      #print(tuple(r1),tuple(r2))
47                      rc_1.append(tuple(r1))
48        #i>j summation that works fines with operation at two sites
49                      rc_2.append(tuple(r2))
50          return(rc_1,rc_2)
51
52
53
54
55
56
57      def op3(self,tup):
58          # op3 is the exciton-rotor coupling operator
59          # returns list of tuples generated by
60          #  the rotor coupling term V(\theta)
61          exrc_1 = list() # exciton-rotor coupling term 1
62          exrc_2 = list() # exciton-rotor coupling term 2
63          i = tup[0]
64          for n in self.sites:
65              tup_list = list(tup)
66              #print(tup)
67              if n!=i:
68                  tup_list[0] = n
69                  er1 = tup_list.copy() #exciton rotor coupling term 1
70                  er2 = tup_list.copy() #exction rotor coupling term 2
71                  #print(i,n)
72                  er1[i]+=1
73          # raises k_i momentum state by +1 changes in \psi_{tilda}
```

```
74                  er1[n]-=1
75          # lowers k_n momentum state by +1 changes in \psi_{tilda}
76
77                  er2[i]-=1
78          # lowers k_i momentum state by +1 changes
79                  er2[n]+=1
80          # raises k_n momentum state by +1 changes
81                  #print(er1,er2)
82                  exrc_1.append(tuple(er1))
83          #i>j summation that works fines with operation at two sites
84                  exrc_2.append(tuple(er2))
85           return(exrc_1,exrc_2)
86
87      def append_to_row(self,lst,row,v):
88          for i in lst:
89              c = self.inverse(i,self.basis)
90              try:
91                  row[c-1] += v
92              except TypeError:
93                  pass
94      #-------------------------------------------------------------------
95      def basis_gen(self):
96      # generates a dictionary of indexed tuples.
97          iterable = list(range(-self.M, self.M + 1))
98      # generates all possible momentum states for all k in [-N,N]
99          momentum_states =list(it.product(iterable,repeat=self.N))
100         momentum_sub_spaces = dict()
101         #print(len(momentum_states))
102
103         self.sites = range(1,self.N + 1)
104
105         # In the following loop a (2M+1)^N tuples (k1,k2,k3,..kn)
106         # are taken from momentum_states and
107         #segregated into momentum subspaces.
108         for i in momentum_states:
109             if sum(i) in momentum_sub_spaces:
110                 if i not in momentum_sub_spaces[(sum(i))]:
111                     momentum_sub_spaces[(sum(i))].append(i)
```

```
112             else:
113                 momentum_sub_spaces[(sum(i))] = [i]
114 
115         index = 0
116         self.basis=dict()
117         final_states =[]
118         for f in self.sites:
119             for i in momentum_sub_spaces:
120                 for j in momentum_sub_spaces[i]:
121                     index+=1
122                     self.basis[index] = (f,)+j
123                     if f== self.N:
124                         final_states.append((f,)+j )
125 
126         # table has information of the segregated basis states
127         table = pd.DataFrame(dict([ (k,pd.Series(v))
128                         for k,v in momentum_sub_spaces.items() ]))
129         return(final_states,table.transpose())
130 
131     #-----------------------------------------------------------------------
132 
133     def time_func_trans_prob(self,t):
134         vecs = self.eigvecs
135         vals = self.eigvals
136         c =0
137         for vec,val in zip(vecs,vals):
138                 c += (vec[self.int_index])*sum([vec[i-1]
139                      for i in self.final_indices])
140                      * np.exp(-1j*(val * t))
141         ts = abs(c)**2
142         return(ts)
143     def time_func_survival_prob(self,t):
144         vecs = self.eigvecs
145         vals = self.eigvals
146         cc =0
147         for vec,val in zip(vecs,vals):
148                 cc += (vec[self.int_index])**2
149                       * np.exp(-1j*(val * t))
```

```
150         ss = abs(cc)**2
151         return(ss)
152     def transition_probability(self,final_states,
153                                end_time,time_step):
154         N = self.N
155         M = self.M
156         time_list = np.linspace(0,end_time,time_step)
157         zero_state = [0]*N
158         int_state = ((1,)+ tuple(zero_state))
159         self.int_index = self.inverse(int_state,self.basis)
160         self.final_indices = [self.inverse(i,self.basis)
161                                             for i in final_states]
162 
163         vec_time_function1 = np.vectorize(self.time_func_trans_prob,
164                                                       otypes=[float])
165         trans_prob = list(vec_time_function1(time_list))
166 
167         file_name = ('trial_Dyn_Dis_'+ str(N)+'_sites_'
168                             +str((2*M + 1))
169                             +'_DOF_'+str(time_step)+'_ts_'
170                             +str(end_time)+'time_range')
171         self.int_time3 =  time.time()
172         print("For timerange "+str(end_time)+
173               " calculation time is "+
174               (str(self.int_time3- self.int_time2)))
175         return(time_list,trans_prob,time_step,end_time,file_name)
176 
177     def survival_probability(self,end_time,time_step):
178         N = self.N
179         time_list = np.linspace(0,end_time,time_step)
180         zero_state = [0]*N
181         int_state = ((1,)+ tuple(zero_state))
182         self.int_index = self.inverse(int_state,self.basis)
183         vec_time_function2 = np.vectorize(self.time_func_survival_prob,
184                                                       otypes=[float])
185         survival_prob = list(vec_time_function2(time_list))
186 
187         file_name1 = ('Dyn_Dis_'+str(N)+'sites'+str(time_step)+'_ts'
```

```
188                 +str(time_step)+'timestep'+'_survival_at_'+str(1))
189         self.int_time3 =  time.time()
190         print("For timerange "+str(end_time)+
191               " calculation time is "+
192               (str(self.int_time3- self.int_time2)))
193         return(time_list,survival_prob,
194                time_step,end_time,file_name1)
195 
196     def save_data(self,time_list,trans_prob,
197                   time_step,end_time,file_name):
198 
199         trans_prob_dict = dict()
200         trans_prob_dict['time'] = time_list
201         trans_prob_dict['trans_prob'+str(self.N)+'sites'] = trans_prob
202         df = pd.DataFrame.from_dict(trans_prob_dict)
203         df.to_csv(file_name +'.dat', sep='\t', encoding='utf-8')
204 
205     def plot_data(self,time_list,
206                   ts,time_step,end_time,file_name):
207         plt.figure()
208         plt.plot(time_list, ts ,color='red')
209         plt.ylabel(r"$|U_N(t)|^{2}$",
210                    fontsize=16, color='black')
211         plt.title(str(self.N)+ ' sites with '+
212                   str(2*self.M +1)+
213                   r" degrees of freedom each",
214                    fontsize=16, color='black')
215         plt.xlabel(r"Time (a.u.)",
216                     fontsize=16, color='black')
217         plt.savefig(file_name+'.png', dpi=1000)
218         plt.show()
219 
220     def plot_survival_data(self,time_list,ts,
221                            time_step,end_time,file_name):
222         plt.figure()
223         plt.plot(time_list, ts ,color='red')
224         plt.ylabel(r"$|S_1(t)|^{2}$",
225                    fontsize=16, color='black')
```

```
226         plt.title(str(self.N)+ ' sites with '
227                   +str(2*self.M +1)+r" degrees of freedom each",
228                    fontsize=16, color='black')
229         plt.xlabel(r"Time (a.u.)",
230                     fontsize=16, color='black')
231         plt.savefig(file_name+'.png', dpi=1000)
232         plt.show()
233 
234     #-------------------------------------------------------------------
235 
236     def matrix_from_basis(self,m):
237         ket = self.basis[m]
238         row = np.zeros(len(self.basis))
239         k_sq = self.op1(ket)
240         row[m-1]= (self.rotor_freq*k_sq) + self.exciton_freq
241         # puts the summation k_i^2 along the diagonal
242         ra,rb = self.op2(ket)
243         ea,eb = self.op3(ket)
244         self.append_to_row(ra,row,self.v)
245         self.append_to_row(rb,row,self.v)
246         self.append_to_row(ea,row,self.u)
247         self.append_to_row(eb,row,self.u)
248         return(row)
249 
250     def calculation(self):
251         self.v=1.0
252         self.u=1.0
253         self.rotor_freq = 1.0
254         self.exciton_freq = 0
255         self.start = time.time()
256         f_states,df_momentum_space = self.basis_gen()
257 
258         self.int_time1 =  time.time()
259         print("Basis generated in "
260                +(str(self.int_time1 - self.start)))
261         # Each basis state is taken and operations
262         # are carried to build the Hamiltonian Matrix.
263         vec_matrix_from_basis = np.vectorize(self.matrix_from_basis,
```

```
264                                                 otypes=[np.ndarray])
265         self.keys = list(self.basis.keys())
266         H = list(vec_matrix_from_basis(self.keys))
267         vals, vecs = np.linalg.eigh(H)
268         #eigh returns vecs as matrix object if input is a matrix
269         self.eigvals = vals
270         self.eigvecs = vecs
271         #print(self.eigvals,self.eigvecs)
272         self.int_time2 =  time.time()
273         print("Matrix generated and diagonalised in "+
274                     (str(self.int_time2- self.int_time1)))
275         #np.set_printoptions(threshold=np.nan)
276         #print(H.shape)
277         #print(H)
278         #print(df_momentum_space)
279         #for a in [(i,basis[i]) for i in basis]:
280         #    print(a)
281         return(H,f_states)
282 #-------------------------------------------------------------------------
283 sys = Dynamic_disorder(2,2)
284 H,f_states = sys.calculation()
285 time_list,
286 trans_prob,
287 time_step,
288 end_time,
289 file_name1 = sys.transition_probability(f_states,200,1000)
290 sys.save_data(time_list,trans_prob,time_step,end_time,file_name1)
291 sys.plot_data(time_list,trans_prob,time_step,end_time,file_name1)
292 time_list,
293 survival_prob,
294 time_step,
295 end_time,
296 file_name2 = sys.survival_probability(200,1000)
297 sys.save_data(time_list,survival_prob,
298               time_step,end_time,file_name2)
299 sys.plot_survival_data(time_list,survival_prob,
300                        time_step,end_time,file_name2)
```

C

Commutation Relations

As mentioned in the chapter 3 of this thesis. The Hamiltonian for the model that we have proposed commutes with the excitation number operator $\hat{N}_{ex}$ and the total momentum operator $\hat{P}_{total}$. In this appendix chapter we prove the commutator relations between the Hamiltonian $\hat{H}_S$ and these two operators.

C.1 Excitation Number Operator $\hat{N}_{ex}$

The commutator between $\hat{H}_S$ and $\hat{N}_{ex}$ is given by

$$\left[\hat{H}_S, \hat{N}_{ex}\right] = \hat{H}_S\hat{N}_{ex} - \hat{N}_{ex}\hat{H}_S$$

We have

$$\hat{H}_S = \Bigg[\hbar\omega_\varepsilon \sum_{i=1}^{N} \sigma_i^+ \sigma_i^- - \hbar\omega_\theta \sum_{i=1}^{N} \frac{\partial^2}{\partial\theta_i^2} + \sum_{\substack{i,j \\ i>j}}^{N} 2u_{ij} \cos(2(\theta_i - \theta_j)) + \sum_{\substack{i,j \\ i\neq j}}^{N} 2v_{ij} \cos(\theta_i - \theta_j)\sigma_i^+ \sigma_j^- \Bigg]$$

and $\hat{N}_{ex} = \sum_n^N \hat{N}_n = \sum_n^N \sigma_n^+ \sigma_n^-$

$$\begin{aligned}
\hat{H}_S \hat{N}_{ex} =& \hbar\omega_\varepsilon \sum_{n}^{N}\sum_{i=1}^{N} \sigma_i^+ \sigma_i^- \sigma_n^+ \sigma_n^- - \hbar\omega_\theta \sum_{n}^{N}\sum_{i=1}^{N} \frac{\partial^2}{\partial\theta_i^2} \sigma_n^+ \sigma_n^- \\
&+ \sum_{\substack{i,j \\ i>j}}^{N}\sum_{n}^{N} 2u_{ij} \cos(2(\theta_i - \theta_j))\sigma_n^+ \sigma_n^- \\
&+ \sum_{n}^{N}\sum_{\substack{i,j \\ i\neq j}}^{N} 2v_{ij} \cos(\theta_i - \theta_j)\sigma_i^+ \sigma_j^- \sigma_n^+ \sigma_n^-
\end{aligned}$$

$$\begin{aligned}
\hat{N}_{ex} \hat{H}_S =& \hbar\omega_\varepsilon \sum_{n}^{N}\sum_{i=1}^{N} \sigma_n^+ \sigma_n^- \sigma_i^+ \sigma_i^- - \hbar\omega_\theta \sum_{n}^{N}\sum_{i=1}^{N} \sigma_n^+ \sigma_n^- \frac{\partial^2}{\partial\theta_i^2} \\
&+ \sum_{\substack{i,j \\ i>j}}^{N}\sum_{n}^{N} \sigma_n^+ \sigma_n^- 2u_{ij} \cos(2(\theta_i - \theta_j)) \\
&+ \sum_{n}^{N}\sum_{\substack{i,j \\ i\neq j}}^{N} \sigma_n^+ \sigma_n^- 2v_{ij} \cos(\theta_i - \theta_j)\sigma_i^+ \sigma_j^-
\end{aligned}$$

We know that

$$\sigma_i^{\pm} = \underbrace{\mathbb{I}_{2\times 2} \otimes \mathbb{I}_{2\times 2} \cdots \otimes \sigma^{\pm} \otimes \cdots \otimes \mathbb{I}_{2\times 2}}_{\sigma^{\pm} \text{ at } i^{\text{th}} \text{ position of a polymer chain of size } N} \qquad i \in \{1, \cdots, N\}$$

From the above we have the following relations for the Pauli spin operators:

$$\begin{aligned}
&(i) \quad [\sigma_i^+, \sigma_n^+] = 0 \\
&(ii) \quad [\sigma_i^-, \sigma_n^-] = 0 \\
&(iii) \quad [\sigma_i^+, \sigma_j^-] = \delta_{ij}(\sigma_i^+\sigma_j^- - \sigma_i^-\sigma_j^+)
\end{aligned}$$

We know that $\hat{N}_{ex}$ commutes with $\frac{\partial^2}{\partial\theta_i^2}$ and $\cos(2(\theta_i - \theta_j))$. Therefore, to prove that $\hat{N}_{ex}$ commutes with $\hat{H}_S$, it is sufficient to show that $\hat{N}_n$ commutes with $\sigma_i^+\sigma_i^-$ and $\sigma_i^+\sigma_j^-$.

Using the commutator relations

$$\begin{aligned}
[AB, C] &= A[B, C] + [A, C]B \\
[A, BC] &= [A, B]C + B[A, C]
\end{aligned}$$

we write,

$$[\sigma_i^+\sigma_i^-, \hat{N}_n] = \sigma_i^+[\sigma_i^-, \hat{N}_n] + [\sigma_i^+, \hat{N}_n]\sigma_i^-$$

$$\begin{aligned}
[\sigma_i^-, \sigma_n^+\sigma_n^-] &= [\sigma_i^-, \sigma_n^+]\sigma_n^- + \sigma_n^+[\sigma_i^-, \sigma_n^-] \\
&= -\delta_{ni}(\sigma_n^+\sigma_i^- - \sigma_i^-\sigma_n^+)\sigma_n^-
\end{aligned}$$

$$\begin{aligned}
[\sigma_i^+, \sigma_n^+\sigma_n^-] &= [\sigma_i^+, \sigma_n^+]\sigma_n^- + \sigma_n^+[\sigma_i^+, \sigma_n^-] \\
&= \sigma_n^+\delta_{ni}(\sigma_i^+\sigma_n^- - \sigma_n^-\sigma_i^+)
\end{aligned}$$

$[\sigma_i^+\sigma_i^-, \hat{N}_n] = \sigma_i^+[-\delta_{in}(\sigma_n^+\sigma_i^- - \sigma_i^-\sigma_n^+)\sigma_n^-] + [\sigma_n^+\delta_{in}(\sigma_i^+\sigma_n^- - \sigma_n^-\sigma_i^+)]\sigma_i^-$

for any i≠n, $[\sigma_i^+\sigma_i^-, \hat{N}_n] = 0$ as the δ_{in} terms vanish.

So , let us consider the case i=n

$[\sigma_n^+\sigma_n^-, \hat{N}_n] = -\sigma_n^+\sigma_n^+\sigma_n^-\sigma_n^- + \sigma_n^+\sigma_n^-\sigma_n^+\sigma_n^- + \sigma_n^+\sigma_n^+\sigma_n^-\sigma_n^- - \sigma_n^+\sigma_n^-\sigma_n^+\sigma_n^-$

$$\therefore [\sigma_i^+\sigma_i^-, \hat{N}_n] = 0$$

Similarly, we have

$$[\sigma_i^+\sigma_j^-, \hat{N}_n] = \sigma_i^+[\sigma_j^-, \hat{N}_n] + [\sigma_i^+, \hat{N}_n]\sigma_j^-$$

$$\begin{aligned}[\sigma_j^-, \sigma_n^+\sigma_n^-] &= [\sigma_j^-, \sigma_n^+]\sigma_n^- + \sigma_n^+[\sigma_j^-, \sigma_n^-] \\ &= -\delta_{nj}(\sigma_n^+\sigma_j^- - \sigma_j^-\sigma_n^+)\sigma_n^-\end{aligned}$$

$$\begin{aligned}[\sigma_i^+, \sigma_n^+\sigma_n^-] &= [\sigma_i^+, \sigma_n^+]\sigma_n^- + \sigma_n^+[\sigma_i^+, \sigma_n^-] \\ &= \sigma_n^+\delta_{ni}(\sigma_i^+\sigma_n^- - \sigma_n^-\sigma_i^+)\end{aligned}$$

$[\sigma_i^+\sigma_j^-, \hat{N}_n] = \sigma_i^+[-\delta_{nj}(\sigma_n^+\sigma_j^- - \sigma_j^-\sigma_n^+)\sigma_n^-] + [\sigma_n^+\delta_{ni}(\sigma_i^+\sigma_n^- - \sigma_n^-\sigma_i^+)]\sigma_j^-$

for n=i=j

$= -\sigma_n^+\sigma_n^+\sigma_n^-\sigma_n^- + \sigma_n^+\sigma_n^-\sigma_n^+\sigma_n^- + \sigma_n^+\sigma_n^+\sigma_n^-\sigma_n^- - \sigma_n^+\sigma_n^-\sigma_n^+\sigma_n^-$

$$\therefore [\sigma_i^+\sigma_j^-, \hat{N}_n] = 0$$

As the excitation number operator commutes with all the terms of the Hamiltonian, we conclude that it commutes with entire Hamiltonian $\hat{H}_S$ and thus the excitation number operator ($\hat{N}_{ex}$) is a good quantum number.

C.2 Total Momentum Operator $\hat{P}_{Total}$

The commutator between $\hat{H}_S$ and $\hat{P}_{Tot}$ is given by

$$\left[\hat{H}_S, \hat{P}_{Tot}\right] = \hat{H}_S\hat{P}_{Tot} - \hat{P}_{Tot}\hat{H}_S$$

where $\hat{P}_{Tot} = -i\hbar\sum_k^N \frac{\partial}{\partial\theta_k}$. First we consider, $\hat{H}_S\hat{P}_{Tot}$

$$\begin{aligned}\hat{H}_S\hat{P}_{Tot} = & -i\hbar^2\omega_\varepsilon \sum_k^N\sum_{i=1}^N \sigma_i^+\sigma_i^- \frac{\partial}{\partial\theta_k} + i\hbar^2\omega_\theta \sum_k^N\sum_{i=1}^N \frac{\partial^2}{\partial\theta_i^2}\frac{\partial}{\partial\theta_k} \\ & - i\hbar\sum_k^N \sum_{\substack{i,j \\ i>j}}^N 2u_{ij}\cos(2(\theta_i-\theta_j))\frac{\partial}{\partial\theta_k} \\ & - i\hbar\sum_k^N \sum_{\substack{i,j \\ i\neq j}}^N 2v_{ij}\cos(\theta_i-\theta_j)\sigma_i^+\sigma_j^-\frac{\partial}{\partial\theta_k}\end{aligned}$$

$$\begin{aligned}\hat{P}_{Tot}\hat{H}_S = & -i\hbar^2\omega_\varepsilon \sum_k^N\sum_{i=1}^N \frac{\partial}{\partial\theta_k}\sigma_i^+\sigma_i^- + i\hbar^2\omega_\theta \sum_k^N\sum_{i=1}^N \frac{\partial}{\partial\theta_k}\frac{\partial^2}{\partial\theta_i^2} \\ & - i\hbar\sum_k^N \sum_{\substack{i,j \\ i>j}}^N \frac{\partial}{\partial\theta_k}2u_{ij}\cos(2(\theta_i-\theta_j)) \\ & - i\hbar\sum_k^N \sum_{\substack{i,j \\ i\neq j}}^N \frac{\partial}{\partial\theta_k}2v_{ij}\cos(\theta_i-\theta_j)\sigma_i^+\sigma_j^-\end{aligned}$$

Where we have

$$\begin{aligned}\sum_{i,j}^{N}\sum_{k}^{N}\frac{\partial}{\partial\theta_k}\cos(2(\theta_i-\theta_j)) &= \sum_{i,j}^{N}\sum_{k}^{N}\left[-2\sin(2(\theta_i-\theta_j))\ \delta_{i,k}+2\sin(2(\theta_i-\theta_j))\ \delta_{j,k}\right]\\ &= \sum_{i,j}^{N}\left[-2\sum_{k}^{N}\sin(2(\theta_i-\theta_j))\ \delta_{i,k}+2\sum_{k}^{N}\sin(2(\theta_i-\theta_j))\ \delta_{j,k}\right]\\ &= 2\left[\sum_{i,j}^{N}\sin(2(\theta_i-\theta_j))-\sum_{i,j}^{N}\sin(2(\theta_i-\theta_j))\right]\end{aligned}$$

$$\therefore \sum_{i,j}^{N}\sum_{k}^{N}\frac{\partial}{\partial\theta_k}\cos(2(\theta_i-\theta_j))=0 \qquad \text{(C 7)}$$

Similarly,

$$\begin{aligned}\sum_{\substack{i,j\\ i>j}}^{N}\sum_{k}^{N}\frac{\partial}{\partial\theta_k}\cos((\theta_i-\theta_j)) &= \sum_{\substack{i,j\\ i>j}}^{N}\sum_{k}^{N}\left[-\sin((\theta_i-\theta_j))\ \delta_{i,k}+\sin((\theta_i-\theta_j))\ \delta_{j,k}\right]\\ &= \sum_{\substack{i,j\\ i>j}}^{N}\left[-\sum_{k}^{N}\sin((\theta_i-\theta_j))\ \delta_{i,k}+\sum_{k}^{N}\sin((\theta_i-\theta_j))\ \delta_{j,k}\right]\\ &= \sum_{\substack{i,j\\ i>j}}^{N}\left[\sin((\theta_i-\theta_j))-\sin((\theta_i-\theta_j))\right]\end{aligned}$$

$$\therefore \sum_{\substack{i,j\\ i>j}}^{N}\sum_{k}^{N}\frac{\partial}{\partial\theta_k}\cos((\theta_i-\theta_j))=0 \qquad \text{(C 8)}$$

Let us consider a general case

$$\left[\left(\sum_k^N \frac{\partial}{\partial\theta_k}\right), \cos(\theta_i-\theta_j)\right] g(\theta_m) = g(\theta_m)\left(\sum_k^N \frac{\partial}{\partial\theta_k}\right)\cos(\theta_i-\theta_j) + g^{'}(\theta_m)\cos(\theta_i-\theta_j) - \cos(\theta_i-\theta_j)g^{'}(\theta_m)$$

From equation (C 7) we know that $\left(\sum_k^N \frac{\partial}{\partial\theta_k}\right)\cos(\theta_i-\theta_j)=0$

$$\therefore \left[\left(\sum_k^N \frac{\partial}{\partial\theta_k}\right), \cos(\theta_i-\theta_j)\right] g(\theta_m) = 0 \tag{C 9}$$

We know that $\frac{\partial}{\partial\theta_k}$ commutes with $\sigma_i^+\sigma_i^-$ and $\sigma_i^+\sigma_j^-$. Also, we know from Clairaut's theorem on equality of mixed partial differentials that

$$\frac{\partial^2}{\partial\theta_i^2}\frac{\partial}{\partial\theta_k} = \frac{\partial}{\partial\theta_k}\frac{\partial^2}{\partial\theta_i^2} \tag{C 10}$$

Therefore, we find that $\hat{H}_S$ commutes with $\hat{P}_{Tot}$ *i.e.* $\left[\hat{H}_S, \hat{P}_{Tot}\right] = 0$. Thus, as the total momentum operator commutes with the Hamiltonian we could project the it into the zero momentum space as total momentum is conserved.

Bibliography

[1] Charles T Walker and Glen A Slack. Who named the-on's? *American journal of physics*, 38(12):1380–1389, 1970.

[2] Hermann Haken. Die theorie des exzitons im festen körper. *Fortschritte der Physik*, 6(6):271–334, 1958.

[3] Robert M Clegg. Förster resonance energy transfer—fret what is it, why do it, and how it's done. *Laboratory techniques in biochemistry and molecular biology*, 33:1–57, 2009.

[4] Gregory S Engel, Tessa R Calhoun, Elizabeth L Read, Tae-Kyu Ahn, Tomáš Mančal, Yuan-Chung Cheng, Robert E Blankenship, and Graham R Fleming. Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems. *Nature*, 446(7137):782, 2007.

[5] Jean-Luc Brédas and Robert Silbey. Excitons surf along conjugated polymer chains. *Science*, 323(5912):348–349, 2009.

[6] Volkhard May and Oliver Kühn. *Charge and energy transfer dynamics in molecular systems*. John Wiley & Sons, 2008.

[7] Lubert Stryer and Richard P Haugland. Energy transfer: a spectroscopic ruler. *Proceedings of the National Academy of Sciences of the United States of America*, 58(2):719, 1967.

[8] Aurélia Chenu and Gregory D Scholes. Coherence in energy transfer and photosynthesis. *Annual review of physical chemistry*, 66:69–96, 2015.

[9] William H Miller. Perspective: Quantum or classical coherence? *The Journal of chemical physics*, 136(21):06B201, 2012.

[10] Ivan Kassal, Joel Yuen-Zhou, and Saleh Rahimi-Keshari. Does coherence enhance transport in photosynthesis? *The journal of physical chemistry letters*, 4(3):362–367, 2013.

[11] Elisabetta Collini and Gregory D Scholes. Coherent intrachain energy migration in a conjugated polymer at room temperature. *Science*, 323(5912):369–373, 2009.

[12] J Clark, T Nelson, S Tretiak, G Cirmi, and Guglielmo Lanzani. Femtosecond torsional relaxation. *Nature Physics*, 8(3):225, 2012.

[13] Robert Binder, David Lauvergnat, and Irene Burghardt. Conformational dynamics guides coherent exciton migration in conjugated polymer materials: First-principles quantum dynamical study. *Physical review letters*, 120(22):227401, 2018.

[14] R Chang, JH Hsu, WS Fann, KK Liang, CH Chang, M Hayashi, J Yu, SH Lin, EC Chang, KR Chuang, et al. Experimental and theoretical investigations of absorption and emission spectra of the light-emitting polymer meh-ppv in solution. *Chemical Physics Letters*, 317(1-2):142–152, 2000.

[15] Wichard JD Beenken and Tonu Pullerits. Spectroscopic units in conjugated polymers: a quantum chemically founded concept? *The Journal of Physical Chemistry B*, 108(20):6164–6169, 2004.

[16] William Barford, Eric R Bittner, and Alec Ward. Exciton dynamics in disordered poly (p-phenylenevinylene). 2. exciton diffusion. *The Journal of Physical Chemistry A*, 116(42):10319–10327, 2012.

[17] William Barford. Excitons in conjugated polymers: A tale of two particles. *The Journal of Physical Chemistry A*, 117(13):2665–2671, 2013.

[18] Haibo Ma, Ting Qin, and Alessandro Troisi. Electronic excited states in amorphous meh-ppv polymers from large-scale first principles calculations. *Journal of chemical theory and computation*, 10(3):1272–1282, 2014.

[19] William Barford and Max Marcus. Theory of optical transitions in conjugated polymers. i. ideal systems. *The Journal of chemical physics*, 141(16):164101, 2014.

[20] Max Marcus, Oliver Robert Tozer, and William Barford. Theory of optical transitions in conjugated polymers. ii. real systems. *The Journal of chemical physics*, 141(16):164102, 2014.

[21] William Barford and Oliver Robert Tozer. Theory of exciton transfer and diffusion in conjugated polymers. *The Journal of chemical physics*, 141(16):164103, 2014.

[22] William Barford. *Electronic and optical properties of conjugated polymers*, volume 159. Oxford University Press, 2013.

[23] Wichard JD Beenken and Hans Lischka. Spectral broadening and diffusion by torsional motion in biphenyl. *The Journal of chemical physics*, 123(14):144311, 2005.

[24] Wichard JD Beenken. Torsional broadening in absorption and emission spectra of bithiophene as calculated by time-dependent density functional theory. *Chemical Physics*, 349(1-3):250–255, 2008.

[25] Philip W Anderson. Absence of diffusion in certain random lattices. *Physical review*, 109(5):1492, 1958.

[26] Eleftherios N Economou. *Green's functions in quantum physics*, volume 3. Springer.

[27] G. Vidal. Entanglement renormalization. *Phys. Rev. Lett.*, 99:220405, Nov 2007.

[28] Steven R White. Density-matrix algorithms for quantum renormalization groups. *Physical Review B*, 48(14):10345, 1993.

[29] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Physical review letters*, 93(4):040502, 2004.