COMPUTATIONAL STUDY OF EXCITONS AND BIEXCITONS IN SEMICONDUCTOR CORE/SHELL NANOCRYSTALS OF TYPE I AND TYPE II

A THESIS

SUBMITTED TO THE DEPARTMENT OF PHYSICS

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Ozan Yerli

July, 2013

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Hilmi Volkan Demir (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Oğuz Gülseren

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Vakur B. Ertürk

Approved for the Graduate School of Engineering and Science:

Prof. Levent Onural
Director of the Graduate School

ABSTRACT

COMPUTATIONAL STUDY OF EXCITONS AND BIEXCITONS IN SEMICONDUCTOR CORE/SHELL NANOCRYSTALS OF TYPE I AND TYPE II

Ozan Yerli
M.S. in Physics
Supervisor: Assoc. Prof. Hilmi Volkan Demir
July, 2013

In this thesis, we studied electronic structure and optical properties of Type-I, Type-II, and quasi Type-II semiconductor nanocrystals (also known as colloidal quantum dots). For a parametric study, we developed quantum mechanical models and solved them using both analytical and numerical techniques. The simulation results were compared to the experimental findings. We showed that charge carrier localization at different spatial locations could be tuned by controlling size parameters of the core and shell.

While tuning charge localization, we also predicted photoluminescence peaks of these core/shell nanocrystals using our theoretical and numerical calculations. We demonstrated that Type-II nanocrystals exhibit different tuning trends compared to the Type-I ones.

We also investigated biexcitonic properties of nanocrystals using quantum mechanical simulations, which are important especially in lasing applications. We showed that two-photon absorption mechanism can be tuned by changing the core and shell size in quantum dots. We calculated at which core and shell sizes biexcitons in quantum dots show attractive or repulsive interaction. The computational studies presented in this thesis played an important role in the experimental demonstrations and understanding of controlling excitonic features of core/shell nanocrystals.

Keywords: Nanocrystals, quantum dots, two photon absorption, exciton, biexciton, amplified spontaneous emission.

ÖZET

YARI İLETKEN TİP I VE TİP II ÇEKİRDEK/KABUK NANOKRİSTALLERDEKİ EKSİTON VE BİEKSİTONLARIN HESAPLAMALI İNCELEMESİ

Ozan Yerli
Fizik, Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Hilmi Volkan Demir
Temmuz 2013

Bu tezde, Tip-I, Tip-II ve Tip-II benzeri kuantum nokta nanokristallerin elektronik ve optik özelliklerini inceledik. Parametrik bir çalışma için kuantum mekaniksel modeller geliştirildi, analitik ve numerik teknikler kullanılarak çözüldü. Simulasyon sonuçları deneysel sonuçlarla karşılaştırıldı. Yük taşıyıcıların farklı bölgelerdeki lokalizasyonunun çekirdek ve kabuk yapılarının boyutlarını değiştirerek ayarlanabileceği gösterildi.

Yük lokalizasyonunun ayarlanmasının yanında, çekirdek/kabuk nanokristallerinin ışıma tepe noktaları teorik ve numerik hesaplarla öngörüldü. Ayrıca Tip-II nanokristallerin Tip-I nanokristallere göre farklı ayar eğilimlerine sahip oldukları gösterildi.

Kuantum mekaniksel simulasyonları kullanarak nanokristallerin özellikle lazer uygulamalarında oldukça önemli olan bieksitonik özellikleri incelendi. İki foton soğurma mekanizmasının kuantum noktaların çekirdek ve kabuk boyutlarıyla ayarlanabileceğini gösterdik. Hangi çekirdek ve kabuk boyutlarında bieksitonların çekici veya itici etkileşim gösterdiklerini hesapladık. Bu tezde sunulan hesaplamalı araştırmalar çekirdek/kabuk nanokristallerin eksitonik özelliklerinin kontrolünün anlaşılmasında ve deneysel olarak gösterilmesinde önemli bir rol oynamıştır.

Anahtar sözcükler: Nanokristaller, kuantum noktaları, iki foton soğurması, eksiton, bieksiton, artırılmış kendiliğinden ışıma.

Acknowledgement

I would like to thank Assoc. Prof. Hilmi Volkan Demir for his invaluable support, suggestions, and guidance during my studies.

I would like to acknowledge Prof. Oğuz Gülseren and Assoc. Prof. Vakur B. Ertürk for reading the thesis and being in the thesis committee.

I would like to thank all my instructors and members of the Physics Department for helping me get a strong understanding of physics.

I would like to acknowledge Ahmet Fatih Cihan, Yusuf Kelestemur, and Burak Guzelturk for their valuable discussions about topics in this thesis. I also thank them for helping me to get the experimental results which I used to compare with my computational results in this thesis.

I would like to thank Kıvanç Güngör, Shahab Akhavan, Talha Erdem, Aydan Yeltik, Can Uran, and all other HVD Group members for their support and help during my master's studies.

I would like to thank my colleagues, İsmail Can Oğuz and Kutan Gürel for making this 2 year of studies delightful and productive.

I would like to thank Elif Yıldız, Enis Dönmez, Yurdakul Göksu Orhun for their moral support during this intensive period of mine.

I would like to thank Almıla for being in my life. She gave extensive support and motivation which helped me complete this thesis.

I would like acknowledge my family, my mother Hilal, my father Nusret, and my grandmother Yüksel. Without their help, support and love I would not be successful.

I would like to acknowledge TUBITAK for scholarship.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Semiconductor nanocrystals, or colloidal quantum dots, find many important applications including photovoltaics, lasers, and light-emitting diodes. They are subject of particular interest due to size dependence of their optical properties arising from quantum confinement effects. This size dependence enables us to tune the optoelectronic properties of nanocrystals [1].

The materials used in nanocrystal heterostructures give rise to different spatial localization configurations for the electrons and holes in the semiconductor. If the bandgap of one material lies within the bandgap of the other material, the electrons and holes will be found at the same spatial location. This type of band alignment is called Type-I. If both of the bandgaps of the materials have non-overlapping energies, the electrons and holes will localize at the different spatial positions. This type of band alignment is called Type-II, which can be seen in Figure 1.2.

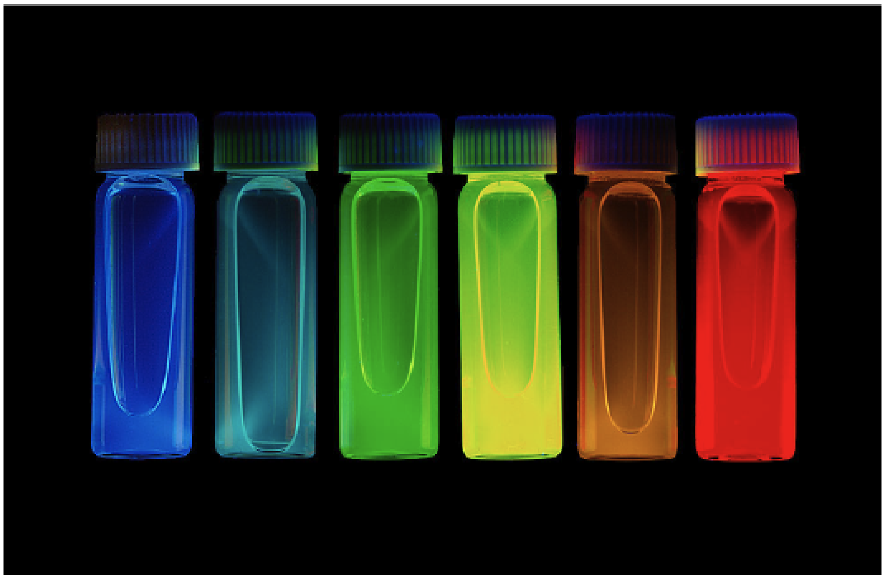


Figure 1.1: Nanocrystals of different sizes illuminated with ultraviolet light after [2].

In the Type-II band alignment we can control confinement locations of the charge carriers, which change their wavefunction overlap. We can localize one charge carrier into the core while the other carrier is delocalized over both core and shell. This opens up new possibilities for the usage of nanocrystals including low threshold lasers, fast optical switches, quantum information processors, infrared detectors, and fast access memories [3]. Since these devices require an accurate control on the optoelectronic properties, Type-II nanocrystals may offer flexibility. Type-II nanocrystals also have increased emission lifetimes [3].

It is shown that characteristics of Type-I nanocrystals can be obtained from theoretical and numerical calculations with a high degree of accuracy [1]. In order to understand optoelectronic properties of nanocrystals, understanding excitonic behaviour is crucial. Various simulation methods and techniques have been developed to investigate excitons in nanocrystals [4] [5] [6] [7] [8] [9].

In this study, we will analytically understand properties of the Type-II heterostructures and use these to deduce photoluminescence peak of the nanocrystals. We will also examine the parameter dependence of the wavefunction localizations and overlap of the electron/hole wavefunctions using numerical techniques.

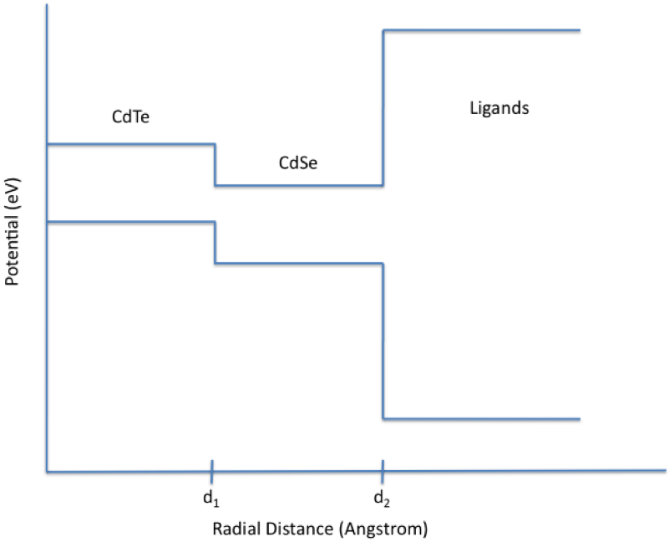


Figure 1.2: Band structure of the CdTe/CdSe core-shell Type-II nanocrystal.

This will allow us to optimize wavefunctions for the desired optoelectronic properties.

Nanocrystals are also suitable for optical gain and lasing applications. Two-photon absorption mechanism has also been shown as a promising method for excitation of optical gain media. In this thesis, using quantum mechanical simulations we show that the single material system of CdSe/CdS can be tuned to give blue or red-shifting behaviour of amplified spontaneous emission (ASE) in two-photon absorption mechanism. This is due to the exciton-exciton interactions becoming either attractive or repulsive by tuning the core and shell size. Here using quantum mechanical simulation techniques biexcitonic properties of nanocrystals are investigated.

The organization of the thesis is as follows. In the second chapter we introduce the basics of nanocrystal optoelectronics. We give brief background information on how to model nanocrystals theoretically and their optical properties. In the third chapter we introduce simulation techniques used for solving quantum mechanical problems. Effective mass approximation and self-consistent solution of multi-particle Schrodinger equations, which are the techniques we used for exciton and biexciton calculations in nanocrystals are explained in this chapter. In the fourth chapter multigrid technique is explained. We used multigrid technique to solve Poisson equation arising from Kohn-Sham equations. In the fifth chapter we present our research work on excitonic and biexcitonic properties of nanocrystals. Experimental and simulation results are given in this chapter. In the last chapter, we conclude by providing a summary of our findings and a comparative description of exciton and biexciton properties in different kinds of nanocrystals.

Chapter 2

NANOCRYSTAL OPTOELECTRONICS

2.1 Introduction

Nanocrystal quantum dots are nanometer-sized, nearly spherical crystal structures. Since the size of a nanocrystal is comparable with the Bohr radius of excitons in the nanocrystal, the nanocrystal size can affect many properties of the excitons. Using various synthesis techniques the size of nanocrystals can be controlled. This allows us to tune excitonic properties of the nanocrystals very precisely. Due to their tunability, nanocrystal quantum dots are suitable for light-emitting diodes [10], [11], [12] and for lasing applications [13].

Beside tunability of their optoelectronic properties by controlling their size, it is also possible to engineer their shape (spherical, nanorod [14], and tetrapod [15]). Different shapes also allow for new possibilities for engineering distinct optoelectronic properties.

2.2 Theoretical Model

A spherically symmetric nanocrystal model is used as the theoretical model for the colloidal quantum dots here. As we investigate s-states (l=0 and m=0), the energy level only depends on the radial part of wavefunction, $R(r)$. For s-states, radial part of the Schrodinger equation is:

$$\left\{-\frac{1}{2m^* r^2}\frac{d}{dr}\left(r^2\frac{d}{dr}\right)+V(r)\right\}R(r)=ER(r) \tag{2.1}$$

By making $u(r) = rR(r)$ transformation, we can convert the problem into an one-dimensional square potential problem. After the transformation the radial Schrodinger equation becomes:

$$[-\frac{1}{2m^*}\frac{\partial^2}{\partial r^2}+V(r)]u(r)=Eu(r) \tag{2.2}$$

Here we assume the core has radius of d_1, and the shell has thickness of $(d_2 - d_1)$. If we impose boundary conditions at the interface (r $= d_1$), from the continuity of the wavefunction, we obtain the condition:

$$u_I(d_1) = u_{II}(d_1) \tag{2.3}$$

From the continuity of the probability currents, we get the condition [16]:

$$\left.\frac{1}{m_1}\frac{\partial u_I}{\partial r}\right|_{d_1} = \left.\frac{1}{m_2}\frac{\partial u_{II}}{\partial r}\right|_{d_1} \tag{2.4}$$

In Equations 2.3 and 2.4, u_I and u_{II} are the wavefunctions in the regions $0 < r < d_1$ and $d_1 < r < d_2$, respectively. In Equation 2.4, the m_1 and m_2 are the effective masses of the charge carriers in different semiconductors. These material parameters are taken from Table I of the reference [17].

Note that, in Equation 2.4 the first order derivatives of the wavefunction are not continuous because of the effective mass mismatch across the two materials.

The discontinuity of the first order derivatives in different materials are proportional to the ratio of the effective masses in these different materials. This is known as the BenDaniel-Duke boundary conditions [16].

Applying these boundary conditions, we obtain the following matrix, which should have zero determinant in order to have non-trivial solutions:

$$A = \begin{bmatrix} -e^{-d_1 k_1} + e^{d_1 k_1} & -sin(d_1 k_1) + cos(d_1 k_2) tan(d_2 k_2) \\ \frac{k_1}{m_1}(e^{-d_1 k_1} + e^{d_1 k_1}) & -\frac{k_2}{m_2}[cos(d_1 k_2) + sin(d_1 k_2) tan(d_2 k_2)] \end{bmatrix} \tag{2.5}$$

We calculated the determinant of the matrix analytically and using Mathematica found the solutions for the equation:

$$detA = 0 \tag{2.6}$$

We calculated maximum photoluminescence (PL) should occur at 802.4 nm wavelength for a CdTe/CdSe nanocrystal heterostructure with $d_1 = 32$Å and $d_2 = 43$Å. These values are lower than the photoluminescence peak of the experimental result in Figure 2 of the reference [18]. Since we assumed an infinite barrier at the ligand interface, we increase the quantum confinement effect, which leads to higher energy elevels. That is the reason why we obtain blue-shifted results for the PL peak compared to the experimental results.

In order to obtain more accurate results for the PL peak calculations, we need to assign a finite potential value to the ligand interface of the heterostructure. In order to solve this problem analytically we need to increase the dimension of the matrix in Equation 2.5. Also, we intend to investigate multi-particle quantum dot systems, which is not possible with the pure analytical model. To address these problems we used numerical techniques that are more efficient than the analytical approach.

2.3 Optical Properties

Since nanocrystals have size comparable to the excitonic Bohr radius, their optical properies are diffent from the bulk materials. Their optical properties can be controlled by changing their size. This gives tunability to the emission and absorption spectra of nanocrystals.

There is a shift between the absorption and emission peaks of nanocrystals. Since this shift suppresses reabsorption, it increases luminiscence. In order to further enhance luminiscence, core/shell quantum structures are used. The shell structure passivates the surface and decreases the nonradiative recombination.

Quantum dots are also a good candidate for a gain medium due to their relatively low optical gain thresholds, temperature stability, and narrow emission bandwidth [19]. Using various photonic structures, quantum dot lasers using optical pumping have been developed [20] [21] [22] [23]. However, in these studies single-photon absorption mechanism has been used for optical pumping. As an alternative to single-photon absorption, two-photon absorption (TPA) mechanism can be used. TPA has an advantage of reducing the risk of photo-damaging [24] [25] [26] [27].

As a result of TPA mechanism, two excitons are created. Since two excitons are created, the resulting amplified spontaneous emission due to decay of these excitons are related to the biexciton binding energy. Hence, it is important to investigate biexciton energy using simulations in order to create structures optimized for TPA.

2.4 Classification of Nanocrystals

2.4.1 Type-I

In Type-I nanocrystals, the minimum energy region of electrons and holes coincide. This causes electrons and holes localize very closely at the same location. A sample potential diagram for a Type-I nanocrystal can be seen in figure 2.1

2.4.2 Type-II

In Type-II nanocrystals, the minimum energy region of electrons is different from the minimum energy region of holes. For this reason, electrons and holes localize in different spatial regions. A sample potential diagram for a Type-II nanocrystal can be seen in Figure 2.2.

Type-II nanocrystals have increased optical gain lifetime compared to Type-I nanocrystals because in Type-II nanocrystals Auger recombination is suppressed due to electrons and holes being seperated in different regions [29], [30].

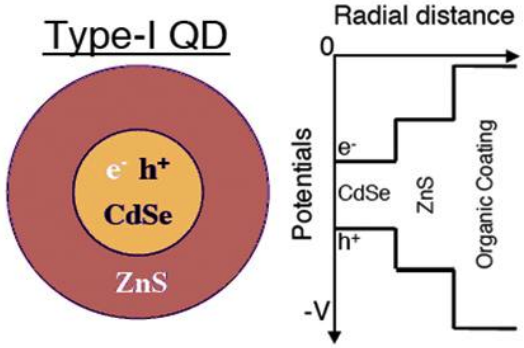


Figure 2.1: Schematic and potential diagram of a Type-I nanocrystal (CdSe / ZnS) after [28].

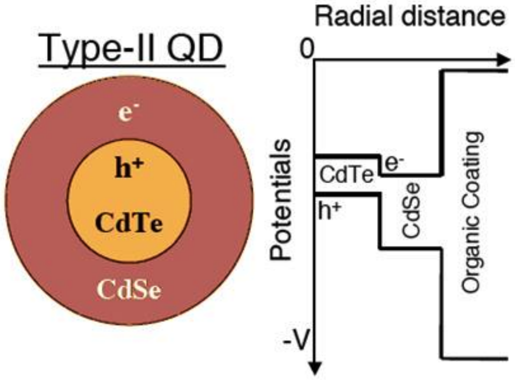


Figure 2.2: Schematic and potential diagram of a Type-II nanocrystal (CdTe / CdSe) after [28].

Chapter 3

QUANTUM MECHANICAL SIMULATIONS

3.1 Introduction

Embedded semiconductor quantum dots have been investigated by Laheld and co-workers theoretically [31]. They used effective-mass approximation and finite-band offset between two materials. An average dielectric constant is assumed between both materials. They showed Type-I, Type-II and quasi Type-II properties theoretically.

Colloidal core/shell quantum dots have been investigated using similar techniques [32] [33] [34] [35]. The main difference comes from the fact that the outermost semiconductor is not infinite, it has a finite thickness. This causes electrons and holes to localize in the shell structure. Klimov and co-workers have used effective mass approximation and perturbation theory, which includes interface polarization effects to investigate colloidal quantum dots [35]. Colloidal quantum dots have also been modeled using k.p method, tight-binding, and pseudopotential methods [36] [32].

3.2 Atomic Unit System

In order to make simulations simpler we used the atomic unit system in our calculations. In the atomic unit system $\hbar = m_e = e = 1$, where m_e is the mass of the electron and $-e$ is the charge of the electron [37].

Lengths are expressed in the units of classical Bohr radius which is $a_0 = 9.5292$Å.

One atomic unit of energy is 27.2116 eV, which is used to convert calculated energies to eV.

3.3 Effective Mass Approximation

Group velocity of a free electron wave packet is given by:

$$v = \frac{dw}{dk} \tag{3.1}$$

We also have the relation between the frequency and the energy:

$$w = \frac{E}{\hbar} \tag{3.2}$$

Hence, the group velocity can be expressed as:

$$v = \frac{1}{\hbar}\frac{dE}{dk} \tag{3.3}$$

Since the momentum of electron is $\hbar k$, if we take its derivative, we can write it in the form of Newton's second law:

$$\frac{dp}{dt} = \hbar\frac{dk}{dt} = m^*\frac{dv}{dt} \tag{3.4}$$

In Equation 3.4, m^* is called the effective mass. If we use the expression for the group velocity given in Equation 3.3, we obtain the relation:

$$\hbar\frac{dk}{dt} = m^*\frac{1}{\hbar}\frac{d^2E}{dk^2}\frac{dk}{dt} \tag{3.5}$$

$\frac{dk}{dt}$ cancels from both sides and we finally arrive at:

$$\frac{1}{m^*} = \frac{1}{\hbar^2}\frac{d^2E}{dk^2} \tag{3.6}$$

Equation 3.6 gives the definition of the effective mass [38]. It shows that in crystalline solids we can calculate effective mass from the dispersion relation. Once we obtain the effective mass we can reduce the problem into a free electron problem. The electron moves in the solid as if it was a free electron with the mass m^*.

3.4 Single Particle Simulations

In our computational study we first made the transformation $u(r) = rR(r)$ in order to convert the problem into an one-dimensional Schrodinger equation problem. After that transformation and using the atomic units, we need to solve the equation:

$$[-\frac{1}{2m^*}\frac{\partial^2}{\partial r^2} + V(r)]u(r) = Eu(r) \tag{3.7}$$

In Equation 3.7, m^* is the ratio of the effective mass of the particle to the free electron mass.

In order to solve this one-dimensional system numerically, we need to discretise radial space, Hamiltonian operator, and wavefunction [39]. We used $h = 0.01$ as our discretization parameter and meshed the space into equally separated discrete parts, which are hÅ apart.

We can write the second order derivative operator as a finite difference:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{3.8}$$

Assume we try to solve the following eigenvalue equation:

$$\frac{\partial^2}{\partial x^2}f(x) = Ef(x) \tag{3.9}$$

For each point in our discretized space, we can write the Equation 3.8. Hence we obtain the following set of equations (assuming the boundary conditions, $f(x_0 - h) = 0$ and $f(x_n + h) = 0$):

$$\begin{cases} \frac{1}{h^2}[-2f(x_0) + f(x_0 + h)] = Ef(x_0) \\ \frac{1}{h^2}[f(x_1 - h) - 2f(x_1) + f(x_1 + h)] = Ef(x_1) \\ \frac{1}{h^2}[f(x_2 - h) - 2f(x_2) + f(x_2 + h)] = Ef(x_2) \\ \ldots \\ \frac{1}{h^2}[f(x_2 - h) - 2f(x_2)] = Ef(x_n) \end{cases} \tag{3.10}$$

We can write these set of equations as a matrix equation:

$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & \cdots & 1 & -2 \end{bmatrix} \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \cdots \\ f(x_n) \end{bmatrix} = E \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \cdots \\ f(x_n) \end{bmatrix} \tag{3.11}$$

If we solve the matrix equation in Equation 3.11, we obtain values of the eigenfunction $f(x)$ at our discretized space points $(x_0, x_1, x_2, ..., x_n)$.

For n space points, the matrix in Equation 3.11 becomes an n-by-n tridiagonal matrix.

Hence, the Hamiltonian operator can be written as a matrix. For example, for three space points, the Hamiltonian operator can be expressed as:

$$\mathbf{H} = -\frac{1}{2h^2}\begin{bmatrix} -2/m_1^* & 1/m_2^* & 0 \\ 1/m_1^* & -2/m_2^* & 1/m_3^* \\ 0 & 1/m_2^* & -2/m_3^* \end{bmatrix} + \begin{bmatrix} V_1 & 0 & 0 \\ 0 & V_2 & 0 \\ 0 & 0 & V_3 \end{bmatrix} \tag{3.12}$$

In Equation 3.12, m_1^*, m_2^*, and m_3^* are the effective masses of the particle and V_1, V_2, and V_3 are the potential energies in the space points 1, 2, and 3, respectively.

By writing the Hamiltonian of the system using finite difference method and finding its eigenvectors using computational techniques, we can obtain energies and wavefunctions of single particles.

3.5 Multi-Particle Simulations

For a multi-particle system, we can write the following set of Schrodinger equations [40]:

$$\left(-\frac{\hbar^2}{2m^*}\nabla^2 + V_{eff}(\vec{r})\right)\phi_i(\vec{r}) = \epsilon_i\phi_i(\vec{r}) \tag{3.13}$$

where $\hbar$ is the reduced Planck constant, m^* is the effective electron (or hole) mass, V_{eff} is the effective potential, ϕ_i is the wavefunction of the i^{th} electron (or hole), ϵ_i is the energy of the i^{th} electron (or hole).

Here we use effective mass approximation to calculate the energy eigenvalue and wavefunction of the first electron. Then we calculate the charge density using the equation:

$$\rho(\vec{r}) = \sum_i^N \mid \phi_i(\vec{r}) \mid^2 \tag{3.14}$$

where N is the total number of charge carriers.

We use the charge density in our multigrid Poisson equation solver to calculate the new effective potential:

$$\nabla^2 V_{eff} = -\frac{\rho}{\epsilon} \tag{3.15}$$

Using this new effective potential we repeat the calculations including the first hole, the second electron, and the second hole until we obtain a self-consistent set of charge densities and wavefunctions.

3.6 Material Parameters

The material parameters for CdSe/CdS quantum dots used in this thesis are taken from the supplementary information of the reference [19]:

Parameter	CdSe	CdS
Electron Effective Mass	0.13 m_0	0.21 m_0
Hole Effective Mass	0.45 m_0	0.68 m_0
Bandgap Energy	1.75 eV	2.50 eV

Table 3.1: Material parameters for CdSe/CdS quantum dots.

For the CdSe/ZnS Type-I core/shell structure, for electrons, we assigned zero potential energy inside the CdSe core and used a barrier of 1.05 eV for the ZnS shell. The ligand barrier for the electrons is 4 eV. For holes, we assigned zero potential energy inside the CdSe core and used a barrier of 0.95 eV for the ZnS shell. The ligand barrier for the holes is 10 eV.

For the CdTe/CdSe Type-II core/shell structure, for electrons, we again assigned zero potential energy inside the CdSe shell and used a barrier of 0.67 eV for the CdTe core. The ligand barrier for the electrons is 4 eV. For holes, we used zero potential energy inside the CdTe core and took a barrier of 0.97 eV for the CdSe shell. The ligand barrier for the holes is 10 eV.

Chapter 4

MULTIGRID TECHNIQUE

4.1 Introduction

In 1D, Poisson's equation can be easily solved by writing Laplacian operator as a matrix, and since we already have our potential and charge density vectors as 1D vectors, we can write the system in the form of $\mathbf{Ax} = \mathbf{b}$, and solve it either directly by taking inverse or using iterative methods (Jacobi, Gauss-Seidel, Successive Over-Relaxation, Multigrid) [41].

In higher dimensions, since we still have a linear operator (Laplacian), we can still write the problem as a linear algebra problem but this time the matrix A and b are not self-evident. We need to develop a technique to map $n \times n \times n$ 3D arrays of potential and charge density into column vectors.

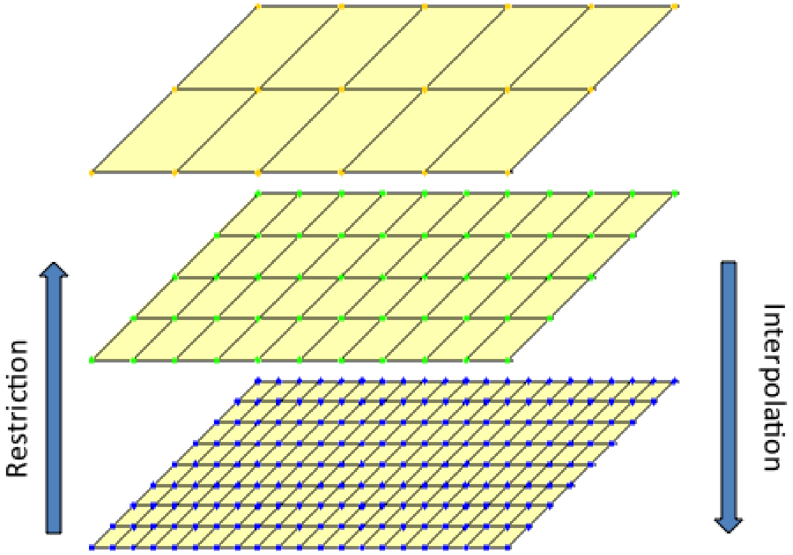


Figure 4.1: Multigrid schema after [41].

Matrix representation of the Laplacian operator in 3D has very large dimensions. Using standard techniques it takes a large amount of time to solve this matrix system. Using the multigrid technique we can solve very large matrix systems efficiently. In the multigrid technique using the restriction operator we map the system into a coarser grid which can be seen in Figure 4.1. Then in this new grid, we recursively call our multigrid procedure. In the coarsest grid, we solve the problem directly. After we solve the problem in the coarsest grid, we interpolate the solution back into to the original grid. This operation takes less time compared to the direct solution of the original problem [41].

In this thesis, we used Kronecker tensor product to create 2D and 3D Laplacian matrices and interpolation/restriction matrices for the multigrid technique. We created these matrices from their analogous 1D matrices. We obtain the potential and solution vectors using the technique “standard ordering”, which will be explained later.

4.2 Kronecker Tensor Product

The Kronecker tensor product of two matrices creates a higher dimensional block matrix in the following way [42]:

Let $\mathbf{A}$ be an $m \times n$ matrix and $\mathbf{B}$ be a $p \times q$ matrix. Their Kronecker tensor product is an $mp \times nq$ matrix given by:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \tag{4.1}$$

We write the matrix $\mathbf{B}$, by multiplying with elements of $\mathbf{A}$ and obtain a large block matrix.

4.3 Poisson's Equation in 1D

Poisson's equation in 1D can be expressed in the matrix form by using the second order finite difference formula. The result is (for 3 points in 1D):

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{4.2}$$

In this point of view, we can easily see the averaging property of the Laplacian operator. Here, b_1, b_2, b_3 are the charge densities divided by the permittivity in the corresponding points. If we want to add the boundary conditions, we add the potential at the two boundaries to b_1 and b_3. Hence, this matrix system fully represents the original problem.

4.4 Standard Ordering

Let M be an $m \times m \times m$ 3D array in MATLAB, then the command v=M(:) gives a vector that has formed by applying standard ordering to the array M. In the graphical representation, the vector v is created in this way:

$$\begin{bmatrix} M(1,1,1) \\ M(2,1,1) \\ M(3,1,1) \\ ... \\ ... \\ M(1,2,1) \\ M(2,2,1) \\ ... \\ ... \\ M(1,3,1) \\ M(2,3,1) \\ ... \\ ... \\ ... \\ M(1,1,3) \\ M(2,1,3) \\ ... \\ ... \\ ... \\ M(3,3,3) \end{bmatrix} \tag{4.3}$$

So this vector has exactly m^3 elements, created by first iterating the first index, then the second, and then the third.

4.5 Poisson's Equation in 2D and 3D

If we use standard ordering in our vectors, x and b, the corresponding Laplacian matrices can be easily obtained by the Kronecker tensor product [42].

$$A_{2D} = A_{1D} \otimes I_{1D} + I_{1D} \otimes A_{1D} \tag{4.4}$$

$$A_{3D} = A_{2D} \otimes I_{2D} + I_{2D} \otimes A_{2D} \tag{4.5}$$

As expected in 2D we obtain a matrix that averages 4 points neighboring the middle point. In 3D matrix we have 6 points between the middle point averaged.

So, the problem in 3D is reduced into a sparse linear system solution, which is very convenient, since we can use all techniques developed for linear algebra, and even we can solve the matrix system exactly (by spending a lot of time).

Here we used the solution obtained using MATLAB's sparse matrix solver subroutines to find the optimal values for multigrid pre-smoothing/post-smoothing parameters.

4.6 Multigrid Algorithm

In order to implement the multigrid method, we need two new matrices: restriction and interpolation. The restriction matrix takes a vector from a finer grid and maps it into a vector in a coarser grid. The interpolation grid does the vice-versa. Since the dimension of linear space changes, the restriction and interpolation matrices are rectangular.

Writing the restriction and interpolation matrices in 1D are easy. For example, for n=7 we can write the restriction matrix as:

$$R_{1D} = \begin{bmatrix} 1/4 & 1/2 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 1/2 & 1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/2 & 1/4 \end{bmatrix} \tag{4.6}$$

As we can see, R_{1D} takes weighted averages of the neighboring points and writes them into the coarser grid. The interpolation operator in 1D is just the transpose of R_{1D} multiplied by 2 (to ensure normalization).

We can obtain the restriction and interpolation matrices for 2D and 3D using the Kronecker Tensor Product:

$$R_{2D} = R_{1D} \otimes R_{1D} \tag{4.7}$$

$$R_{3D} = R_{2D} \otimes R_{1D} \tag{4.8}$$

Once we have the restriction and interpolation matrices at hand, we can use the multigrid algorithm as follows [41]:

1. Make an initial guess for the solution: x_h
2. Calculate the residual: $r_h = b_h - Ax_h$
3. Restrict the residual to a coarser grid: $r_{2h} = Rr_h$
4. "Solve" the system $A_{2h}E_{2h} = r_{2h}$ and get a correction term (E_{2h}). Here, A_{2h} is obtained from the formula: $A_{2h} = RAI$ (It is equivalent to interpolating a vector, applying A to it, and then restricting it back.) In this step, for "solving" this new system, we again recursively call our multigrid procedure to solve it again using the multigrid. In the coarsest level, we solve the system directly. If we iterate this step more than once, we get W-cycle and Full Multigrid.
5. Interpolate back the correction term $E_h = IE_{2h}$ and add the interpolated term to the old approximate solution.

6. Using the new approximate solution return to step 1. If the residual becomes "negligible" (in our case $\sim 10^{-5}$), terminate iteration.

In Figures 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 we present some sample results of our multigrid Poisson solver.

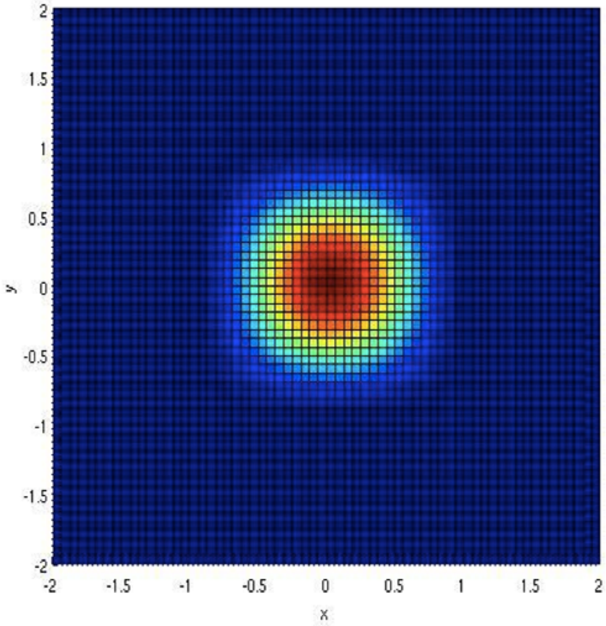


Figure 4.2: Charge distribution of a single electron in the state (1,1,1) of a quantum dot.

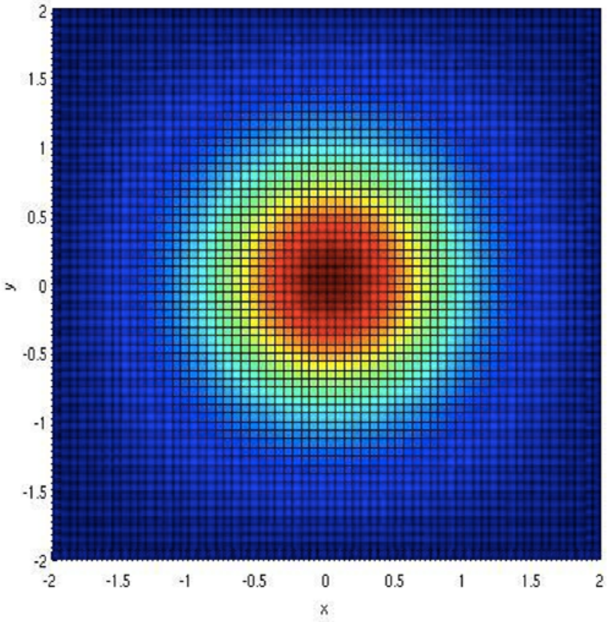


Figure 4.3: Potential of a single electron in the state (1,1,1) of a quantum dot.

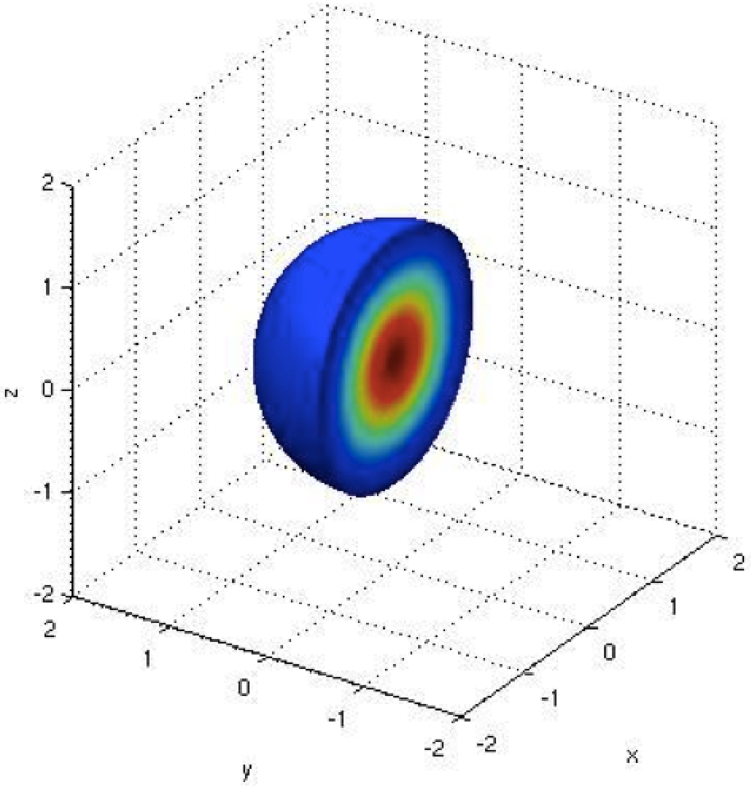


Figure 4.4: Isopotential surface slice of a single electron in the state (1,1,1) of a quantum dot.

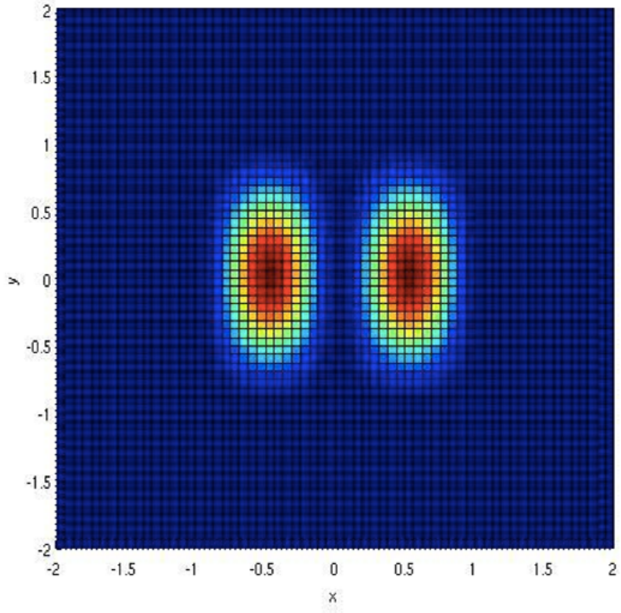


Figure 4.5: Charge distribution of a single electron in the state (2,1,1) of a quantum dot.

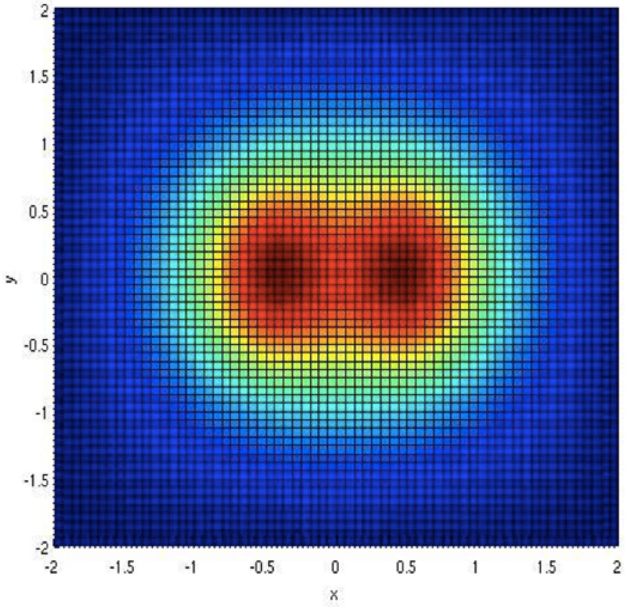


Figure 4.6: Potential of a single electron in the state (2,1,1) of a quantum dot.

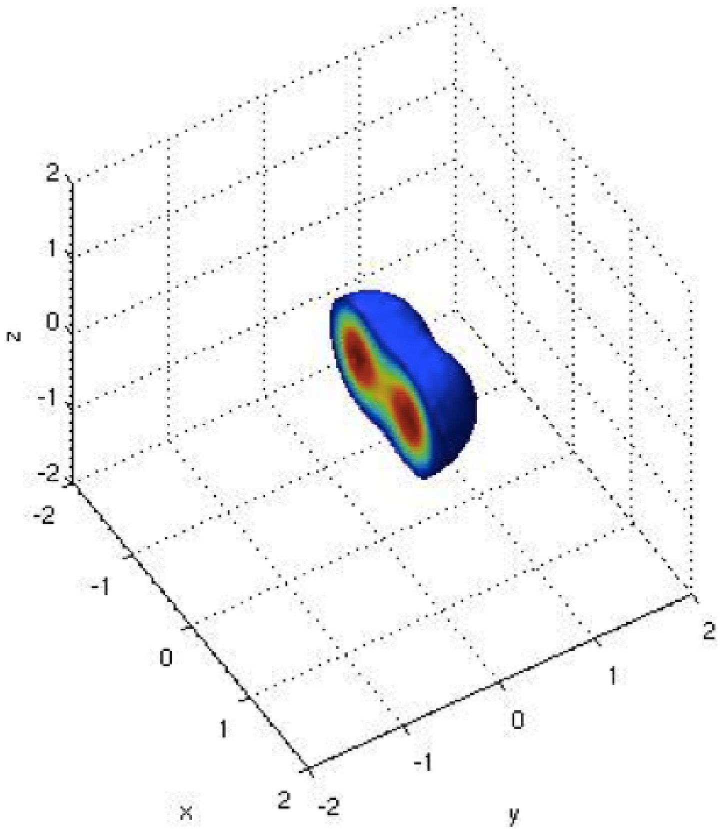


Figure 4.7: Isopotential surface slice of a single electron in the state (2,1,1) of a quantum dot.

Chapter 5

RESULTS AND DISCUSSION

5.1 Electron and Hole Wavefunctions and Energies for Type I/Type II Spherical QDs

As observed in Figure 5.2 exciton energy decreases by increasing core radius since this decreases the quantum confinement effect. The increasing shell thickness slightly decrease energy for small cores but does not alter the energy very much. This can be understood when we look at the wavefunction distribution in Figure 5.1. Since both wavefunctions are mostly confined to the core, varying the shell thickness does not affect the exciton energy much.

We calculated that the wavefunctions of electron and hole can be localized at different spatial locations in Type-II heterostructures as seen in Figure 5.3. Also, by changing the shell thickness, we can confine the electron and hole to the same spatial location as observed in Figure 5.4. This shows that Type-II heterostructures give us more control on the charge carrier localizations.

When we consider the PL peak energies of different Type-II structures, we observe that both the core and shell size affect the exciton energy since one of the charge carrier is confined in the core while the other in the shell. This leads to separated energy curves presented in Figure 5.5. These curves coincide when

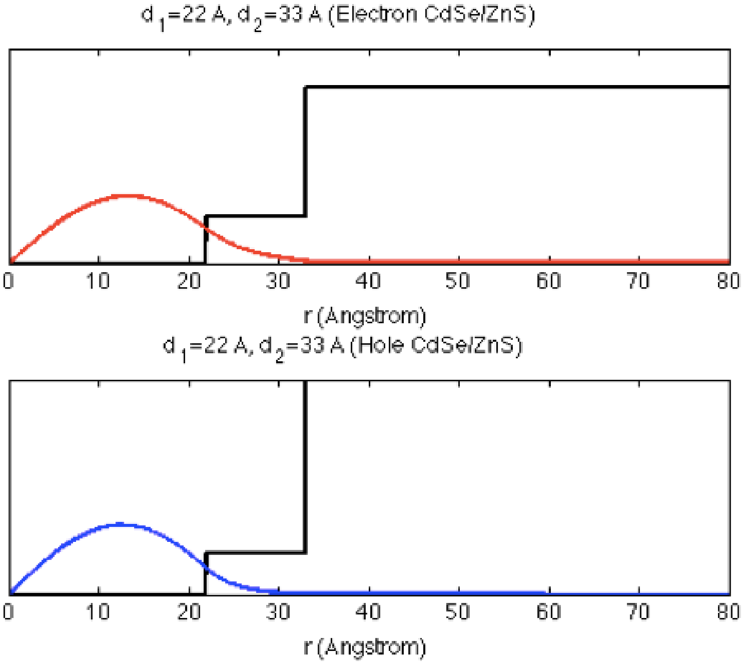


Figure 5.1: Wavefunctions of electron and hole for CdSe/ZnS (Type-I) 1s-state.

the structure is Type I as observed in Figure 5.2.

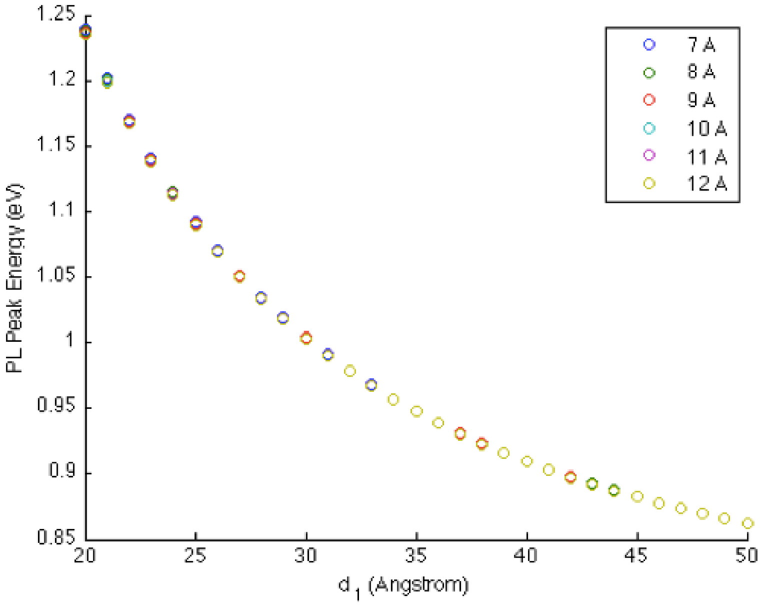


Figure 5.2: Exciton energy of CdSe/ZnS (Type-I) with respect to d_1 for different shell thicknesses.

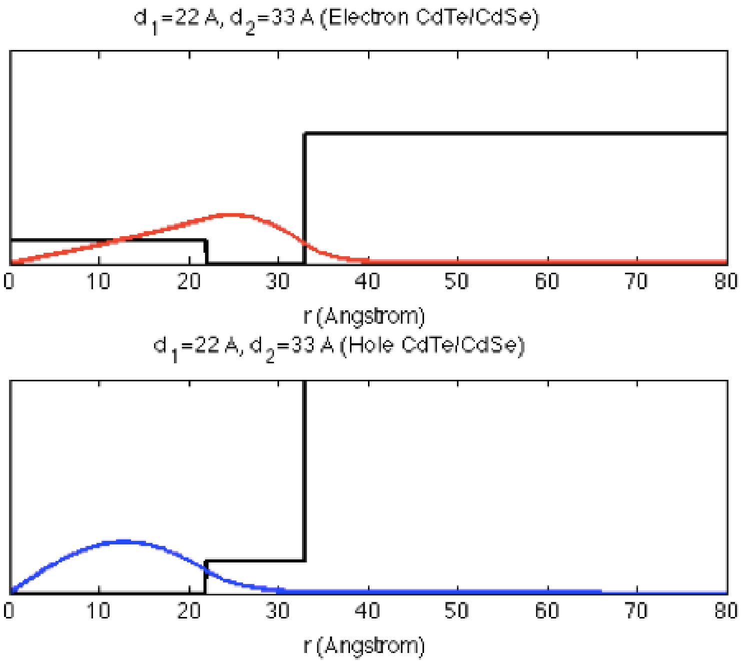


Figure 5.3: Wavefunctions of electron and hole for CdTe/CdSe (Type-II) 1s-state with parameters $d_1 = 22A$, $d_2 = 33A$.

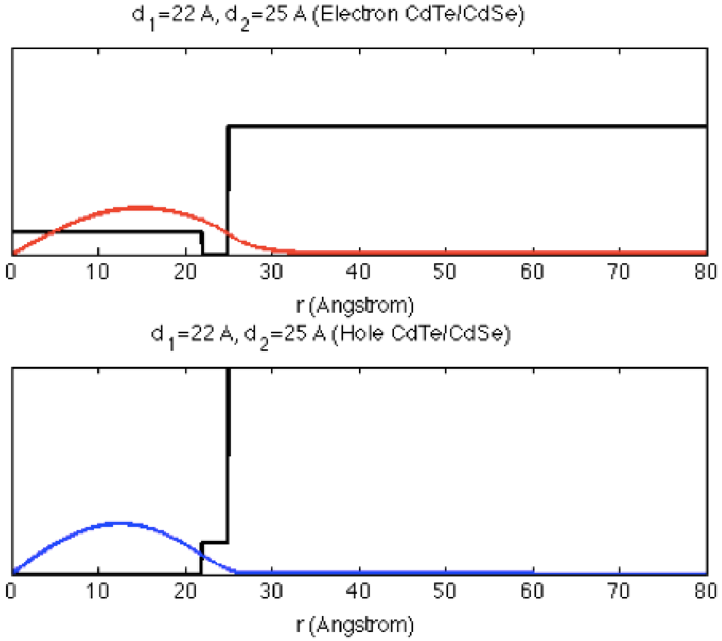


Figure 5.4: Wavefunctions of electron and hole for CdTe/CdSe (Type-II) 1s-state with parameters $d_1 = 22A$, $d_2 = 25A$.

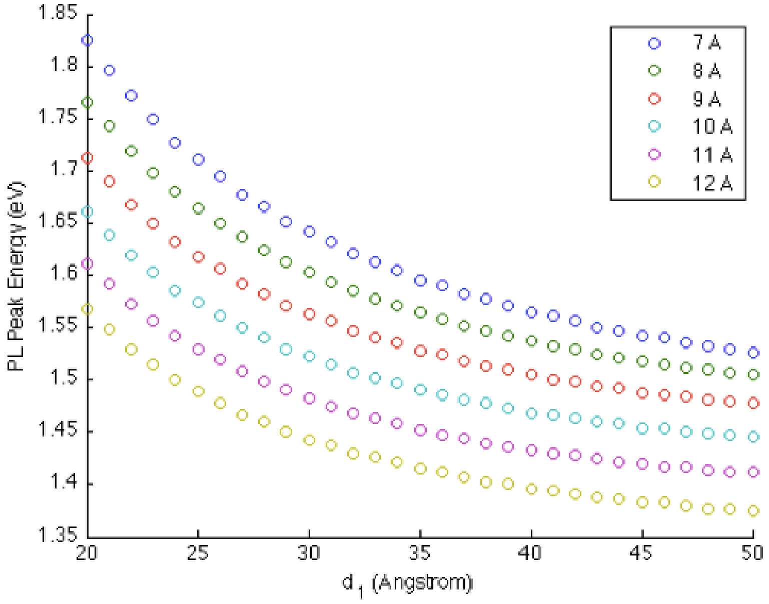


Figure 5.5: Exciton energy of CdTe/CdSe (Type-II) with respect to d_1 for different shell thicknesses.

5.2 Oscillator Strength of Excitons Confined in Type I/Type II Spherical QDs

Oscillator strength is proportional to the interaction strength of the quantum dot with the external electromagnetic field. It is very important to have a high oscillator strength for the applications that utilize spontaneous emission. In cavity applications, oscillator strength determines the strength of interaction between the cavity field and the quantum dot [43].

In theoretical calculations, the oscillator strength is proportional to the overlap integral of the charge carriers [44]:

$$f = \frac{E_p}{2E_{exc}} \mid \int d^3r\psi_e(r)\psi_h(r) \mid^2 \tag{5.1}$$

where E_p is the Kane energy; E_{exc} is the exciton energy; ψ_e and ψ_h are the wavefunctions of the electron and hole, respectively.

The oscillator strength of the Type-I heterostructure increases with increasing core radius for all shell thicknesses as we see in Figure 5.6. However, in the Type-II heterostructure, when the shell thickness is high, increasing the core radius decreases the oscillator strength. For small shell thicknesses of Type-II structure, increasing the core radius increases the oscillator strength. These different regimes can be seen in Figure 5.7. This can be explained by the above discussion that small shell thicknesses cause electron and hole to localize in the same spatial location like in the Type-I structure. That is why the oscillator strength behaves as in the Type-I structure for small shell thicknesses of the Type-II structure.

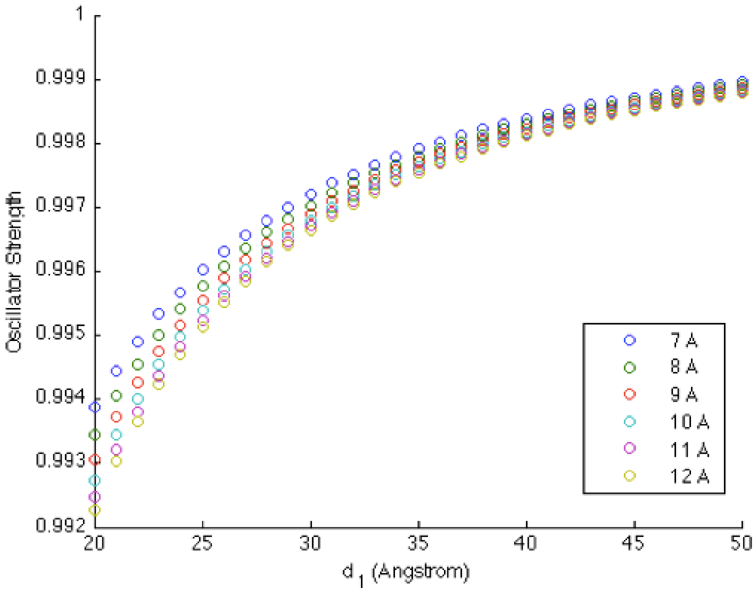


Figure 5.6: Oscillator strength of CdSe/ZnS (Type-I) with respect to d_1 for different shell thicknesses.

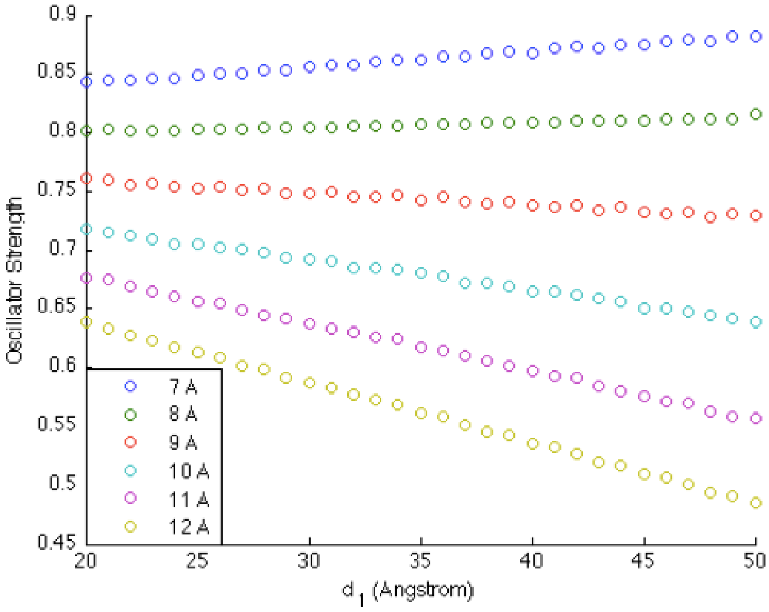


Figure 5.7: Oscillator strength of CdTe/CdSe (Type-II) with respect to d_1 for different shell thicknesses.

5.3 Biexciton Binding Energies for Type I/-Type II Spherical QDs

Since electron and hole energies of the second exciton are higher than the first exciton, if exciton-exciton interaction is not strong, amplified spontaneous emission peak (i.e., energy of the second exciton in the biexction) will be blue-shifted with respect to the spontaneous emission peak (i.e., energy of a single exciton).

Since exciton-exciton interaction is attractive, it decreases the total energy of the biexciton. Hence, as exciton-exciton interaction gets stronger amount of blue-shift decreases. When there are very strong exciton-exciton interaction, amplified spontaneous emission peak may become red-shifted.

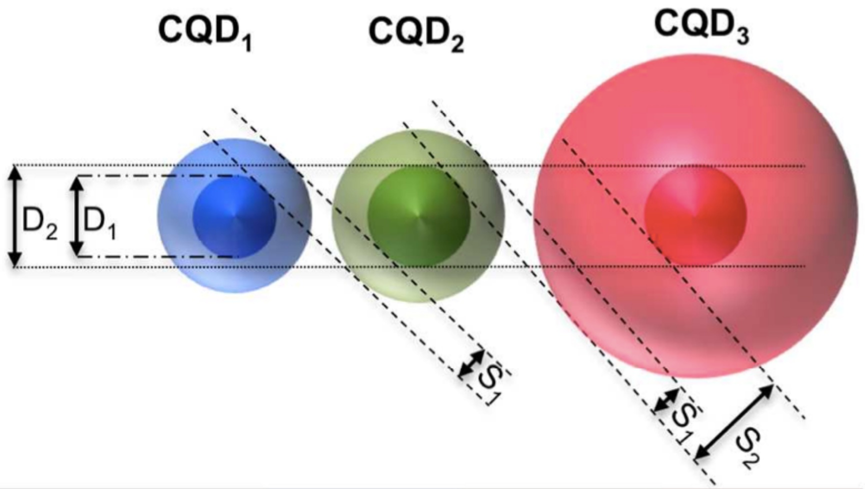
In Type-I quantum dots, since excitons and holes confine in the same spatial region, there are very strong exciton-exciton interaction. Hence, ASE peak become red-shifted due to decrease in the total biexciton energy.

In Type-II quantum dots, excitons and holes may be confined in different spatial regions. This causes weak exciton-exciton interaction. Hence, in large Type-II quantum dots, ASE peak will be blue-shifted. As the quantum dot size decreases, since electrons and holes get closer, the amount of blue-shift in ASE peak decreases.

If the core and shell sizes of a quantum dot is tuned appropriately, electrons and holes might localize very closely. In this situation, ASE peak may become red-shifted. These kind of quantum dots are called quasi Type-II since they may exhibit both Type-I-like and Type-II-like quantum dot behaviour.

In this thesis, we calculated for which core and shell sizes quantum dots show Type-I or Type-II behaviour. This tunability is also confirmed with the experimental results.

In Figure 5.9 we show our simulation results for CdSe/CdS core/shell quantum dots. Experimentally we verified these results in our group by synthesizing three



	CQD_1	CQD_2	CQD_3
Core Diameter	D_1=2.6 nm	D_2=3.2 nm	D_2=3.2 nm
Shell Thickness	S_1=1.1 nm	S_1=1.1 nm	S_2=3.5 nm

Figure 5.8: Schematics of the quantum dots synthesized.

different sizes of CdSe/CdS quantum dots. These dots are shown in Figure 5.8. Sizes of these quantum dots are marked using red-stars in Figure 5.9. These core/shell sizes are selected such that they give blue-shift, red-shift and no-shift in their ASE spectra. Their spectra under intense two-photon optical excitation and spectra obtained from the simulation results are given in Figures 5.10 and 5.11, respectively. Simulation results are in good agreement with the experimental results.

The calculated electron and hole wavefunctions for the two colloidal quantum dot (CQD) sizes of CQD_1 and CQD_3 samples exhibiting quasi Type-II and Type-II behaviors, respectively, are provided in Figures 5.12 and 5.13. The clear delocalization difference of electrons for these two samples is a very good example of the Type-tunability feature of CdSe/CdS CQDs.

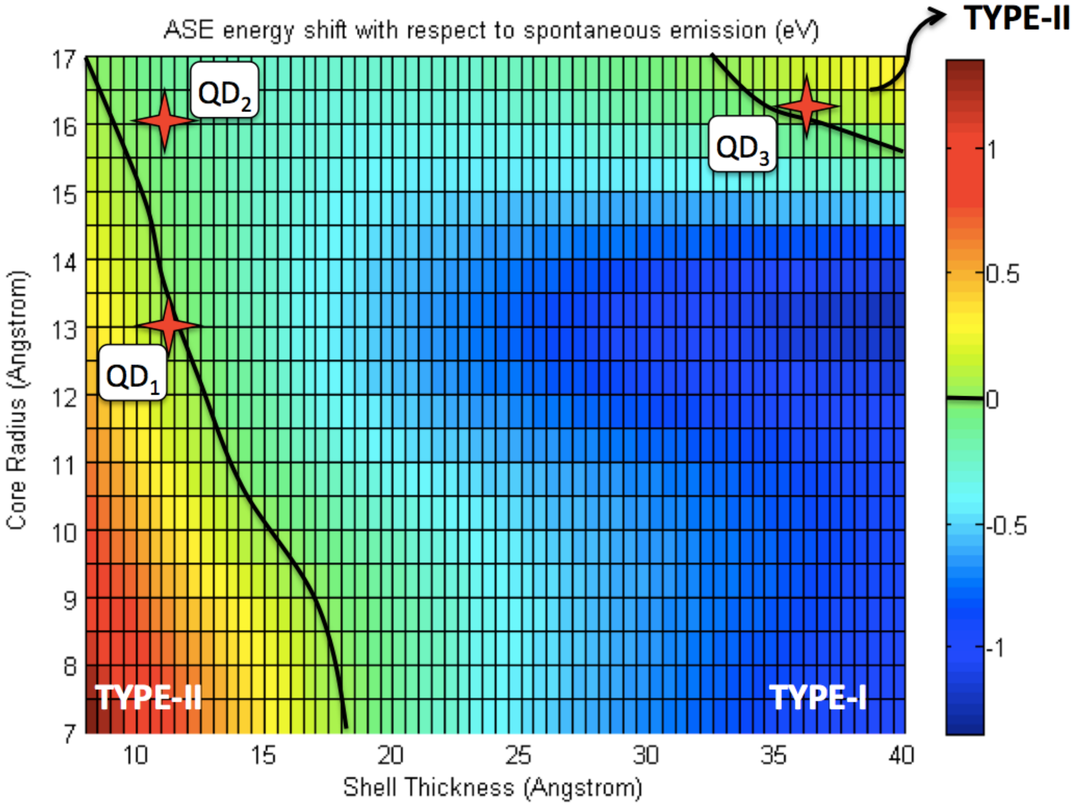


Figure 5.9: ASE peak shift with respect to spontaneous emission for different core/shell sizes of CdSe/CdS core/shell quantum dots.

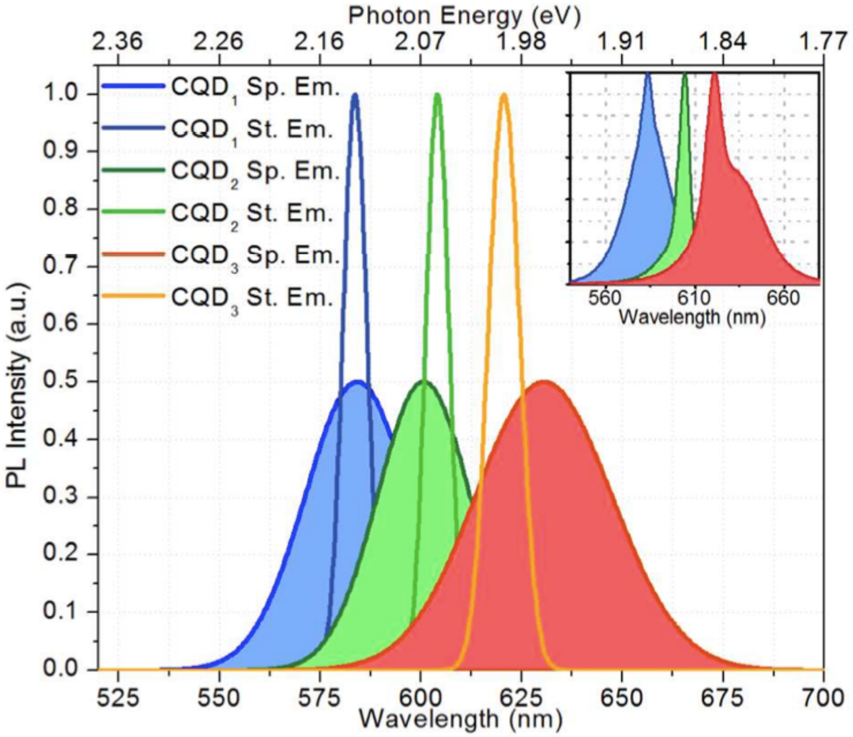


Figure 5.10: Experimental ASE spectra of the quantum dots synthesized.

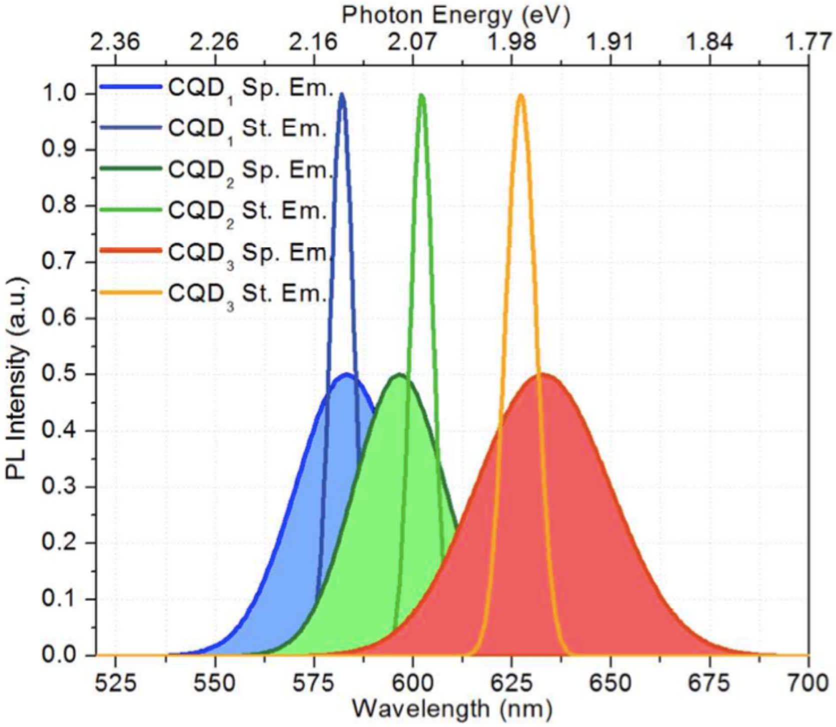


Figure 5.11: Simulation results for ASE spectra of the quantum dots synthesized.

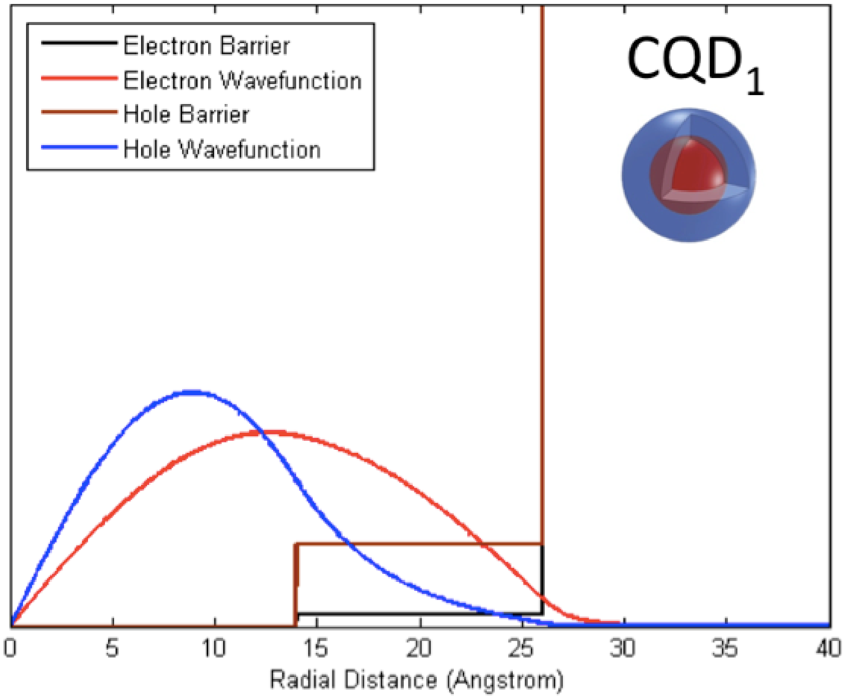


Figure 5.12: The calculated wavefunctions of electrons and holes for CQD_1.

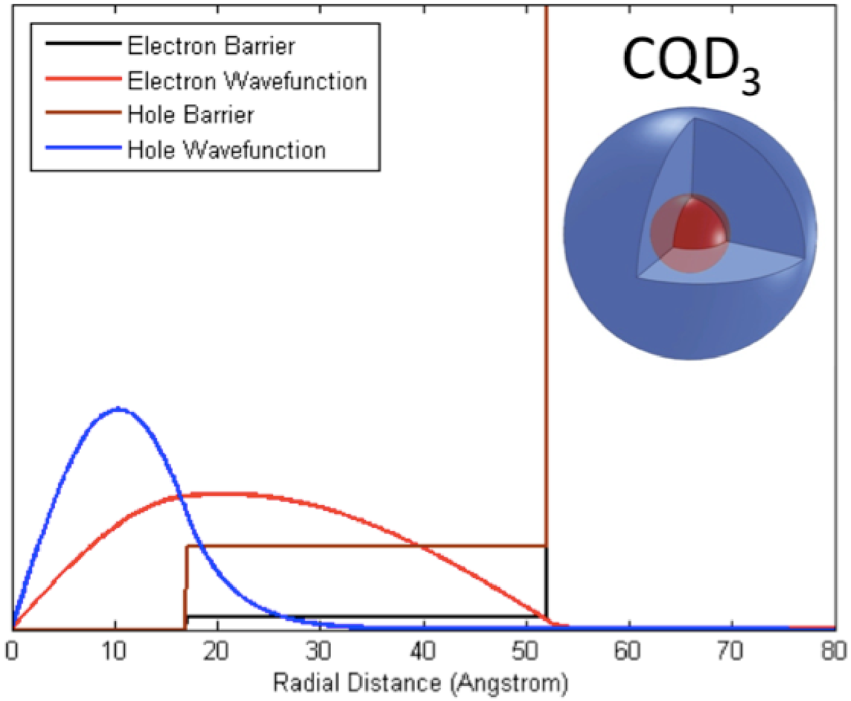


Figure 5.13: The calculated wavefunctions of electrons and holes for CQD_3.

Chapter 6

CONCLUSION

In this thesis, we developed quantum mechanical models necessary for simulating various nanocrystal quantum dot-quantum well heterostructures and solved them using analytical and numerical techniques. These results are consistent with experimental findings.

We implemented quantum mechanical simulation techniques to solve the Schrodinger equation of multi-particle systems self-consistently. In order to solve the Poisson equations arising from the multi-particle Schrodinger equation, we used the multigrid technique. Multigrid technique allowed us to solve Poisson equation very efficiently. This efficiency allowed us to make a large scale parametrical study.

We demonstrated the effects of the different spatial carrier localizations arising from Type-II heterostructures. In Type-II heterostructures, we observed charge carrier localization at different spatial locations. This allows us to tune wavefunctions of electrons and holes by changing size parameters of the core and shell.

We showed that we do not have that much control on the individual wavefunctions in Type-I structures.

Also, we found the core/shell dimensions necessary for the Type-I and Type-II biexcitonic behaviors of CdSe/CdS nanocrystal quantum dots by numerical calculations.

Using quantum mechanical simulations we demonstrated that it is possible to get attractive or repulsive biexcitonic interaction by changing the core and shell size in the same material system. We confirmed these results by amplified spontaneous emission experiments.

Contributions

[1] M. Şahin, S. Nizamoglu, **O. Yerli**, and H. V. Demir, “Reordering orbitals of semiconductor multi-shell quantum dot-quantum well heteronanocrystals,” Journal of Applied Physics, vol. 111, no. 2, pp. 023713–023713–6, Jan. 2012.

[2] A. F. Cihan, Y. Kelestemur, B. Guzelturk, **O. Yerli**, U. Kurum, H. G. Yaglioglu, A. Elmali, and H. V. Demir, “Attractive versus repulsive excitonic interactions of colloidal quantum dots control blue- to red-shifting (and non-shifting) amplified spontaneous emission”, submitted.

Bibliography

[1] S. Nizamoglu and H. V. Demir, "Onion-like (CdSe)ZnS/CdSe/ZnS quantum-dot-quantum-well heteronanocrystals for investigation of multi-color emission," *Optics Express*, vol. 16, pp. 3515–3526, Mar. 2008. PMID: 18542444.

[2] "http://www.felicefrankel.com/wp-content/uploads/2012/01/nanocrystals2 .jpg."

[3] C. de Mello Donegá, "Formation of nanoscale spatially indirect excitons: Evolution of the type-II optical character of CdTe/CdSe heteronanocrystals," *Physical Review B*, vol. 81, p. 165303, Apr. 2010.

[4] T. Takagahara, "Biexciton states in semiconductor quantum dots and their nonlinear optical properties," *Physical Review B*, vol. 39, pp. 10206–10231, May 1989.

[5] U. E. H. Laheld and G. T. Einevoll, "Excitons in CdSe quantum dots," *Physical Review B*, vol. 55, pp. 5184–5204, Feb. 1997.

[6] L.-W. Wang and A. Zunger, "Pseudopotential calculations of nanoscale CdSe quantum dots," *Physical Review B*, vol. 53, pp. 9579–9582, Apr. 1996.

[7] V. Shuvayev, L. Deych, I. Ponomarev, and A. Lisyansky, "Self-consistent hartree method for calculations of exciton binding energy in quantum wells," *Superlattices and Microstructures*, vol. 40, pp. 77–92, Aug. 2006.

[8] L. Banyai, Y. Z. Hu, M. Lindberg, and S. W. Koch, "Third-order optical nonlinearities in semiconductor microstructures," *Physical Review B*, vol. 38, pp. 8142–8153, Oct. 1988.

[9] Y. Kayanuma, “Wannier exciton in microcrystals,” *Solid State Communications*, vol. 59, pp. 405–408, Aug. 1986.

[10] S. Nizamoglu, G. Zengin, and H. V. Demir, “Color-converting combinations of nanocrystal emitters for warm-white light generation with high color rendering index,” *Applied Physics Letters*, vol. 92, pp. 031102–031102–3, Jan. 2008.

[11] S. Coe, W.-K. Woo, M. Bawendi, and V. Bulović, “Electroluminescence from single monolayers of nanocrystals in molecular organic devices,” *Nature*, vol. 420, pp. 800–803, Dec. 2002.

[12] Y. Shirasaki, G. J. Supran, M. G. Bawendi, and V. Bulović, “Emergence of colloidal quantum-dot light-emitting technologies,” *Nature Photonics*, vol. 7, pp. 13–23, Jan. 2013.

[13] V. I. Klimov, A. A. Mikhailovsky, S. Xu, A. Malko, J. A. Hollingsworth, C. A. Leatherdale, H.-J. Eisler, and M. G. Bawendi, “Optical gain and stimulated emission in nanocrystal quantum dots,” *Science*, vol. 290, pp. 314–317, Oct. 2000. PMID: 11030645.

[14] R. Krahne, M. Zavelani-Rossi, M. G. Lupo, L. Manna, and G. Lanzani, “Amplified spontaneous emission from core and shell transitions in CdSe/CdS nanorods fabricated by seeded growth,” *Applied Physics Letters*, vol. 98, pp. 063105–063105–3, Feb. 2011.

[15] Y. Liao, G. Xing, N. Mishra, T. C. Sum, and Y. Chan, “Low threshold, amplified spontaneous emission from core-seeded semiconductor nanotetrapods incorporated into a Sol–Gel matrix,” *Advanced Materials*, vol. 24, no. 23, p. OP159–OP164, 2012.

[16] D. BenDaniel and C. Duke, “Space-charge effects on electron tunneling,” *Physical Review*, vol. 152, pp. 683–692, Dec. 1966.

[17] S. Baskoutas and A. F. Terzis, “Size-dependent band gap of colloidal quantum dots,” *Journal of Applied Physics*, vol. 99, pp. 013708–013708–4, Jan. 2006.

[18] S. Kim, B. Fisher, H.-J. Eisler, and M. Bawendi, “Type-II quantum dots: CdTe / CdSe (Core / Shell) and CdSe / ZnTe (Core / Shell) heterostructures,” *Journal of the American Chemical Society*, vol. 125, pp. 11466–11467, Sept. 2003.

[19] F. Garcia-Santamaria, Y. Chen, J. Vela, R. D. Schaller, J. A. Hollingsworth, and V. I. Klimov, “Suppressed auger recombination in ”Giant” nanocrystals boosts optical gain performance,” *Nano Letters*, vol. 9, pp. 3482–3488, Oct. 2009. PMID: 19505082 PMCID: PMC2897714.

[20] Y. Chan, J. S. Steckel, P. T. Snee, J.-M. Caruge, J. M. Hodgkiss, D. G. Nocera, and M. G. Bawendi, “Blue semiconductor nanocrystal laser,” *Applied Physics Letters*, vol. 86, pp. 073102–073102–3, Feb. 2005.

[21] Y. Chen, B. Guilhabert, J. Herrnsdorf, Y. Zhang, A. R. Mackintosh, R. A. Pethrick, E. Gu, N. Laurand, and M. D. Dawson, “Flexible distributed-feedback colloidal quantum dot laser,” *Applied Physics Letters*, vol. 99, pp. 241103–241103–3, Dec. 2011.

[22] Y. Chan, J.-M. Caruge, P. T. Snee, and M. G. Bawendi, “Multiexcitonic two-state lasing in a CdSe nanocrystal laser,” *Applied Physics Letters*, vol. 85, pp. 2460–2462, Sept. 2004.

[23] C. Dang, J. Lee, C. Breen, J. S. Steckel, S. Coe-Sullivan, and A. Nurmikko, “Red, green and blue lasing enabled by single-exciton gain in colloidal quantum dot films,” *Nature Nanotechnology*, vol. 7, pp. 335–339, May 2012.

[24] R. Signorini, I. Fortunati, F. Todescato, S. Gardin, R. Bozio, J. J. Jasieniak, A. Martucci, G. Della Giustina, G. Brusatin, and M. Guglielmi, “Facile production of up-converted quantum dot lasers,” *Nanoscale*, vol. 3, pp. 4109–4113, Oct. 2011. PMID: 21912805.

[25] J. J. Jasieniak, I. Fortunati, S. Gardin, R. Signorini, R. Bozio, A. Martucci, and P. Mulvaney, “Highly efficient amplified stimulated emission from CdSe-CdS-ZnS quantum dot doped waveguides with two-photon infrared optical pumping,” *Advanced Materials*, vol. 20, no. 1, p. 69–73, 2008.

[26] F. Todescato, I. Fortunati, S. Gardin, E. Garbin, E. Collini, R. Bozio, J. J. Jasieniak, G. Della Giustina, G. Brusatin, S. Toffanin, and R. Signorini, “Soft-lithographed up-converted distributed feedback visible lasers based on CdSe–CdZnS–ZnS quantum dots,” *Advanced Functional Materials*, vol. 22, no. 2, p. 337–344, 2012.

[27] C. Zhang, F. Zhang, T. Zhu, A. Cheng, J. Xu, Q. Zhang, S. E. Mohney, R. H. Henderson, and Y. A. Wang, “Two-photon-pumped lasing from colloidal nanocrystal quantum dots,” *Optics Letters*, vol. 33, pp. 2437–2439, Nov. 2008.

[28] “http://nanocluster.mit.edu/research.php.”

[29] J. Nanda, S. A. Ivanov, M. Achermann, I. Bezel, A. Piryatinski, and V. I. Klimov, “Light amplification in the single-exciton regime using Exciton-Exciton repulsion in type-II nanocrystal quantum dots,” *The Journal of Physical Chemistry C*, vol. 111, pp. 15382–15390, Oct. 2007.

[30] J. Nanda, S. A. Ivanov, H. Htoon, I. Bezel, A. Piryatinski, S. Tretiak, and V. I. Klimov, “Absorption cross sections and auger recombination lifetimes in inverted core-shell nanocrystals: Implications for lasing performance,” *Journal of Applied Physics*, vol. 99, pp. 034309–034309–7, Feb. 2006.

[31] U. E. H. Laheld, F. B. Pedersen, and P. C. Hemmer, “Excitons in type-II quantum dots: Finite offsets,” *Physical Review B*, vol. 52, pp. 2697–2703, July 1995.

[32] A. Pandey and P. Guyot-Sionnest, “Intraband spectroscopy and band offsets of colloidal II-VI core/shell structures,” *The Journal of Chemical Physics*, vol. 127, pp. 104710–104710–10, Sept. 2007.

[33] D. Oron, M. Kazes, and U. Banin, “Multiexcitons in type-II colloidal semiconductor quantum dots,” *Physical Review B*, vol. 75, p. 035330, Jan. 2007.

[34] J. E. Halpert, V. J. Porter, J. P. Zimmer, and M. G. Bawendi, “Synthesis of CdSe/CdTe nanobarbells,” *Journal of the American Chemical Society*, vol. 128, pp. 12590–12591, Oct. 2006.

[35] S. A. Ivanov, A. Piryatinski, J. Nanda, S. Tretiak, K. R. Zavadil, W. O. Wallace, D. Werder, and V. I. Klimov, "Type-II Core/Shell CdS/ZnSe nanocrystals: Synthesis, electronic structures, and spectroscopic properties," *Journal of the American Chemical Society*, vol. 129, pp. 11708–11719, Sept. 2007.

[36] A. Franceschetti, L. W. Wang, G. Bester, and A. Zunger, "Confinement-induced versus correlation-induced electron localization and wave function entanglement in semiconductor nano dumbbells," *Nano Letters*, vol. 6, pp. 1069–1074, May 2006.

[37] E. R. Cohen, *Quantities, Units and Symbols in Physical Chemistry.* Royal Society of Chemistry, 2007.

[38] S. Glutsch, *Excitons in Low-Dimensional Semiconductors: Theory, Numerical Methods, Applications.* Springer, Mar. 2004.

[39] J. Strikwerda, *Finite Difference Schemes and Partial Differential Equations.* SIAM, Sept. 2007.

[40] K. Barnham and D. Vvedensky, *Low-Dimensional Semiconductor Structures: Fundamentals and Device Applications.* Cambridge University Press, Dec. 2008.

[41] W. L. Briggs, V. E. Henson, and S. S. F. McCormick, *A Multigrid Tutorial.* SIAM, 2000.

[42] W.-H. Steeb and Y. Hardy, *Matrix Calculus and Kronecker Product: A Practical Approach to Linear and Multilinear Algebra.* World Scientific, 2011.

[43] M. D. Leistikow, J. Johansen, A. J. Kettelarij, P. Lodahl, and W. L. Vos, "Size-dependent oscillator strength and quantum efficiency of CdSe quantum dots controlled via the local density of states," *Physical Review B*, vol. 79, p. 045301, Jan. 2009.

[44] J. H. Davies, *The physics of low-dimensional semiconductors: an introduction.* Cambridge: Cambridge Univ. Press, 1998.

Appendix A

Codes

```
==> 1_Type_I_Core_Shell/energyElectron.m <==
function energy=energyElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1.05;
    else
```

```
            V(i)=4;
        end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.20))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 1_Type_I_Core_Shell/energyHole.m <==
function energy=energyHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
```

```
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.95;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.47))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 1_Type_I_Core_Shell/energyPlot.m <==
clear;

d1=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d1)
```

```
        energy(j,i)=energyElectron(d1(i),d1(i)+shell(j))+
            energyHole(d1(i),d1(i)+shell(j))+0.77;
        j
end
end

% plot
scatter(d1,energy(1,:),'DisplayName','7␣A');
hold on
scatter(d1,energy(2,:),'DisplayName','8␣A');
scatter(d1,energy(3,:),'DisplayName','9␣A');
scatter(d1,energy(4,:),'DisplayName','10␣A');
scatter(d1,energy(5,:),'DisplayName','11␣A');
scatter(d1,energy(6,:),'DisplayName','12␣A');

% Create xlabel
xlabel({'d_1␣(Angstrom)'});

% Create ylabel
ylabel({'PL␣Peak␣Energy␣(eV)'});
==> 1_Type_I_Core_Shell/oscillator.m <==
function strength=oscillator(dd1,dd2)
clf;
hold on;
waveE=oscillatorElectron(dd1,dd2);
waveH=oscillatorHole(dd1,dd2);
wave=abs(waveE).*abs(waveH);

strength=trapz(wave);
==> 1_Type_I_Core_Shell/oscillatorElectron.m <==
function wavefx=oscillatorElectron(dd1,dd2)

% Parameter
```

```
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1.05;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.20))./h^2 + sparse(diag(V));
```

```
[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
%plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot

%xlim([0 80]);
%ylim([0 0.18]);
%title(sprintf('d_1=%d A, d_2=%d A (Electron CdSe/ZnSe)',
    dd1,dd2));
%xlabel('r (Angstrom)');
%set(gca,'YTick',zeros(1,0));
==> 1_Type_I_Core_Shell/oscillatorHole.m <==
function wavefx=oscillatorHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);
```

```
h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.95;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.47))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);
```

```
% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
%plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot

%xlim([0 80]);
%ylim([0 0.18]);
%title(sprintf('d_1=%d A, d_2=%d A (Hole CdSe/ZnSe)',dd1,
    dd2));
%xlabel('r (Angstrom)');
%set(gca,'YTick',zeros(1,0));
==> 1_Type_I_Core_Shell/oscillatorPlot.m <==
clear;

d1=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d1)
     oscillators(j,i)=oscillator(d1(i),d1(i)+shell(j));
     j
end
end

% plot
clf;
```

```
hold on;
scatter(d1,oscillators(1,:),'DisplayName','7␣A');
scatter(d1,oscillators(2,:),'DisplayName','8␣A');
scatter(d1,oscillators(3,:),'DisplayName','9␣A');
scatter(d1,oscillators(4,:),'DisplayName','10␣A');
scatter(d1,oscillators(5,:),'DisplayName','11␣A');
scatter(d1,oscillators(6,:),'DisplayName','12␣A');

% Create xlabel
xlabel({'d_1␣(Angstrom)'});

% Create ylabel
ylabel({'Oscillator␣Strength'});
==> 1_Type_I_Core_Shell/plotElectron.m <==
function plotElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
      V(i)=0;
    elseif x(i)<d2
      V(i)=1.05;
```

```
        else
            V(i)=4;
        end
    end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.20))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot
```

```
xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A␣(Electron␣CdSe/ZnS)',dd1
    ,dd2));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 1_Type_I_Core_Shell/plotHole.m <==
function plotHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.95;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
```

```
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.47))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A␣(Hole␣CdSe/ZnS)',dd1,dd2
    ));
xlabel('r␣(Angstrom)');
```

```
set(gca,'YTick',zeros(1,0));
==> 1_Type_I_Core_Shell/plotWave.m <==
function plotWave(dd1,dd2)
subplot(2,1,1);
plotElectron(dd1,dd2);
subplot(2,1,2);
plotHole(dd1,dd2);

==> 2_Type_II_Core_Shell/energyElectron.m <==
function energy=energyElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0.67;
    elseif x(i)<d2
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
```

```
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 2_Type_II_Core_Shell/energyHole.m <==
function energy=energyHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
     if x(i)<d1
         V(i)=0;
     elseif x(i)<d2
         V(i)=0.97;
```

```
        else
            V(i)=10;
        end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 2_Type_II_Core_Shell/energyPlot.m <==
clear;

d1=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d1)
        energy(j,i)=energyElectron(d1(i),d1(i)+shell(j))+
            energyHole(d1(i),d1(i)+shell(j))+0.77;
        j
end
end
```

```
% plot
scatter(d1,energy(1,:),'DisplayName','7␣A');
hold on
scatter(d1,energy(2,:),'DisplayName','8␣A');
scatter(d1,energy(3,:),'DisplayName','9␣A');
scatter(d1,energy(4,:),'DisplayName','10␣A');
scatter(d1,energy(5,:),'DisplayName','11␣A');
scatter(d1,energy(6,:),'DisplayName','12␣A');

% Create xlabel
xlabel({'d_1␣(Angstrom)'});

% Create ylabel
ylabel({'PL␣Peak␣Energy␣(eV)'});
==> 2_Type_II_Core_Shell/oscillator.m <==
function strength=oscillator(dd1,dd2)
clf;
hold on;
waveE=oscillatorElectron(dd1,dd2);
waveH=oscillatorHole(dd1,dd2);
wave=abs(waveE).*abs(waveH);

strength=trapz(wave);
==> 2_Type_II_Core_Shell/oscillatorElectron.m <==
function wavefx=oscillatorElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);
```

```
h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0.67;
    elseif x(i)<d2
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
```

```
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
%plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot

%xlim([0 80]);
%ylim([0 0.24]);
%title(sprintf('d_1=%d A, d_2=%d A (Electron CdSe/ZnSe)',
    dd1,dd2));
%xlabel('r (Angstrom)');
%set(gca,'YTick',zeros(1,0));
==> 2_Type_II_Core_Shell/oscillatorHole.m <==
function wavefx=oscillatorHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
```

```
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.97;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
```

```
%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
%plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot

%xlim([0 80]);
%ylim([0 0.24]);
%title(sprintf('d_1=%d A, d_2=%d A (Hole CdSe/ZnSe)',dd1,
    dd2));
%xlabel('r (Angstrom)');
%set(gca,'YTick',zeros(1,0));
==> 2_Type_II_Core_Shell/oscillatorPlot.m <==
clear;

d1=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d1)
    oscillators(j,i)=oscillator(d1(i),d1(i)+shell(j));
    j
end
end

% plot
clf;
hold on;
scatter(d1,oscillators(1,:),'DisplayName','7␣A');
scatter(d1,oscillators(2,:),'DisplayName','8␣A');
scatter(d1,oscillators(3,:),'DisplayName','9␣A');
scatter(d1,oscillators(4,:),'DisplayName','10␣A');
scatter(d1,oscillators(5,:),'DisplayName','11␣A');
scatter(d1,oscillators(6,:),'DisplayName','12␣A');

% Create xlabel
xlabel({'d_1␣(Angstrom)'});

% Create ylabel
ylabel({'Oscillator␣Strength'});
==> 2_Type_II_Core_Shell/plotElectron.m <==
function plotElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0.67;
    elseif x(i)<d2
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
```

```
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.24]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A␣(Electron␣CdTe/CdSe)',
    dd1,dd2));
xlabel('r␣(Angstrom)');
```

```
set(gca,'YTick',zeros(1,0));
==> 2_Type_II_Core_Shell/plotHole.m <==
function plotHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.97;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;
```

```
A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.24]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A␣(Hole␣CdTe/CdSe)',dd1,
    dd2));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 2_Type_II_Core_Shell/plotWave.m <==
function plotWave(dd1,dd2)
subplot(2,1,1);
plotElectron(dd1,dd2);
subplot(2,1,2);
```

```
plotHole(dd1,dd2);

==> 3_Type_I_Core_Shell_Shell/energyElectron.m <==
function energy=energyElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1.05;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
```

```
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.20))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 3_Type_I_Core_Shell_Shell/energyHole.m <==
function energy=energyHole(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
     if x(i)<d1
         V(i)=0;
     elseif x(i)<d2
         V(i)=0.95;
     elseif x(i)<d3
         V(i)=0;
```

```
        else
            V(i)=10;
        end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.47))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 3_Type_I_Core_Shell_Shell/energyPlot.m <==
clear;

d1=10;
d2=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d2)
      energy(j,i)=energyElectron(d1,d2(i),d2(i)+shell(j))+
          energyHole(d1,d2(i),d2(i)+shell(j))+0.77;
      j
end
end
```

```
% plot
scatter(d2,energy(1,:),'DisplayName','7␣A');
hold on
scatter(d2,energy(2,:),'DisplayName','8␣A');
scatter(d2,energy(3,:),'DisplayName','9␣A');
scatter(d2,energy(4,:),'DisplayName','10␣A');
scatter(d2,energy(5,:),'DisplayName','11␣A');
scatter(d2,energy(6,:),'DisplayName','12␣A');

title('d1=10␣A');
% Create xlabel
xlabel({'d_2␣(Angstrom)'});

% Create ylabel
ylabel({'PL␣Peak␣Energy␣(eV)'});
==> 3_Type_I_Core_Shell_Shell/oscillator.m <==
function strength=oscillator(dd1,dd2,dd3)
clf;
hold on;
waveE=oscillatorElectron(dd1,dd2,dd3);
waveH=oscillatorHole(dd1,dd2,dd3);
wave=abs(waveE).*abs(waveH);

strength=trapz(wave);
==> 3_Type_I_Core_Shell_Shell/oscillatorElectron.m <==
function wavefx=oscillatorElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);
```

```
% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1.05;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.20))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
```

```
% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 3_Type_I_Core_Shell_Shell/oscillatorHole.m <==
function wavefx=oscillatorHole(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.95;
    elseif x(i)<d3
        V(i)=0;
    else
```

```
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.47))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 3_Type_I_Core_Shell_Shell/oscillatorPlot.m <==
clear;

d1=10;
d2=20:50;
shell=7:12;
```

```
for j=1:length(shell)
for i=1:length(d2)
    oscillators(j,i)=oscillator(d1,d2(i),d2(i)+shell(j));
    j
end
end

% plot
clf;
hold on;
scatter(d2,oscillators(1,:),'DisplayName','7␣A');
scatter(d2,oscillators(2,:),'DisplayName','8␣A');
scatter(d2,oscillators(3,:),'DisplayName','9␣A');
scatter(d2,oscillators(4,:),'DisplayName','10␣A');
scatter(d2,oscillators(5,:),'DisplayName','11␣A');
scatter(d2,oscillators(6,:),'DisplayName','12␣A');

title('d1=10␣A');
% Create xlabel
xlabel({'d_2␣(Angstrom)'});

% Create ylabel
ylabel({'Oscillator␣Strength'});
==> 3_Type_I_Core_Shell_Shell/plotElectron.m <==
function plotElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);
```

```
h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1.05;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.20))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
```

```
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A,␣d_3=%d␣A␣(Electron␣CdSe
    /ZnS/CdSe)',dd1,dd2,dd3));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 3_Type_I_Core_Shell_Shell/plotHole.m <==
function plotHole(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
```

```
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.95;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
   Laplacian matrix
H= -(A1D./(2*0.47))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
   /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
```

```
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
   wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A,␣d_3=%d␣A␣(Hole␣CdSe/ZnS
   /CdSe)',dd1,dd2,dd3));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 3_Type_I_Core_Shell_Shell/plotWave.m <==
function plotWave(dd1,dd2,dd3)
subplot(2,1,1);
plotElectron(dd1,dd2,dd3);
subplot(2,1,2);
plotHole(dd1,dd2,dd3);

==> 4_Type_II_Core_Shell_Shell/energyElectron.m <==
function energy=energyElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);
```

```
h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0.67;
    elseif x(i)<d2
        V(i)=0;
    elseif x(i)<d3
        V(i)=0.67;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 4_Type_II_Core_Shell_Shell/energyHole.m <==
function energy=energyHole(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.97;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;
```

```
A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 4_Type_II_Core_Shell_Shell/energyPlot.m <==
clear;

d1=10;
d2=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d2)
     energy(j,i)=energyElectron(d1,d2(i),d2(i)+shell(j))+
         energyHole(d1,d2(i),d2(i)+shell(j))+0.77;
     j
end
end

% plot
scatter(d2,energy(1,:),'DisplayName','7␣A');
hold on
scatter(d2,energy(2,:),'DisplayName','8␣A');
scatter(d2,energy(3,:),'DisplayName','9␣A');
scatter(d2,energy(4,:),'DisplayName','10␣A');
scatter(d2,energy(5,:),'DisplayName','11␣A');
scatter(d2,energy(6,:),'DisplayName','12␣A');

title('d1=10␣A');
```

```
% Create xlabel
xlabel({'d_2 (Angstrom)'});

% Create ylabel
ylabel({'PL Peak Energy (eV)'});
==> 4_Type_II_Core_Shell_Shell/oscillator.m <==
function strength=oscillator(dd1,dd2,dd3)
clf;
hold on;
waveE=oscillatorElectron(dd1,dd2,dd3);
waveH=oscillatorHole(dd1,dd2,dd3);
wave=abs(waveE).*abs(waveH);

strength=trapz(wave);
==> 4_Type_II_Core_Shell_Shell/oscillatorElectron.m <==
function wavefx=oscillatorElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
       V(i)=0.67;
    elseif x(i)<d2
        V(i)=0;
    elseif x(i)<d3
        V(i)=0.67;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 4_Type_II_Core_Shell_Shell/oscillatorHole.m <==
function wavefx=oscillatorHole(dd1,dd2,dd3)
```

```
% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.97;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;
```

```
A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 4_Type_II_Core_Shell_Shell/oscillatorPlot.m <==
clear;

d1=10;
d2=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d2)
     oscillators(j,i)=oscillator(d1,d2(i),d2(i)+shell(j));
     j
end
end

% plot
clf;
hold on;
scatter(d2,oscillators(1,:),'DisplayName','7␣A');
scatter(d2,oscillators(2,:),'DisplayName','8␣A');
scatter(d2,oscillators(3,:),'DisplayName','9␣A');
scatter(d2,oscillators(4,:),'DisplayName','10␣A');
scatter(d2,oscillators(5,:),'DisplayName','11␣A');
scatter(d2,oscillators(6,:),'DisplayName','12␣A');

title('d1=10␣A');
% Create xlabel
xlabel({'d_2␣(Angstrom)'});

% Create ylabel
ylabel({'Oscillator␣Strength'});
==> 4_Type_II_Core_Shell_Shell/plotElectron.m <==
function plotElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0.67;
    elseif x(i)<d2
        V(i)=0;
    elseif x(i)<d3
      V(i)=0.67;
    else
      V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues/eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial wavefunction by dividing coordinate
```

```
% Plot
plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A,␣d_3=%d␣A␣(Electron␣CdTe
    /CdSe/CdTe)',dd1,dd2,dd3));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 4_Type_II_Core_Shell_Shell/plotHole.m <==
function plotHole(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.97;
    elseif x(i)<d3
        V(i)=0;
    else
```

```
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot
```

```
xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A,␣d_3=%d␣A␣(Hole␣CdTe/CdSe/CdTe)',dd1,dd2,dd3));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 4_Type_II_Core_Shell_Shell/plotWave.m <==
function plotWave(dd1,dd2,dd3)
subplot(2,1,1);
plotElectron(dd1,dd2,dd3);
subplot(2,1,2);
plotHole(dd1,dd2,dd3);

==> 5_Three_Material/energyElectron.m <==
function energy=energyElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=2;
    elseif x(i)<d2
        V(i)=1;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 5_Three_Material/energyHole.m <==
function energy=energyHole(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
```

```
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
   Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
   /eigenvectors
energy=E(5,5)*27.2116;
==> 5_Three_Material/energyPlot.m <==
clear;
```

```
d1=10;
d2=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d2)
    energy(j,i)=energyElectron(d1,d2(i),d2(i)+shell(j))+
        energyHole(d1,d2(i),d2(i)+shell(j))+0.77;
    j
end
end

% plot
scatter(d2,energy(1,:),'DisplayName','7 A');
hold on
scatter(d2,energy(2,:),'DisplayName','8 A');
scatter(d2,energy(3,:),'DisplayName','9 A');
scatter(d2,energy(4,:),'DisplayName','10 A');
scatter(d2,energy(5,:),'DisplayName','11 A');
scatter(d2,energy(6,:),'DisplayName','12 A');

title('d1=10 A');
% Create xlabel
xlabel({'d_2 (Angstrom)'});

% Create ylabel
ylabel({'PL Peak Energy (eV)'});
==> 5_Three_Material/energyPlotNew.m <==
clear;
t=5:35; % core thickness
s=5:20; % first shell thickness

for i=1:length(s)
```

```
    energy(i,:)=arrayfun(@(x)energyHole(x, x+s(i), x+s(i)
        +20), t)+arrayfun(@(x)energyElectron(x, x+s(i), x+s(
        i)+20), t);
    i
end

clf;
hold all;
for i=1:length(s)
    plot(t,energy(i,:));
end
==> 5_Three_Material/oscillator.m <==
function strength=oscillator(dd1,dd2,dd3)
clf;
hold on;
waveE=oscillatorElectron(dd1,dd2,dd3);
waveH=oscillatorHole(dd1,dd2,dd3);
wave=abs(waveE).*abs(waveH);

strength=trapz(wave);
==> 5_Three_Material/oscillatorElectron.m <==
function wavefx=oscillatorElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
```

```
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=2;
    elseif x(i)<d2
        V(i)=1;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
   Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
   /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
```

```
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 5_Three_Material/oscillatorHole.m <==
function wavefx=oscillatorHole(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
```

```
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 5_Three_Material/oscillatorPlot.m <==
clear;

d1=10;
d2=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d2)
     oscillators(j,i)=oscillator(d1,d2(i),d2(i)+shell(j));
     j
end
```

```
end

% plot
clf;
hold on;
scatter(d2,oscillators(1,:),'DisplayName','7␣A');
scatter(d2,oscillators(2,:),'DisplayName','8␣A');
scatter(d2,oscillators(3,:),'DisplayName','9␣A');
scatter(d2,oscillators(4,:),'DisplayName','10␣A');
scatter(d2,oscillators(5,:),'DisplayName','11␣A');
scatter(d2,oscillators(6,:),'DisplayName','12␣A');

title('d1=10␣A');
% Create xlabel
xlabel({'d_2␣(Angstrom)'});

% Create ylabel
ylabel({'Oscillator␣Strength'});
==> 5_Three_Material/oscillatorPlotNew.m <==
clear;
t=5:35; % core thickness
s=5:20; % first shell thickness

for i=1:length(s)
    osc(i,:)=arrayfun(@(x)oscillator(x, x+s(i), x+s(i)+20),
         t);
end

clf;
hold all;
for i=1:length(s)
    plot(t,osc(i,:));
end
```

```
==> 5_Three_Material/plotElectron.m <==
function plotElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=2;
    elseif x(i)<d2
        V(i)=1;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
```

```
hold on;

A1D = spdiags(ones(n,1)*[1 −2 1],−1:1,n,n);   % 1d
    Laplacian matrix
H= −(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A,␣d_3=%d␣A␣(Electron)',
    dd1,dd2,dd3));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 5_Three_Material/plotHole.m <==
function plotHole(dd1,dd2,dd3)
```

```
% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;
```

```
A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d A, d_2=%d A, d_3=%d A (Hole)',dd1,
    dd2,dd3));
xlabel('r (Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 5_Three_Material/plotWave.m <==
function plotWave(dd1,dd2,dd3)
subplot(2,1,1);
plotElectron(dd1,dd2,dd3);
subplot(2,1,2);
```

```
plotHole(dd1,dd2,dd3);

==> 6_Angular_Momentum/energyElectron.m <==
function energy=energyElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0.67;
    elseif x(i)<d2
        V(i)=0;
    elseif x(i)<d3
        V(i)=0.67;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
```

```
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 6_Angular_Momentum/energyHole.m <==
function energy=energyHole(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.97;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
==> 6_Angular_Momentum/energyPlot.m <==
clear;

d1=10;
d2=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d2)
    energy(j,i)=energyElectron(d1,d2(i),d2(i)+shell(j))+
        energyHole(d1,d2(i),d2(i)+shell(j))+0.77;
    j
end
end
```

```
% plot
scatter(d2,energy(1,:),'DisplayName','7␣A');
hold on
scatter(d2,energy(2,:),'DisplayName','8␣A');
scatter(d2,energy(3,:),'DisplayName','9␣A');
scatter(d2,energy(4,:),'DisplayName','10␣A');
scatter(d2,energy(5,:),'DisplayName','11␣A');
scatter(d2,energy(6,:),'DisplayName','12␣A');

title('d1=10␣A');
% Create xlabel
xlabel({'d_2␣(Angstrom)'});

% Create ylabel
ylabel({'PL␣Peak␣Energy␣(eV)'});
==> 6_Angular_Momentum/oscillator.m <==
function strength=oscillator(dd1,dd2,dd3)
clf;
hold on;
waveE=oscillatorElectron(dd1,dd2,dd3);
waveH=oscillatorHole(dd1,dd2,dd3);
wave=abs(waveE).*abs(waveH);

strength=trapz(wave);
==> 6_Angular_Momentum/oscillatorElectron.m <==
function wavefx=oscillatorElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);
```

```
% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0.67;
    elseif x(i)<d2
        V(i)=0;
    elseif x(i)<d3
        V(i)=0.67;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
```

```
% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 6_Angular_Momentum/oscillatorHole.m <==
function wavefx=oscillatorHole(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.97;
    elseif x(i)<d3
        V(i)=0;
    else
```

```
            V(i)=10;
        end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 6_Angular_Momentum/oscillatorPlot.m <==
clear;

d1=10;
d2=20:50;
shell=7:12;
```

```
for j=1:length(shell)
for i=1:length(d2)
    oscillators(j,i)=oscillator(d1,d2(i),d2(i)+shell(j));
    j
end
end

% plot
clf;
hold on;
scatter(d2,oscillators(1,:),'DisplayName','7␣A');
scatter(d2,oscillators(2,:),'DisplayName','8␣A');
scatter(d2,oscillators(3,:),'DisplayName','9␣A');
scatter(d2,oscillators(4,:),'DisplayName','10␣A');
scatter(d2,oscillators(5,:),'DisplayName','11␣A');
scatter(d2,oscillators(6,:),'DisplayName','12␣A');

title('d1=10␣A');
% Create xlabel
xlabel({'d_2␣(Angstrom)'});

% Create ylabel
ylabel({'Oscillator␣Strength'});
==> 6_Angular_Momentum/plotElectron.m <==
function plotElectron(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
last=(dd1+dd2+dd3+100)/(0.5292);
```

```
h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.67;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=3;
    end
end

for i=1:n
    V(i)=V(i)+10/(x(i)*x(i));
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

% A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
   Laplacian matrix
% H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));
%
```

```
% [U,E] = eigs(H,5,'sm'); % calculate 5 smallest
   eigenvalues/eigenvectors
%
% % Take wavefunction
% wavefx=U(:,5);
%
% % Integrate
% A=trapz(wavefx.*wavefx);
%
% % Normalize
% wavefx=wavefx./(A);
%
% %wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
   wavefunction by dividing coordinate
%
% % Plot
% plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot
%
xlim([0 80]);
ylim([0 0.12]);
% title(sprintf('d_1=%d A, d_2=%d A, d_3=%d A (Electron
   CdTe/CdSe/CdTe)',dd1,dd2,dd3));
% xlabel('r (Angstrom)');
% set(gca,'YTick',zeros(1,0));
==> 6_Angular_Momentum/plotHole.m <==
function plotHole(dd1,dd2,dd3)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
d3=dd3/(0.5292);

% Problem domain
```

```
last=(dd1+dd2+dd3+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.97;
    elseif x(i)<d3
        V(i)=0;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
```

```
% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d A, d_2=%d A, d_3=%d A (Hole CdTe/
    CdSe/CdTe)',dd1,dd2,dd3));
xlabel('r (Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 6_Angular_Momentum/plotWave.m <==
function plotWave(dd1,dd2,dd3)
subplot(2,1,1);
plotElectron(dd1,dd2,dd3);
subplot(2,1,2);
plotHole(dd1,dd2,dd3);

==> 7_Type_I_Core_Shell_Multiexciton/energyElectron.m <==
function [energy wavefx]=energyElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);
```

```
% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1.05;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.20))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;
```

```
% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 7_Type_I_Core_Shell_Multiexciton/energyHole.m <==
function [energy wavefx]=energyHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.95;
    else
        V(i)=10;
    end
end
```

```
% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.47))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 7_Type_I_Core_Shell_Multiexciton/energyMultiexc.m <==
function energy=energyMultiexc(wE,wH)
wave=abs(wE).*abs(wH);
strength=trapz(wave);

energy=0.3*strength;
==> 7_Type_I_Core_Shell_Multiexciton/energyPlot.m <==
clear;
```

```
d1=20:50;
shell=40;

for j=1:length(shell)
for i=1:length(d1)
    [eE wE]=energyElectron(d1(i),d1(i)+shell(j));
    [eH wH]=energyHole(d1(i),d1(i)+shell(j));

    energy(j,i)=eE+eH+0.77;
    energyM(j,i)=eE+eH-energyMultiexc(wE,wH)+0.77;

    energyMultiexc(wE,wH)
    j
end
end

% plot
scatter(d1,energy(1,:),'DisplayName','Single␣Exciton');
hold all
scatter(d1,energyM(1,:),'DisplayName','Biexciton');

% Create xlabel
xlabel({'d_1␣(Angstrom)'});

% Create ylabel
ylabel({'PEnergy␣(eV)'});
==> 7_Type_I_Core_Shell_Multiexciton/oscillator.m <==
function strength=oscillator(dd1,dd2)
clf;
hold on;
waveE=oscillatorElectron(dd1,dd2);
waveH=oscillatorHole(dd1,dd2);
wave=abs(waveE).*abs(waveH);
```

```
strength=trapz(wave);
==> 7_Type_I_Core_Shell_Multiexciton/oscillatorElectron.m
  <==
function wavefx=oscillatorElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1.05;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
```

```
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.20))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
%plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot

%xlim([0 80]);
%ylim([0 0.18]);
%title(sprintf('d_1=%d A, d_2=%d A (Electron CdSe/ZnSe)',
    dd1,dd2));
%xlabel('r (Angstrom)');
%set(gca,'YTick',zeros(1,0));
==> 7_Type_I_Core_Shell_Multiexciton/oscillatorHole.m <==
function wavefx=oscillatorHole(dd1,dd2)
```

```
% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.95;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.47))./h^2 + sparse(diag(V));
```

```
[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial wavefunction by dividing coordinate

% Plot
%plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot

%xlim([0 80]);
%ylim([0 0.18]);
%title(sprintf('d_1=%d A, d_2=%d A (Hole CdSe/ZnSe)',dd1,dd2));
%xlabel('r (Angstrom)');
%set(gca,'YTick',zeros(1,0));
==> 7_Type_I_Core_Shell_Multiexciton/oscillatorPlot.m <==
clear;

d1=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d1)
    oscillators(j,i)=oscillator(d1(i),d1(i)+shell(j));
    j
end
end

% plot
clf;
hold on;
scatter(d1,oscillators(1,:),'DisplayName','7␣A');
scatter(d1,oscillators(2,:),'DisplayName','8␣A');
scatter(d1,oscillators(3,:),'DisplayName','9␣A');
scatter(d1,oscillators(4,:),'DisplayName','10␣A');
scatter(d1,oscillators(5,:),'DisplayName','11␣A');
scatter(d1,oscillators(6,:),'DisplayName','12␣A');

% Create xlabel
xlabel({'d_1␣(Angstrom)'});

% Create ylabel
ylabel({'Oscillator␣Strength'});
==> 7_Type_I_Core_Shell_Multiexciton/plotElectron.m <==
function plotElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points
```

```
V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1.05;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.20))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
```

```
%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
   wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A␣(Electron␣CdSe/ZnS)',dd1
   ,dd2));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 7_Type_I_Core_Shell_Multiexciton/plotHole.m <==
function plotHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.95;
        else
            V(i)=10;
        end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.47))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot
```

```
xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A␣(Hole␣CdSe/ZnS)',dd1,dd2
   ));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 7_Type_I_Core_Shell_Multiexciton/plotWave.m <==
function plotWave(dd1,dd2)
subplot(2,1,1);
plotElectron(dd1,dd2);
subplot(2,1,2);
plotHole(dd1,dd2);

==> 8_Type_II_Core_Shell_Multiexciton/energyElectron.m <==
function [energy wavefx]=energyElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0.67;
    elseif x(i)<d2
        V(i)=0;
        else
            V(i)=4;
        end
    end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
energy=E(5,5)*27.2116;

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 8_Type_II_Core_Shell_Multiexciton/energyHole.m <==
function [energy wavefx]=energyHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
```

```
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=0.97;
    else
        V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors
```

```
energy=E(5,5)*27.2116;

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);
==> 8_Type_II_Core_Shell_Multiexciton/energyMultiexc.m <==
function energy=energyMultiexc(wE,wH)
wave=abs(wE).*abs(wH);
strength=trapz(wave);

energy=0.3*strength;
==> 8_Type_II_Core_Shell_Multiexciton/energyPlot.m <==
clear;

d1=20:50;
shell=30;

for j=1:length(shell)
for i=1:length(d1)
    [eE wE]=energyElectron(d1(i),d1(i)+shell(j));
    [eH wH]=energyHole(d1(i),d1(i)+shell(j));

    energy(j,i)=eE+eH+0.77;
    energyM(j,i)=eE+eH-energyMultiexc(wE,wH)+0.77;

    energyMultiexc(wE,wH)
    j
end
```

```
end

% plot
scatter(d1,energy(1,:),'DisplayName','Single␣Exciton');
hold all
scatter(d1,energyM(1,:),'DisplayName','Biexciton');

% Create xlabel
xlabel({'d_1␣(Angstrom)'});

% Create ylabel
ylabel({'PEnergy␣(eV)'});
==> 8_Type_II_Core_Shell_Multiexciton/oscillator.m <==
function strength=oscillator(dd1,dd2)
clf;
hold on;
waveE=oscillatorElectron(dd1,dd2);
waveH=oscillatorHole(dd1,dd2);
wave=abs(waveE).*abs(waveH);

strength=trapz(wave);
==> 8_Type_II_Core_Shell_Multiexciton/oscillatorElectron.m
    <==
function wavefx=oscillatorElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
```

```
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0;
    elseif x(i)<d2
        V(i)=1.05;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.20))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);
```

```
% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
%plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot

%xlim([0 80]);
%ylim([0 0.18]);
%title(sprintf('d_1=%d A, d_2=%d A (Electron CdSe/ZnSe)',
    dd1,dd2));
%xlabel('r (Angstrom)');
%set(gca,'YTick',zeros(1,0));
==> 8_Type_II_Core_Shell_Multiexciton/oscillatorHole.m <==
function wavefx=oscillatorHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
     if x(i)<d1
```

```
            V(i)=0;
        elseif x(i)<d2
            V(i)=0.95;
        else
            V(i)=10;
        end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
%plot(x*(0.5292),V,'black','LineWidth',2);
%hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);   % 1d
    Laplacian matrix
H= -(A1D./(2*0.47))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate
```

```
% Plot
%plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot

%xlim([0 80]);
%ylim([0 0.18]);
%title(sprintf('d_1=%d A, d_2=%d A (Hole CdSe/ZnSe)',dd1,
    dd2));
%xlabel('r (Angstrom)');
%set(gca,'YTick',zeros(1,0));
==> 8_Type_II_Core_Shell_Multiexciton/oscillatorPlot.m <==
clear;

d1=20:50;
shell=7:12;

for j=1:length(shell)
for i=1:length(d1)
     oscillators(j,i)=oscillator(d1(i),d1(i)+shell(j));
     j
end
end

% plot
clf;
hold on;
scatter(d1,oscillators(1,:),'DisplayName','7␣A');
scatter(d1,oscillators(2,:),'DisplayName','8␣A');
scatter(d1,oscillators(3,:),'DisplayName','9␣A');
scatter(d1,oscillators(4,:),'DisplayName','10␣A');
scatter(d1,oscillators(5,:),'DisplayName','11␣A');
scatter(d1,oscillators(6,:),'DisplayName','12␣A');
```

```
% Create xlabel
xlabel({'d_1 (Angstrom)'});

% Create ylabel
ylabel({'Oscillator Strength'});
==> 8_Type_II_Core_Shell_Multiexciton/plotElectron.m <==
function plotElectron(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
        V(i)=0.67;
    elseif x(i)<d2
        V(i)=0;
    else
        V(i)=4;
    end
end

% Convert from eV to atomic units
V=V./27.2116;
```

```
% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d
    Laplacian matrix
H= -(A1D./(2*0.12))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'red','LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A␣(Electron␣CdSe/ZnS)',dd1
    ,dd2));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 8_Type_II_Core_Shell_Multiexciton/plotHole.m <==
function plotHole(dd1,dd2)

% Parameter
d1=dd1/(0.5292);
d2=dd2/(0.5292);

% Problem domain
last=(dd1+dd2+100)/(0.5292);

h=0.08; % discretization
x=h:h:last; % coordinate vector
n=length(x); % number of discrete points

V=zeros(n,1);
for i=1:n
    if x(i)<d1
      V(i)=0;
    elseif x(i)<d2
      V(i)=0.97;
    else
      V(i)=10;
    end
end

% Convert from eV to atomic units
V=V./27.2116;

% Plot potential
plot(x*(0.5292),V,'black','LineWidth',2);
hold on;

A1D = spdiags(ones(n,1)*[1 -2 1],-1:1,n,n);  % 1d Laplacian matrix
```

```
H= -(A1D./(2*0.32))./h^2 + sparse(diag(V));

[U,E] = eigs(H,5,'sm'); % calculate 5 smallest eigenvalues
    /eigenvectors

% Take wavefunction
wavefx=U(:,5);

% Integrate
A=trapz(wavefx.*wavefx);

% Normalize
wavefx=wavefx./(A);

%wavefx=wavefx./(x(:).*(0.5292)); % convert to radial
    wavefunction by dividing coordinate

% Plot
plot(x*(0.5292),abs(wavefx),'LineWidth',2); % plot

xlim([0 80]);
ylim([0 0.18]);
title(sprintf('d_1=%d␣A,␣d_2=%d␣A␣(Hole␣CdSe/ZnS)',dd1,dd2
    ));
xlabel('r␣(Angstrom)');
set(gca,'YTick',zeros(1,0));
==> 8_Type_II_Core_Shell_Multiexciton/plotWave.m <==
function plotWave(dd1,dd2)
subplot(2,1,1);
plotElectron(dd1,dd2);
subplot(2,1,2);
plotHole(dd1,dd2);

==> GSforMG.m <==
function xG=GSforMG(A,b,xG,iteration)

n=ceil((length(b))^(1/3));

% Gauss-Seidel
D=spdiags(diag(A),0,n^3,n^3);
L=sparse(tril(A)) - D;
U=sparse(triu(A)) - D;

M=D+L;
N=-U;



for i=1:iteration
    xG=M\(N*xG) + M\b;
end


end
==> boundary_QB111.m <==
r = linspace(-2,2,65);
[x,y,z] = meshgrid(r,r,r);

R_111 = zeros(65,65,65);

for i = 1:65
    for j = 1:65
        for k = 1:65

            if abs(x(i,j,k)) <= 1 && abs(y(i,j,k)) <= 1 &&
                abs(z(i,j,k)) <= 1
                R_111(i,j,k) = ((1/2)^(3/2)*cos(pi*x(i,j,k
                    )/2)*cos(pi*y(i,j,k)/2)*cos(pi*z(i,j,k)
                    /2))^2;
                else
                    R_111(i,j,k) = 0;
                end
            end
        end
    end

    slice(x,y,z,R_111,0,0,0)
==> boundary_QB211.m <==
r = linspace(-2,2,65);
[x,y,z] = meshgrid(r,r,r);

R_211 = zeros(65,65,65);

for i = 1:65
    for j = 1:65
        for k = 1:65

            if abs(x(i,j,k)) <= 1 && abs(y(i,j,k)) <= 1 &&
                abs(z(i,j,k)) <= 1
                R_211(i,j,k) = ((1/2)^(3/2)*sin(2*pi*x(i,j
                   ,k)/2)*cos(pi*y(i,j,k)/2)*cos(pi*z(i,j,
                   k)/2))^2;
            else
                R_211(i,j,k) = 0;
            end
        end
    end
end

slice(x,y,z,R_211,0,0,0)
==> boundary_QB221.m <==
r = linspace(-2,2,65);
```

```
[x,y,z] = meshgrid(r,r,r);

R_211 = zeros(65,65,65);

for i = 1:65
    for j = 1:65
        for k = 1:65


            if abs(x(i,j,k)) <= 1 && abs(y(i,j,k)) <= 1 &&
                abs(z(i,j,k)) <= 1
                R_221(i,j,k) = ((1/2)^(3/2)*sin(2*pi*x(i,j
                    ,k)/2)*sin(2*pi*y(i,j,k)/2)*cos(pi*z(i,
                    j,k)/2))^2;
            else
                R_221(i,j,k) = 0;
            end
        end
    end
end

slice(x,y,z,R_221,0,0,0)
==> createA.m <==
function A3D=createA(n)

A1D = spdiags(ones(n,1)*[-1 2 -1],-1:1,n,n);    % 1d
    Laplacian matrix



I1D = speye(size(A1D));                         % 1d
    identity matrix
A2D = kron(A1D,I1D)+kron(I1D,A1D);              % 2d
    Laplacian matrix
```

```
I2D = speye(size(A2D));                                    % 2d
    identity matrix
A3D = kron(A2D,I1D)+kron(I2D,A1D);                         % 3d
    Laplacian matrix

end
==> createAwithEpsilon.m <==
function A=createAwithEpsilon()
A=createA(65);

counter=1;

E_new=zeros(65,65,65);

% Write epsilon values
r = linspace(-2,2,65);
[x,y,z] = meshgrid(r,r,r);

for i = 1:65
     for j = 1:65
          for k = 1:65
               if abs(x(i,j,k)) <= 1 && abs(y(i,j,k)) <= 1 &&
                    abs(z(i,j,k)) <= 1
                    E_new(i,j,k) = 2;
               else
                    E_new(i,j,k) = 50;
               end
          end
     end
end

clear x, clear y,clear z;
```

```
% Update matrix A row by row
for z=1:65
    for y=1:65
        for x=1:65
            if x~=1 && y~=1 && z~=1 && x~=65 && y~=65 && z
               ~=65
                x_plus = 2/(1+E_new(x,y,z)/E_new(x+1,y,z))
                   ;
                x_minus = 2/(1+E_new(x,y,z)/E_new(x-1,y,z)
                   );
                y_plus = 2/(1+E_new(x,y,z)/E_new(x,y+1,z))
                   ;
                y_minus = 2/(1+E_new(x,y,z)/E_new(x,y-1,z)
                   );
                z_plus = 2/(1+E_new(x,y,z)/E_new(x,y,z+1))
                   ;
                z_minus = 2/(1+E_new(x,y,z)/E_new(x,y,z-1)
                   );

                sum=(x_plus + x_minus + y_plus + y_minus +
                    z_plus + z_minus);

                if sum ~= 6
                    disp(['correction' num2str(x) num2str(
                       y) num2str(z)]);

                    testM=zeros(65,65,65);

                    testM(x,y,z)=(x_plus + x_minus +
                       y_plus + y_minus + z_plus + z_minus
                       );
                    testM(x-1,y,z)=-x_minus;
                    testM(x,y-1,z)=-y_minus;
                    testM(x,y,z-1)=-z_minus;

                    testM(x+1,y,z)=-x_plus;
                    testM(x,y+1,z)=-y_plus;
                    testM(x,y,z+1)=-z_plus;

                    testRow=sparse(testM(:)');
                    % Replace the corresponding row of A
                    A(counter,:)=testRow;
                end
            end
            counter=counter+1;
        end
    end
end

end
==> solve_GS.m <==
function xG=solve_GS(A,b,xG)

xExact=A\b;

n=ceil((length(b))^(1/3));


% Gauss-Seidel
tic;
D=spdiags(diag(A),0,n^3,n^3);
L=sparse(tril(A)) - D;
U=sparse(triu(A)) - D;
```

```
M=D+L;
N=-U;

%initial guess
if nargin<3
    xG=zeros(n^3,1);
end

for i=1:10000
    xG=M\(N*xG) + M\b;

    if norm(xExact - xG) < 1e-5
        break;
    end

end
toc;

disp(['iteration:␣' num2str(i)]);

end
==> solve_Jakobi.m <==
function xJ=solve_Jakobi(A,b)

xExact=A\b;

n=ceil((length(b))^(1/3));

% Jakobi
tic;
D=spdiags(diag(A),0,n^3,n^3);
L=sparse(tril(A)) - D;
U=sparse(triu(A)) - D;
```

```
M=D;
N=-(L+U);

xJ=zeros(n^3,1);
for i=1:10000
    xJ=M\(N*xJ) + M\b;

    if norm(xExact - xJ) < 1e-5
        break;
    end

end
toc;

disp(['iteration: ' num2str(i)]);

end
==> solve_MG.m <==
function x=solve_MG(A,b,x)

n=round((length(b))^(1/3))

if n<4
    % If the level is too coarse, solve exactly
    x=A\b;
else
    % Create Restriction and Interpolation matricies
    R1D=sparse(zeros(floor((n-1)/2),n));

    j=1;
    for i=1:(n-1)/2
        R1D(i,j)=1;
        R1D(i,j+1)=2;
        R1D(i,j+2)=1;
        j=j+2;
    end

    R1D=R1D/4;
    R2D=kron(R1D,R1D);

    IN1D=R1D'*2;
    IN2D=kron(IN1D,IN1D);

    R3D=kron(R2D,R1D);
    IN3D=kron(IN2D,IN1D);
    % Done!


    % initial guess
    if nargin<3
        x=zeros(n^3,1);
    end

    % Presmoothing
    x=GSforMG(A,b,x,500);

    res=b-A*x;

    % Restrict residual
    resDown=R3D*res;

    ADown=R3D*A*IN3D;

    % initial guess for error
    errDown=resDown*0;
```

```
    % Solve new system
    for k=1:2
    errDown=solve_MG(ADown,resDown,errDown);
    end
    %errDown=ADown\resDown;

    % Interpolate back the error
    errUp=IN3D*errDown;

    % Correction
    x=x+errUp;

    % Postsmoothing
    x=GSforMG(A,b,x,500);

end
==> solve_SOR.m <==
function xS=solve_SOR(A,b)

xExact=A\b;

n=ceil((length(b))^(1/3));

% SOR
tic;
D=spdiags(diag(A),0,n^3,n^3);
L=sparse(tril(A)) - D;
U=sparse(triu(A)) - D;

M=D+L;
N=-U;
```

```
xS=zeros(n^3,1);
for i=1:10000
    xS=xS + (1.5)*( (M\(N*xS)) + (M\b) - xS);

    if norm(xExact - xS) < 1e-3
        break;
    end

end
toc;

disp(['iteration:␣' num2str(i)]);

end
```