UNIVERSITY OF CALIFORNIA
RIVERSIDE

ICE-CREAM: Integrated Codes to Efficiently Control Real-Time Excitations in Architectured Materials

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Electrical Engineering

by

Simon Sandhofer

September 2022

Thesis Committee:

Dr. Bryan M. Wong, Chairperson
Dr. Ming Liu
Dr. Roger K. Lake

The Thesis of Simon Sandhofer is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am thankful for my advisor, without whose help, I would not have been here.

To my friends and family for all the support.

Contents

List of Figures

Chapter 1

Introduction

The field of nanotechnology is centered around designing, constructing, and manipulating materials with dimensions on the order of several nanometers. At this scale, quantum effects strongly influence the behavior of any given system, leading to new phenomena not observed at larger size scales. As a result, advances in nanotechnology have led to advancements in the fields of medicine, mechanical engineering, electrical engineering, chemistry, and biotechnology.

Moore's law (the observation that the number of transistors on a computer chip roughly doubles every two years) is close to reaching a stopping point. Increases in computing power, brought on by the continuance of Moore's Law, have enabled many of the technologies we rely on today. However, current transistor technology is beginning to reach a limit to its miniturizibility; the dominance of quantum phenomena at smaller scales prevents transistors from operating properly. Since nanotechnology relies on quantum phenomena, it is seen as a potential workaround to the size constraints of current transistor designs.

As such, being able to understand and manipulate the quantum behavior of nanostructured materials is extremely important. The evolution of quantum/nanoscale systems in an external field has been studied quite extensively. However, to effectively design systems making use of quantum phenomena, one wants to find an external field that will drive the evolution of the system along a specified path. The optimization of external control fields forms is the main goal of quantum optimal control. Previous work in quantum optimal control has primarily been used to simulate systems with simple geometry[1][2]. This is ultimately due to the limitations of the basis used to solve the time-dependent Schrödinger equation. A common choice of basis is the finite-difference method, which uses an equidistant grid of points to discretize the spatial components of the time-dependent Schrödinger equation. However, the finite-difference method struggles when the specified boundary conditions are not aligned with the grid axes. The Chebychev basis is also commonly used, but requires the geometry to display various degrees of symmetry[2].

This work presents the novel ICE-CREAM (Integrated Codes to Efficiently Control Real-time Excitations in Architectured Materials) algorithm for performing quantum optimal control in spatially-complex systems. ICE-CREAM makes use of the finite-element (FE) basis, where the solution to a partial differential equation (PDE) is discretized over a series of triangular elements. The full set of triangular elements form a mesh, where solutions to the PDE are well-approximated by evaluating a set of simpler equations at the corners of each element (known as nodes). The finite element method is capable of representing highly-complex geometries as long as the mesh elements are sufficiently small/dense. As finite elements are commonly used in science and engineering, there are many pre-existing

tools/software packages for finite-element analysis. ICE-CREAM specifically makes use of MATLAB's Partial Differential Equation Toolbox^{TM}, which operates natively in the MATLAB environment. This additionally allows for compatibility with other projects created using MATLAB's PDE Toolbox ^{TM} such as the HADOKEN software package[3], which computes the static electronic wavefunctions for GaN/AlGaN core-shell nanowires. The use of finite elements for simulation of confined electron states in various types of 2D nanostructures has been demonstrated extensively in previous works[4][5][6]. As such, ICE-CREAM made use of the conduction band structures and electronic states generated by HADOKEN to demonstrate quantum control in complex nanostructures.

The algorithmic structure of ICE-CREAM follows a similar structure to the NIC-CAGE software package[1], which uses Crank-Nicholson propagation and gradient-based optimization to solve for optimal control fields for 1D photochemical systems. However, there are many differences in how various parameters are computed due to the NIC-CAGE algorithm's use of the finite-difference basis in only one dimension. Considerable optimization of the ICE-CREAM algorithm was required, as unlike in previous work, symmetry and finite-difference approximations could not be used to lower computational costs.

Chapter 2

Theory

2.1 Time-Evolution

The purpose of our propagation scheme is to evaluate how the starting state of a system will evolve under the influence of a time-dependent control-field ε. For a quantum system, the time-evolution of any given state is fully described by the time-dependent Schrödinger equation. In Hartree-atomic units the time-dependent Schrödinger is as follows:

$$i\frac{\partial}{\partial t}\psi(r,t) = \mathcal{H}(r,t)\,\psi(r,t) \tag{2.1}$$

Where $\mathcal{H}(r,t)$ is the Hamiltonian, given by the sum of the kinetic and potential energies within the system. For a single-particle, the Hamiltonian takes the form:

$$\mathcal{H}(r,t) = -\frac{1}{2m}\nabla^2 + V(r,t) \tag{2.2}$$

Where m is the mass of the particle, and $V(r,t)$ is the potential energy of the particle for a given point in space and time. Given that the Hamiltonian is time-invariant over an interval τ, solutions to the time-dependent Schrödinger equation take the following form:

$$\psi(t+\tau) = \mathcal{U}(t+\tau,t)\,\psi(t) \tag{2.3}$$

where:

$$\mathcal{U}(t+\tau,t) = e^{-i\tau\mathcal{H}(t+\frac{\tau}{2})} \tag{2.4}$$

The matrix $\mathcal{U}$, known as the time-evolution operator, describes how the original state will change over time. By splitting and Taylor-expanding this operator one obtains:

$$e^{-i\tau\mathcal{H}\left(t+\frac{\tau}{2}\right)} = \frac{e^{-\frac{i\tau}{2}\mathcal{H}\left(t+\frac{\tau}{2}\right)}}{e^{\frac{i\tau}{2}\mathcal{H}\left(t+\frac{\tau}{2}\right)}} \approx \frac{\left[\mathbb{I}-\frac{i\tau}{2}\mathcal{H}\left(t+\frac{\tau}{2}\right)\right]}{\left[\mathbb{I}+\frac{i\tau}{2}\mathcal{H}\left(t+\frac{\tau}{2}\right)\right]} \tag{2.5}$$

Otherwise known as the second-order Crank-Nicholson propagator. The Crank-Nicholson propagator is commonly used as it avoids the explicit computation of a matrix-exponential, conserves the norm of the wavefunction it operates on, and is numerically stable[7]. By defining the two matrices:

$$\begin{aligned}\mathbf{U} &= \left[\mathbb{I}+\tfrac{i\tau}{2}\mathcal{H}\left(t+\tfrac{\tau}{2}\right)\right]\\ \mathbf{W} &= \left[\mathbb{I}-\tfrac{i\tau}{2}\mathcal{H}\left(t+\tfrac{\tau}{2}\right)\right]\end{aligned} \tag{2.6}$$

Time-evolution in the Crank-Nicholson scheme takes the form:

$$\psi(t+\tau) = \mathbf{U}^{-1}\mathbf{W}\,\psi(t) = (2\mathbf{U}^{-1}-\mathbb{I})\,\psi(t) \tag{2.7}$$

To evolve our starting state over the time interval $[0, T]$, we split it into N discrete steps of length τ. Given that our Hamiltonian takes the following form:

$$\mathcal{H}(r, t + \tfrac{\tau}{2}) = -\frac{1}{2m}\nabla^2 + V(r) - \varepsilon(t + \tfrac{\tau}{2})\mu(r) \tag{2.8}$$

In our case, we write the Hamiltonian in the form:

$$\mathcal{H}(r, t) = H_0(r) - \varepsilon(t) \cdot \mu(r) \tag{2.9}$$

Where it is quite apparent that the time-dependency of the Hamiltonian stems solely from that of the control-field ε. As such, as long as the value of the field remains fairly constant over each timestep τ, time-evolution is well approximated by repeated application of the Crank-Nicholson propagator using a time-discretized control field.

$$\psi(r, N\tau) = \prod_{j=1}^{N} \left(2\left[\mathbb{I} + \tfrac{i\tau}{2}\mathcal{H}(j\tau + \tfrac{\tau}{2})\right]^{-1} - \mathbb{I}\right) \psi(r, 0) \tag{2.10}$$

2.2 Gradients

The goal of the quantum optimization algorithm is to find values of the time-varying control field that maximize the probability that a given starting state ψ_0 will be found in ψ_f after some time $T = N\tau$. The transition probability can be written as:

$$P = |\langle\psi_f|\psi_N\rangle|^2 = \langle\psi_f|\psi_N\rangle^*\langle\psi_f|\psi_N\rangle \tag{2.11}$$

where ψ_N denotes the state of the system after N timesteps:

$$|\psi_N\rangle = (\mathcal{U}_N \mathcal{U}_{N\text{-}1} \ldots \mathcal{U}_{j+1} \mathcal{U}_j \mathcal{U}_{j\text{-}1} \ldots \mathcal{U}_2 \mathcal{U}_1) |\psi_0\rangle \tag{2.12}$$

To find a control field that maximizes the transition probability, we will define an objective function J that depends on both the transition probability P, and a fluence term F.

$$J[\varepsilon] = P[\varepsilon] + F[\varepsilon] \tag{2.13}$$

This fluence term serves to prevent the magnitude of the control field from becoming unphysically large, making the control fields from ICE-CREAM applicable to experimental settings where achievable field strengths are often quite limited. The fluence penalty term is defined as:

$$F = -\int_0^T \alpha(t)\, \varepsilon(t)^2\, dt = -\tau \sum_{j=1}^{N} \alpha_j\, (\varepsilon_j)^2 \tag{2.14}$$

where α is a fluence penalty factor, determining how much field-damping should be applied to your system. While α can simply be set to a constant value over the simulation interval, by setting alpha in proportion to some time-dependent shape function, it should be possible to restrict the optimized control fields to follow some desired pulse profile.

In our control scheme, subsequent guesses for an optimal-control field are generated via the gradient-descent method as shown:

$$\varepsilon_j^{new} = \varepsilon_j + \gamma \frac{\partial J}{\partial \varepsilon_j} \tag{2.15}$$

In this equation, ε_j denotes the value of the control field at time $t = j\tau$, and γ is the scalar factor representing the learning rate step-size of the gradient-descent algorithm. The value of the step-size γ is chosen such that the objective computed using the updated control field: $J[\varepsilon_j^{new}]$ is maximized. The determination of this value γ is done using a golden-section line search, which is capable of finding a local maximum of the objective.

We will now look into obtaining a functional form of the $\frac{\partial J}{\partial \varepsilon_j}$ term used to generate subsequent guesses for an optimal control field. Taking the derivative of the objective with respect to the current values of the control-field yields:

$$\frac{\partial J}{\partial \varepsilon_j} = \frac{\partial P}{\partial \varepsilon_j} + \frac{\partial F}{\partial \varepsilon_j} \tag{2.16}$$

Where the gradient of the fluence term can easily be shown to be:

$$\frac{\partial F}{\partial \varepsilon_j} = -\frac{\partial}{\partial \varepsilon_j}\left(\tau \sum_{j=1}^{N} \alpha_j \left(\varepsilon_j\right)^2\right) = -\tau \sum_{j=1}^{N} 2\alpha_j \, \varepsilon_j \tag{2.17}$$

To compute the gradient of the transition probability $\frac{\partial P}{\partial \varepsilon_j}$, we will re-write P such that:

$$P = g \cdot g^* \quad \text{where: } g = \langle \psi_f | \psi_N \rangle = \int \psi_f^*(r) \cdot \psi_N(r) \, d^n r \tag{2.18}$$

This allows us to expand the partial derivative of P with respect to ε_j in terms of g:

$$\frac{\partial P}{\partial \varepsilon_j} = \left(\frac{\partial P}{\partial g} \cdot \frac{\partial g}{\partial \varepsilon_j}\right) + \left(\frac{\partial P}{\partial g^*} \cdot \frac{\partial g^*}{\partial \varepsilon_j}\right) \tag{2.19}$$

$$\frac{\partial P}{\partial g} = g^* \quad \frac{\partial P}{\partial g^*} = g \tag{2.20}$$

$$\frac{\partial P}{\partial \varepsilon_j} = \left(g^* \cdot \frac{\partial g}{\partial \varepsilon_j} \right) + \left(g \cdot \frac{\partial g^*}{\partial \varepsilon_j} \right) \tag{2.21}$$

From here it can be seen that:

$$\frac{\partial P}{\partial \varepsilon_j} = 2\,\mathrm{Real}\left(g^* \frac{\partial g}{\partial \varepsilon_j} \right) \tag{2.22}$$

$$\frac{\partial g}{\partial \varepsilon_j} = \langle \psi_f | \cdot \frac{\partial}{\partial \varepsilon_j} | \psi_N \rangle \tag{2.23}$$

recalling equation 2.12, one can expand the $|\psi_N\rangle$ term as such:

$$\frac{\partial}{\partial \varepsilon_j} |\psi_N\rangle = \frac{\partial}{\partial \varepsilon_j} \left(\mathcal{U}_N \, \mathcal{U}_{N\text{-}1} \ldots \mathcal{U}_{j+1} \, \mathcal{U}_j \, \mathcal{U}_{j\text{-}1} \ldots \mathcal{U}_2 \, \mathcal{U}_1 \right) |\psi_0\rangle \tag{2.24}$$

It is noted that $\mathcal{U}_j$ is the only term that depends on ε_j. As such, the partial derivative with respect to ε_j will only act on the $\mathcal{U}_j$ term within this product. It can be seen that:

$$\begin{aligned} \frac{\partial}{\partial \varepsilon_j} |\psi_N\rangle &= \left(\prod_{k=j+1}^{N} \mathcal{U}_k \right) \left(\frac{\partial}{\partial \varepsilon_j} \mathcal{U}_j \right) \left(\prod_{k=1}^{j-1} \mathcal{U}_k \right) |\psi_0\rangle \\ &= \left(\prod_{k=j+1}^{N} \mathcal{U}_k \right) \left(\frac{\partial}{\partial \varepsilon_j} \mathcal{U}_j \right) |\psi_{j-1}\rangle \end{aligned} \tag{2.25}$$

Taking care to note the definition of our matrix product at the limits of the indexing:

$$\prod_{k=x}^{y} A_k \equiv \mathbb{I} \qquad \text{for: } x > y \tag{2.26}$$

As such, it is now necessary to evaluate $\frac{\partial \mathcal{U}_j}{\partial \varepsilon_j}$. By recalling the definition of our Crank-

Nicholson propagator and differentiating by parts we get:

$$\frac{\partial \mathcal{U}_j}{\partial \varepsilon_j} = \frac{\partial \left(\mathbf{U}_j^{-1}\mathbf{W}_j\right)}{\partial \varepsilon_j} = \left(\frac{\partial \mathbf{U}_j^{-1}}{\partial \varepsilon_j}\right)\mathbf{W}_j + \mathbf{U}_j^{-1}\left(\frac{\partial \mathbf{W}_j}{\partial \varepsilon_j}\right) \tag{2.27}$$

Given that the derivative of an inverse matrix is known to be:

$$\frac{\partial A^{-1}}{\partial x} = -A^{-1}\left(\frac{\partial A}{\partial x}\right)A^{-1} \tag{2.28}$$

$$\frac{\partial \left(\mathbf{U}_j^{-1}\mathbf{W}_j\right)}{\partial \varepsilon_j} = -\mathbf{U}_j^{-1}\left(\frac{\partial \mathbf{U}_j}{\partial \varepsilon_j}\right)\mathbf{U}_j^{-1}\mathbf{W}_j + \mathbf{U}_j^{-1}\left(\frac{\partial \mathbf{W}_j}{\partial \varepsilon_j}\right) \tag{2.29}$$

recalling the definition of the $\mathbf{U}$ and $\mathbf{W}$ matrices:

$$\begin{aligned}\frac{\partial \mathbf{U}_j}{\partial \varepsilon_j} =& \frac{\partial}{\partial \varepsilon_j}\left(\mathbb{I} + \tfrac{i\tau}{2}\mathcal{H}_j\right) = \left(\frac{i\tau}{2}\right)\frac{\partial \mathcal{H}_j}{\partial \varepsilon_j} \\ \frac{\partial \mathbf{W}_j}{\partial \varepsilon_j} =& \frac{\partial}{\partial \varepsilon_j}\left(\mathbb{I} - \tfrac{i\tau}{2}\mathcal{H}_j\right) = -\left(\frac{i\tau}{2}\right)\frac{\partial \mathcal{H}_j}{\partial \varepsilon_j}\end{aligned} \tag{2.30}$$

where:

$$\frac{\partial \mathcal{H}_j}{\partial \varepsilon_j} = \frac{\partial}{\partial \varepsilon_j}(H_0 - \varepsilon_j \cdot \mu) = -\mu \tag{2.31}$$

Putting all of these equations together results in the formula:

$$\begin{aligned}\frac{\partial \mathcal{U}_j}{\partial \varepsilon_j} &= \mathbf{U}_j^{-1}\left(\frac{i\tau}{2}\mu\right)\mathbf{U}_j^{-1}\mathbf{W}_j + \mathbf{U}_j^{-1}\left(\frac{i\tau}{2}\mu\right) \\ &= \frac{i\tau}{2}\mathbf{U}_j^{-1}\mu\left(\mathcal{U}_j + \mathbb{I}\right)\end{aligned} \tag{2.32}$$

finally, it is possible to write a formulation of the gradient of $|\psi_N\rangle$ which can be simplified

to the form:

$$\begin{aligned}\frac{\partial}{\partial \varepsilon_j}|\psi_N\rangle &= \left(\prod_{k=j+1}^{N} \mathcal{U}_k\right)\left(\frac{i\tau}{2}\mathbf{U}_j^{\text{-1}}\mu\left(\mathcal{U}_j + \mathbb{I}\right)\right)|\psi_{j-1}\rangle \\ &= i\tau\left(\prod_{k=j+1}^{N} \mathcal{U}_k\right)\mathbf{U}_j^{\text{-1}}\mu\left(\frac{|\psi_j\rangle + |\psi_{j-1}\rangle}{2}\right)\end{aligned} \tag{2.33}$$

By plugging in the evaluation as seen above to equation 2.22, it is now possible to express the gradient of the transition probability in relatively simple terms:

$$\frac{\partial P}{\partial \varepsilon_j} = 2\,\text{Real}\left[\langle\psi_f|\psi_N\rangle^* \cdot i\tau\langle\psi_f|\left(\prod_{k=j+1}^{N} \mathcal{U}_k\right)\mathbf{U}_j^{\text{-1}}\mu\left(\frac{|\psi_j\rangle + |\psi_{j-1}\rangle}{2}\right)\right] \tag{2.34}$$

2.3 Finite-Elements

For a given system of differential equations, finding analytical solutions becomes increasingly difficult as the geometry in which the system is being solved grows more complex. By splitting the geometry of the problem into a series of subdomains (or elements), the solution to the differential equation can be well-approximated by the solution to a series of simpler equations computed within each subdomain. In the case of 2D geometry, space is divided into an ensemble of triangles forming a mesh as seen in Figure 2.1.

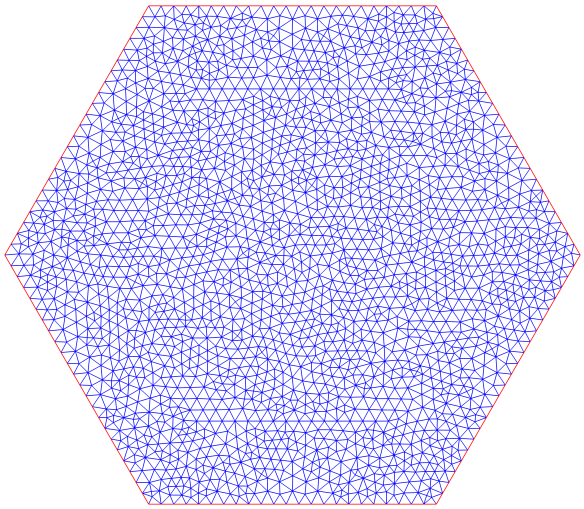

Figure 2.1: An example mesh used for the hexagonal core-shell nanowire. This mesh contains approximately 2200 mesh points and has side lengths of 50nm. The core-shell interface should be visible due to the constraint of several nodes to lie along the boundary.

ICE-CREAM was built in MATLAB, which allows it to take advantage of MATLAB's PDE Toolbox, a highly-developed library for working with finite elements. This also allows for compatibility with HADOKEN, an algorithm built for computing the band-structure and electron wave functions in core-shell GaN/AlGaN nanowires. The electronic eigenstates found with HADOKEN are used as both starting and target states for ICE-CREAM. MATLAB's PDE Toolbox is capable of solving partial differential equations in the form of either:

$$\begin{aligned} -\nabla \cdot (c\nabla \mathbf{u}) + a\mathbf{u} &= f \quad \text{(elliptic)} \\ -\nabla \cdot (c\nabla \mathbf{u}) + a\mathbf{u} &= \lambda d\mathbf{u} \quad \text{(eigenvalue)} \end{aligned} \tag{2.35}$$

Where $c, a, f,$ and, d are coefficients specified by the user. Boundary conditions are specified

along some path in your geometry using either:

$$\begin{aligned} \overrightarrow{n} \cdot (c\nabla \mathbf{u}) + q\mathbf{u} = g \quad \text{(Neumann)} \\ h\mathbf{u} = r \quad \text{(dirichlet)} \end{aligned} \tag{2.36}$$

To begin finding solutions to your system of differential equations, MATLAB's PDE Toolbox™ requires the assembly of several finite-element matrices. Linear methods can then be used to solve for desired quantities in your system. These matrices are assembled using the PDE coefficients, integrated over the test functions ϕ_i , ϕ_j.

$$\begin{aligned} K_{i,j} &= \int_{\Omega} (c\nabla\phi_j) \cdot \nabla\phi_i \\ M_{i,j} &= \int_{\Omega} a\,\phi_j\,\phi_i \\ Q_{i,j} &= \int_{\partial\Omega} q\,\phi_j\,\phi_i \\ F_i &= \int_{\Omega} f\,\phi_i \\ G_i &= \int_{\partial\Omega} g\,\phi_i \end{aligned} \tag{2.37}$$

Where the PDE now takes the form: $(K + M + Q)\,U = (F + G)$. In our case, we are solely interested in constructing the Hamiltonian for use in the propagation matrix. Using the following expressions from MATLAB's PDE Toolbox documentation[8]:

$$\begin{aligned} K_c u_i = \lambda \left(B'\,D\,B\right) u_i \\ \mathbf{u} = B u_i \end{aligned} \tag{2.38}$$

Here D is computed similarly as the mass matrix $M_{i,j}$ with the exception that the integral

is evaluated over the coefficient d rather than a. For our purposes, d is defined to be one. Another key term is B, known as the Dirichlet null-space matrix. This matrix puts the PDE system into a reduced basis by eliminating the elements where the value of the nodes is fixed (i.e. at specified edges within the geometry). The matrix K_c (known as the stiffness-matrix) is defined as $K_c = (K + M + Q)$. By expanding and re-structuring the previous definition we get:

$$\left[B^{-1}\left(B'\,D\,B\right)^{-1}K_cB\right]\mathbf{u} = \lambda\mathbf{u} \tag{2.39}$$

This expression has an equivalent form to the time-independent Schrödinger equation $\mathcal{H}|\psi\rangle = \lambda|\psi\rangle$. This means it is possible to define our Hamiltonian in terms of finite-element matrices as such:

$$\mathcal{H} = \left[B^{-1}\left(B'\,D\,B\right)^{-1}(K + M + Q)\,B\right] \tag{2.40}$$

For our purposes, we want to split the Hamiltonian into a time-independent and time-dependent portion. Recalling that our problem takes the form:

$$-\frac{1}{2m}\nabla^2\,\psi_n + \Big(V_{CB}(r) - \varepsilon(t)\,\boldsymbol{\mu}(r)\Big)\,\psi_n = E_n\psi_n \tag{2.41}$$

Where the expression $V_{CB}(r) - \varepsilon(t)\,\boldsymbol{\mu}(r)$ is equivalent to the coefficient a. As such, the mass matrix as defined in equation 2.37 can be written as:

$$\begin{aligned} M_{i,j} &= \int_\Omega \Big(V_{CB}(r) - \varepsilon(t)\,\boldsymbol{\mu}(r)\Big)\,\phi_j\,\phi_i\,dx \\ M_{i,j} &= \int_\Omega V_{CB}(r)\,\phi_j\,\phi_i \;-\; \varepsilon(t)\int_\Omega \boldsymbol{\mu}(r)\,\phi_j\,\phi_i \\ M_{i,j} &= M_{CB} - \varepsilon(t)M_\mu \end{aligned} \tag{2.42}$$

This can now be plugged into our expression for the Hamiltonian:

$$\begin{aligned}
\mathcal{H} &= B^{-1}\left(B'\,D\,B\right)^{-1}\left(K + M_{CB} - \varepsilon(t)M_{\mu} + Q\right)B \\
\mathcal{H} &= \left[B^{-1}\left(B'\,D\,B\right)^{-1}\left(K + M_{CB} + Q\right)B\right] - \varepsilon(t)\left[B^{-1}\left(B'\,D\,B\right)^{-1}M_{\mu}\,B\right] \\
\mathcal{H} &= H_0 - \varepsilon(t)\boldsymbol{\mu}
\end{aligned} \tag{2.43}$$

By extracting the time-dependent portion from our Hamiltonian, the value of the finite-element Hamiltonian is easily computable for any arbitrary value of $\varepsilon(t)$ given that the matrices H_0 and $\boldsymbol{\mu}$ have been computed previously.

It is also important to note how integrals are structured in finite elements. Given a function f whose value is defined at every node within the finite-element mesh, the integral is defined as a weighted sum over the value of f approximated at the midpoint of each element within the mesh. For 2-D triangular elements, this integral can be written as:

$$\iint f(x,y)\,dx\,dy \;\approx\; \sum_i \Delta area_i \left[\sum_{n=1,2,3} \frac{f(x_n,y_n)}{3}\right]_i \tag{2.44}$$

Where for a given triangular mesh element i, (x_n, y_n) represents the x-y coordinates of the nth node of the element, and $\Delta area$ represents the volume taken up by the specified element. In our case, it is especially important to examine integrals taking the following form:

$$\langle\psi_a|\psi_b\rangle = \iint \psi_a^* \cdot \psi_b\,dx\,dy \;\approx\; \sum_i \Delta area_i \left[\sum_{n=1,2,3} \frac{\psi_a^*(x_n,y_n)\odot\psi_b(x_n,y_n)}{3}\right]_i \tag{2.45}$$

Here it is key to note that as opposed to conventional quantum mechanics, where the com-

mutation between states can be expressed via the dot-product, here commutation between states requires a summation over the element-wise product of the two states.

2.4 Transition Dipole-Moment

It is important to note, that for a given quantum system, not all transitions between states are equally probable, let alone possible. For systems whose evolution depends solely on electric-dipole interactions, the transition dipole-moment(TDM):

$$\text{transition dipole-moment} = \langle \psi_f | \mu_x \, \hat{\mathbf{x}} + \mu_y \, \hat{\mathbf{y}} | \psi_0 \rangle \tag{2.46}$$

is a complex-valued vector that quantifies how strongly an electric field (polarized in either the x or y direction) couples to a given quantum-state transition. For a given combination of initial/final states of your system, if the magnitude of the transition dipole-moment is zero, said transition is impossible to achieve via interaction with an electric field. If a combination of states results in large values of the transition-dipole moment, said transition is likely to be easily facilitated via interaction with an electric field. As such, one can pre-determine whether a given quantum state transition is achievable via the application of an external electric control field.

Chapter 3

Algorithm

3.1 Overview

This section explains and describes the overall structure of the ICE-CREAM algorithm, as well as examines the methods of pre-conditioning the inputs to ensure convergence of the algorithm. The first main method of pre-conditioning used in ICE-CREAM is scaling the timestep duration proportionally to the quantum speed limit of the given transition. This results in timestep values that correctly capture the dynamics of the system. The second main method of preconditioning is to rotate the components of the electric field such that the new basis is strongly coupled to the system along one axis and weakly coupled along the other. These methods are further discussed later in the section.

The structure of ICE-CREAM closely follows that of the NIC-CAGE algorithm[1], with some important considerations in how specific parameters are calculated. The need to iteratively compute a large number of inverted matrices combined with the large non-

Algorithm 1: ICE-CREAM

Inputs : mesh data: $[p, e, t]$, PDE coefficients: $[a, c]$, starting-state: $\psi_{initial}$, target-state: ψ_{target}, timestep-scaling factor: τ_{scale}, number of timesteps: N, fluence penalty factor: α, maximum iterations: $iter_{max}$, maximum transition probability: P_{max}

Outputs: optimal-control fields: $[\varepsilon_x, \varepsilon_y]$, final propagated states: $[\psi_1 \rightarrow \psi_{final}]$, final transition-probability: P_{final}

Initialize:

Calculate H_0 and $\boldsymbol{\mu}$ using Eq.(2.43)

Rotate E-field basis using Eq.(3.5) and Eq.(3.6)

Set the timestep using $\tau_{scale} \cdot t_{min}$ from Eq.(3.9)

$P = 0,\ iter = 1$

while $P < P_{max}$ *&* $iter < iter_{max}$ **do**

  $|\psi_1\rangle = |\psi_{inital}\rangle$

  for $j = 1 \rightarrow N$ **do**

    $\mathbf{U}_j = \mathbb{I} + \frac{i\tau}{2}(H_0 - \varepsilon_j \cdot \boldsymbol{\mu})$

    $|\psi_{j+1}\rangle = \left(2\mathbf{U}_j^{-1} - \mathbb{I}\right) \cdot |\psi_j\rangle$

  end

  $P = |\langle\psi_{target}|\psi_{N+1}\rangle|^2$

  if $P \geq P_{max}$ **then break**

  $UW = \mathbb{I}$

  for $j = N \rightarrow 1$ **do**

    if $j < N$ **then**

      $UW = UW \cdot \left(2\mathbf{U}_j^{-1} - \mathbb{I}\right)$

    end

    $\frac{\partial}{\partial \varepsilon_j}|\psi_{N+1}\rangle = (\frac{i\tau}{2})\ UW \cdot \mathbf{U}_j^{-1} \cdot \boldsymbol{\mu} \cdot (|\psi_{j+1}\rangle + |\psi_j\rangle)$

    Calculate $\frac{\partial J}{\partial \varepsilon_j}$ using $\frac{\partial}{\partial \varepsilon_j}|\psi_{N+1}\rangle$

  end

  Calculate γ that maximizes $J[\varepsilon_j + \gamma \frac{\partial J}{\partial \varepsilon_j}]$ using a golden-section linesearch

  $\varepsilon_j = \varepsilon_j + \gamma \cdot \frac{\partial J}{\partial \varepsilon_j}$

end

diagonal/banded Hamiltonian in the finite-element basis results in the computational cost of ICE-CREAM being quite high. As such, all results shown in this work were computed using the HPCC cluster at UCR.

3.2 Rotation of the Field Basis

Recalling from the previous section, transition dipole-moment acts as a descriptor of the coupling between the x and y components of the electric field, for any given quantum-state transition. By examining the relative strength of the x and y components of the TDM, it is possible to find the direction in which the electric field has the strongest coupling to any specific transition. We can refer to the component of the electric field along this direction as being in the "strong-axis". Additionally, we find that any oscillations in the electric-field perpendicular to the strong-axis are very weakly coupled to the state transition. As such, this direction is referred to as the "weak-axis". The following equation can be used to determine the angle between the x-axis and the strong-axis for state transition from $|\psi_a\rangle$ to $|\psi_b\rangle$.

$$\begin{aligned} x &= \langle\psi_a|\boldsymbol{\mu}_x|\psi_b\rangle \\ y &= \langle\psi_a|\boldsymbol{\mu}_y|\psi_b\rangle \\ \theta &= 2 \cdot arctan\left(\frac{y}{x+\sqrt{x^2+y^2}}\right) \end{aligned} \tag{3.1}$$

By re-writing our field-dependent potential in the form:

$$V_\mu = \begin{bmatrix} \mu_x & \mu_y \end{bmatrix} \cdot \begin{bmatrix} \varepsilon_x \\ \\ \varepsilon_y \end{bmatrix} \tag{3.2}$$

We can insert the rotation matrix:

$$R[\theta] = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{3.3}$$

To get:

$$V_\mu = \begin{bmatrix} \mu_x & \mu y \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \\ \varepsilon_y \end{bmatrix} \tag{3.4}$$

Which can be written in terms of new coefficients:

$$\begin{aligned} \mu_u &= \mu_x \cos\theta + \mu_y \sin\theta \\ \mu_v &= \mu_y \cos\theta - \mu_x \sin\theta \end{aligned} \tag{3.5}$$

and:

$$\begin{aligned} \varepsilon_u &= \varepsilon_x \cos\theta + \varepsilon_y \sin\theta \\ \varepsilon_v &= \varepsilon_y \cos\theta - \varepsilon_x \sin\theta \end{aligned} \tag{3.6}$$

By restructuring the fields in this manner, oscillations in the electric field along the u-direction are strongly coupled to our desired transition while oscillations in the v-direction are not strongly correlated to our desired quantum-state transition. In theory, this could be

used to effectively ignore the "weak-axis" contributions and only examine control fields with a singular polarization. However, this mostly serves as a pre-conditioner for the gradient descent algorithm, as the gradient will now primarily depend on contributions from the "strong-axis" polarization.

3.3 Time-step Scaling

The ability of the ICE-CREAM algorithm to converge to an effective control field is quite sensitive to changes in the timestep duration τ. As such, it was necessary to find means of placing suitable limits on the values of the timestep parameter. Previous work has shown that there is a lower bound to the amount of time over which a state-transition can ocurr in a quantum system, known as the quantum speed limit[9][10][11]. The quantum speed limit can be calculated for two states using the Mandelstam-Tamm relation:

$$\Delta t \geq \frac{\arccos |\langle \psi_f | \psi_0 \rangle|}{\Delta E} \tag{3.7}$$

In the absence of any external fields, the energy-uncertainty ΔE is equivalent to ΔH_0 where the dispersion relation Δ of a given operator is calculated via:

$$\Delta A = \sqrt{\langle A^2 \rangle - \langle A \rangle^2} \tag{3.8}$$

As such we get the minimum time needed for the state $|\psi_0\rangle$ to spontaneously transition to the orthogonal state $|\psi_f\rangle$.

$$t_{min} = \frac{\pi/2}{\sqrt{\langle H^2\rangle - \langle H\rangle^2}} \tag{3.9}$$

As we wish for the evolution of our system to depend primarily on our control fields, the timestep value used in our algorithm should be much smaller than this minimum time. From here we can set our timestep value using: $\tau = \gamma * t_{min}$, where γ is some user-set scaling factor that should be significantly smaller than one. Meanwhile, the energy uncertainty over the entire transition can be estimated from the energy-eigenvalues of the initial and final states i.e. $\Delta E \approx \frac{(E_f - E_0)}{2}$. From this we can find a minimum requirement for the total transition time T, where:

$$T_{min} = \frac{\pi}{|E_f - E_0|} \tag{3.10}$$

By choosing timestep parameters that meet both of these criteria, the convergence of the algorithm is ensured.

Chapter 4

Results

To demonstrate the efficacy of the ICE-CREAM algorithm in generating optimal control fields, quantum control was performed for various combinations of initial and target state using GaN/AlGaN core-shell nanowires modeled by the HADOKEN algorithm. These nanowires can have either hexagonal or triangular cross-sections. To visualize the quantum electronic states for these nanowires, the single-electron wavefunction is shown for the ground state and the first five excited states.

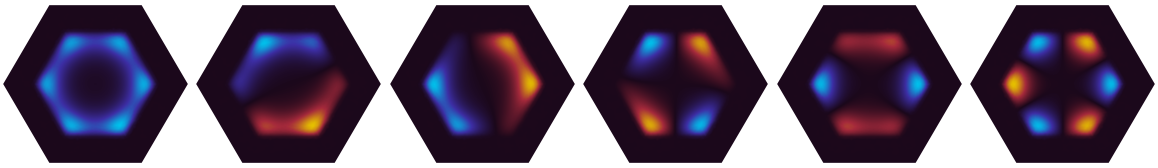

Figure 4.1: Electron wavefunctions in the hexagonal core-shell nanowire with increasing energy moving from left to right.

In the case of a triangular GaN/AlGaAs core-shell nanowire, depending on the configuration of the crystal structure, the bottom core-shell interface will have either an upwards or downwards polarization. This spontaneous polarization of the interface results

in two possible configurations of the band structure. In the case of the N-face configuration, the minimum of the conduction band is located at the top corner of the core region. In the case of a Ga-face configuration, the minimum of the conduction band is located along the bottom edge of the core region. The result is the localization of the electronic wavefunction around the minima of the conduction band as seen in Fig 4.2 and Fig 4.3.

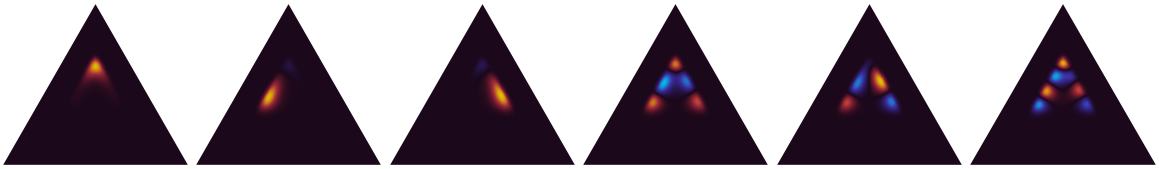

Figure 4.2: Electron wave-functions in the N-face core-shell nanowire with increasing energy moving from left to right.

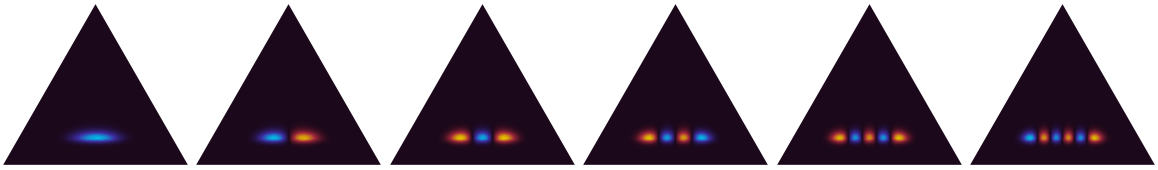

Figure 4.3: Electron wave-functions in the Gallium-face core-shell nanowire with increasing energy moving from left to right.

By computing the magnitude of the TDM for multiple combinations of state transitions, one obtains a map of the allowed transitions for each nanowire configuration as seen in Fig 4.4.

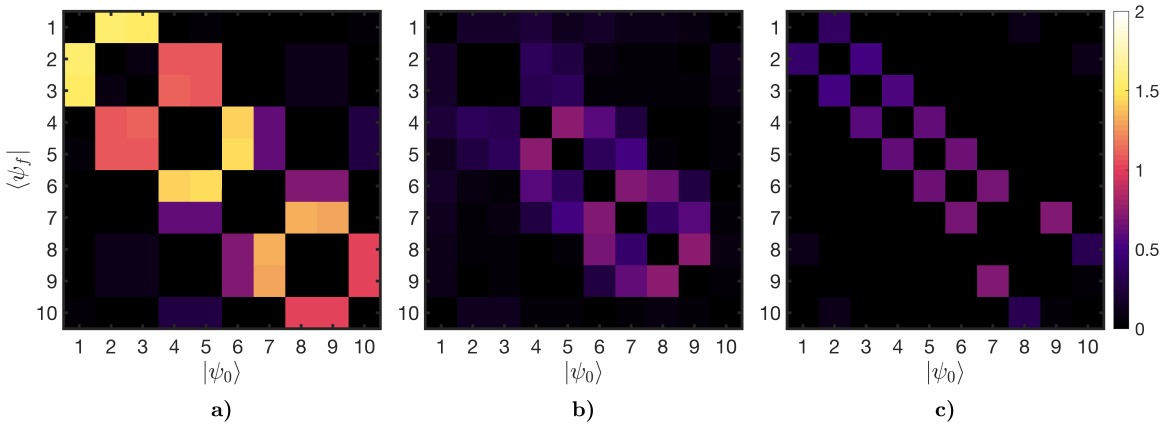

Figure 4.4: Magnitudes of the transition dipole-moment couplings in **(a)** the hexagonal mesh **(b)** the N-face triangular mesh **(c)** the Ga-face triangular mesh

As seen in Fig 4.4, the TDM magnitude is generally much smaller in the triangular geometries compared to the hexagonal geometries. As such, when attempting to run the ICE-CREAM algorithm using the triangular configurations, the optimized transition probability would converge towards values significantly lower than our cutoff threshold ($P_{max} = 0.95$). This result is backed by the physical interpretation of the TDM, where zero (or extremely small) values of the TDM mean that a given transition is impossible via electric-dipole interactions. As such, our current study was restricted to working within the hexagonal mesh, using transitions with high TDM magnitudes. As such, the state-transitions of $1 \rightarrow 2$, $1 \rightarrow 3$, $4 \rightarrow 6$, and $5 \rightarrow 6$ were chosen. The resulting control fields of the ICE-CREAM algorithm are shown below:

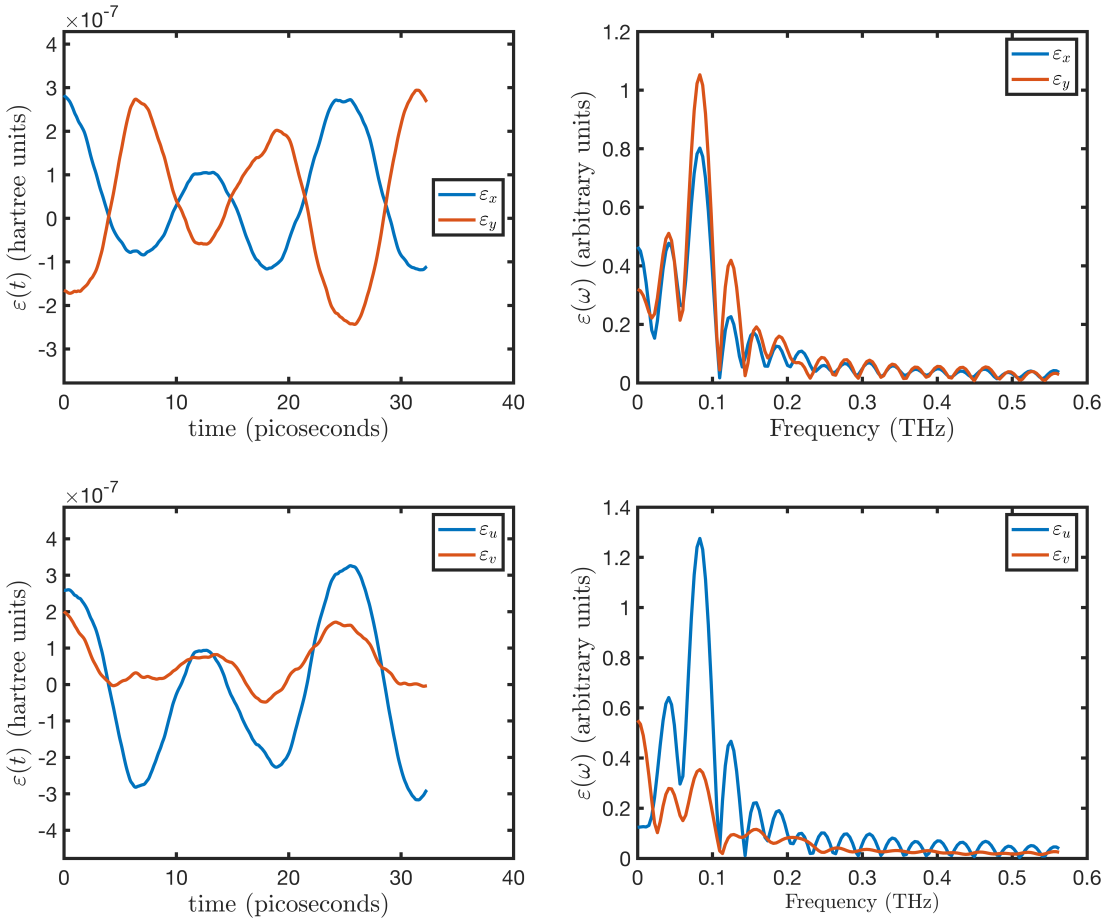

Figure 4.5: Optimized control fields for the $1 \rightarrow 2$ state transition in both the x-y and u-v axes. The angle between the x-y and u-v axes is $\theta = 5.093$ radians.

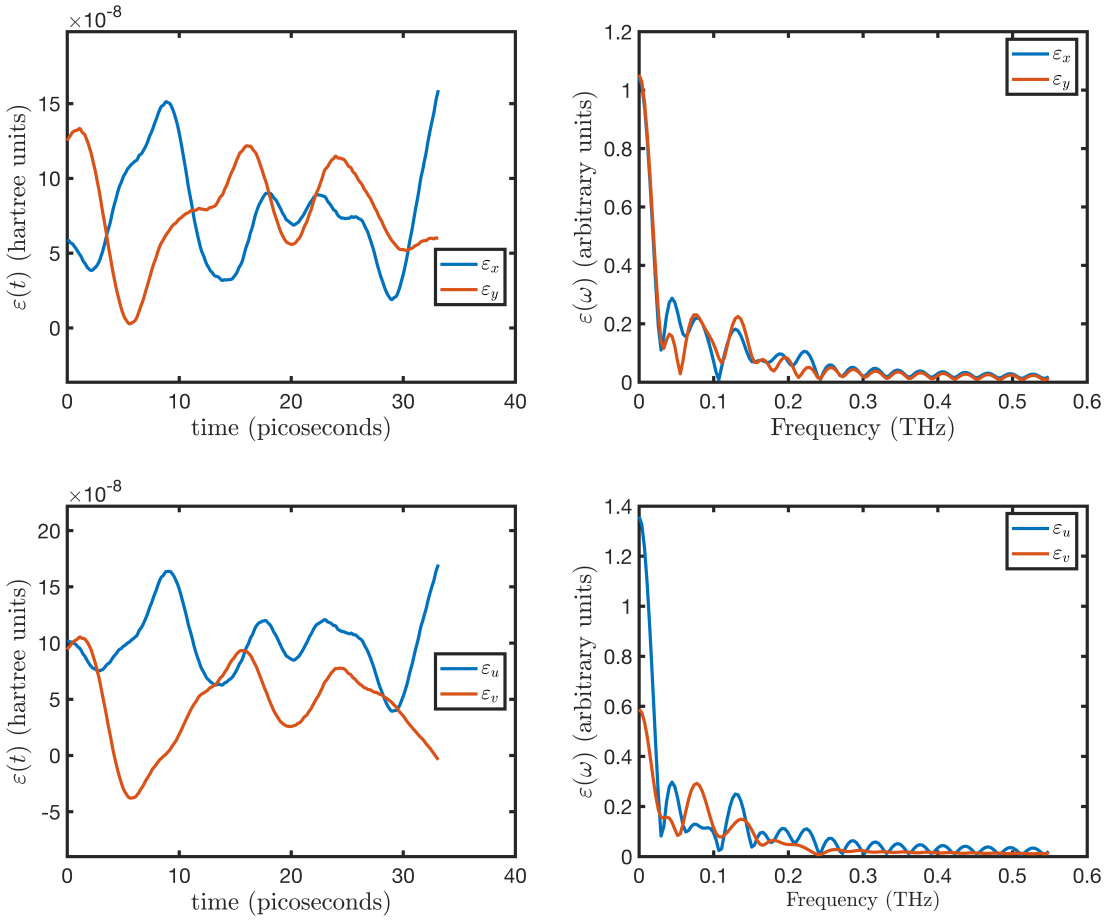

Figure 4.6: Optimized control fields for the $1 \rightarrow 3$ state transition in both the x-y and u-v axes. The angle between the x-y and u-v axes is $\theta = 0.382$ radians.

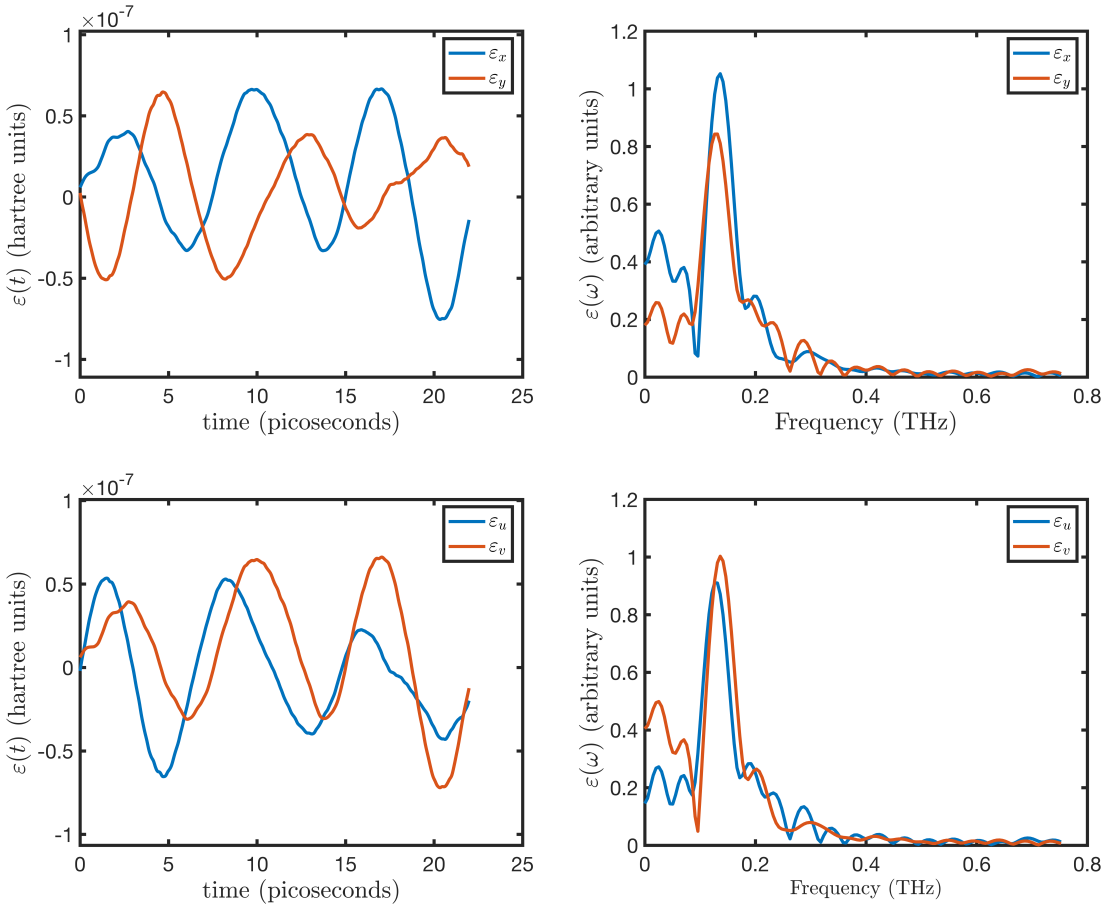

Figure 4.7: Optimized control fields for the $4 \rightarrow 6$ state transition in both the x-y and u-v axes. The angle between the x-y and u-v axes is $\theta = 4.799$ radians

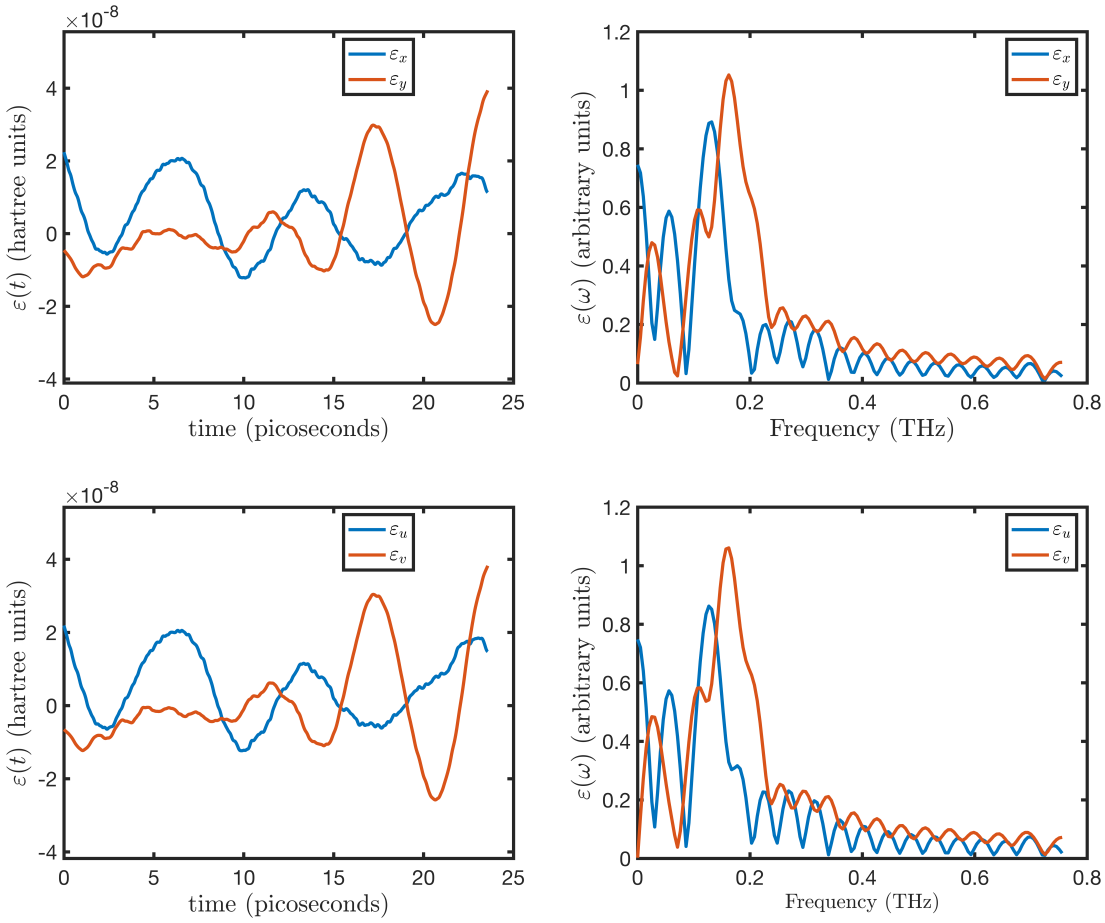

Figure 4.8: Optimized control fields for the $5 \rightarrow 6$ state transition in both the x-y and u-v axes. The angle between the x-y and u-v axes is $\theta = 0.089$ radians.

Chapter 5

Conclusion

In this work, the use of the finite-element basis for performing quantum optimal control of geometrically-complex systems has been successfully demonstrated. Control fields have been obtained for a variety of state transitions in simulated 2D-cross-sections of hexagonal GaN/AlGaN core-shell nanowires, with ultimate transition probabilities exceeding 95%. Future work could seek to improve the computational cost and reliability of convergence of the algorithm, as finding the optimal simulation parameters can be fairly time-consuming. Other possibilities for future research include the simulation of 3D structures, implementing exchange potentials for capturing multi-electron dynamics, and the inclusion of magnetic fields into the control scheme. Nevertheless, ICE-CREAM shows potential for use in research and industry to aid in the development of control methods in nanoscale architectured materials.

Bibliography

[1] A. Raza, C. Hong, X. Wang, A. Kumar, C. R. Shelton, and B. M. Wong, "Nic-cage: An open-source software package for predicting optimal control fields in photo-excited chemical systems," *Computer Physics Communications*, vol. 258, p. 107541, Jan 2021.

[2] A. W. Long and B. M. Wong, "Pamela: An open-source software package for calculating nonlocal exact exchange effects on electron gases in core-shell nanowires," *AIP Advances*, vol. 2, no. 3, p. 032173, 2012.

[3] C. Chevalier and B. M. Wong, "Hadoken: An open-source software package for predicting electron confinement effects in various nanowire geometries and configurations," *Computer Physics Communications*, vol. 274, p. 108299, 2022.

[4] B. M. Wong, F. Léonard, Q. Li, and G. T. Wang, "Nanoscale effects on heterojunction electron gases in gan/algan core/shell nanowires," *Nano Letters*, vol. 11, no. 8, p. 3074–3079, Aug 2011.

[5] M. Fickenscher, T. Shi, H. E. Jackson, L. M. Smith, J. M. Yarrison-Rice, C. Zheng, P. Miller, J. Etheridge, B. M. Wong, Q. Gao, S. Deshpande, H. H. Tan, and C. Jagadish, "Optical, structural, and numerical investigations of gaas/algaas core–multishell nanowire quantum well tubes," *Nano Letters*, vol. 13, no. 3, p. 1016–1022, Mar 2013.

[6] K. Pemasiri, H. E. Jackson, L. M. Smith, B. M. Wong, S. Paiman, Q. Gao, H. H. Tan, and C. Jagadish, "Quantum confinement of excitons in wurtzite inp nanowires," *Journal of Applied Physics*, vol. 117, no. 19, p. 194306, May 2015.

[7] H. Gharibnejad, B. Schneider, M. Leadingham, and H. Schmale, "A comparison of numerical approaches to the solution of the time-dependent schrödinger equation in one dimension," *Computer Physics Communications*, vol. 252, p. 106808, 2020.

[8] I. The MathWorks, *Partial Differential Equation Toolbox*, Natick, Massachusetts, United State, 2021. [Online]. Available: https://www.mathworks.com/help/pde/

[9] K. Bhattacharyya, "Quantum decay and the mandelstam-tamm-energy inequality," *Journal of Physics A: Mathematical and General*, vol. 16, no. 13, p. 2993–2996, 1983.

[10] S. Deffner and S. Campbell, "Quantum speed limits: From heisenberg's uncertainty principle to optimal quantum control," *Journal of Physics A: Mathematical and Theoretical*, vol. 50, no. 45, p. 453001, 2017.

[11] N. Hörnedal, D. Allan, and O. Sönnerborn, "Extensions of the mandelstam–tamm quantum speed limit to systems in mixed states," *New Journal of Physics*, vol. 24, no. 5, p. 055004, 2022.