

Time-Domain Simulation of the Quantum Magnetic Dipole Moment

A Thesis

Presented in Partial Fulfilment of the Requirements for the

Degree of Master of Science

with a

Major in Electrical Engineering

in the

College of Graduate Studies

University of Idaho

by

Jennifer Ellaine Houle

Major Professor: Dennis Sullivan, Ph.D.

Committee Members: Mark Kuzyk, Ph.D.; Mohsen Guizani, Ph.D.

Department Administrator: Mohsen Guizani, Ph.D.

May 2017

### Authorization to Submit Thesis

This thesis of Jennifer Ellaine Houle, submitted for the Degree of Master of Science with a Major in Electrical Engineering and titled "Time-Domain Simulation of the Quantum Magnetic Dipole Moment," has been reviewed in final form. Permission, as indicated by the signatures and dates below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor: \_\_\_\_\_ Date: \_\_\_\_\_  
Dennis Sullivan, Ph.D.

Committee Members: \_\_\_\_\_ Date: \_\_\_\_\_  
Mark Kuzyk, Ph.D.

\_\_\_\_\_ Date: \_\_\_\_\_  
Mohsen Guizani, Ph.D.

Department

Administrator: \_\_\_\_\_ Date: \_\_\_\_\_  
Mohsen Guizani, Ph.D.

## **Abstract**

In this thesis, the Finite-Difference Time Domain (FDTD) method is used to implement the Schrödinger equation in Python. This method is used to find the ground eigenstate and to simulate an electron within a three-dimensional torus. The magnetic dipole moment operator is developed, both with and without an applied magnetic field, and the equations describing a magnetic field applied to the torus are developed using the FDTD method. The magnetic dipole moment operator and implementation of a magnetic field are verified using a classical method. The magnetic dipole moment operator is used to calculate the magnetic susceptibility of a grating torus.

## **Acknowledgements**

I would like to thank Dr. Dennis Sullivan for his incredible help and support in this project. I did not imagine it would be possible to complete a thesis remotely, but Dr. Sullivan found a way to make it run smoothly. He answered countless questions I had on the simulation method, and gave me encouragement to pursue this topic. I am grateful for the opportunity to learn and research so much!

I would also like to thank Dr. Mark Kuzyk for helping me learn about nonlinear optics. He provided a constant source of information and sanity checks for what was going on in my project. Sean Mossman and Ethan Crowell were also wonderfully helpful in explaining so many quantum mechanical concepts to me, and answering my numerous questions.

## **Dedication**

I would not have been able to complete this thesis without the love and support of my husband, Bijan. I could never have found time to work on this without his help with countless other things in my life. As a bonus, he is also my resident Python expert! My lovely daughters, Adelaide and Alana, were also very patient in sharing my attention with this thesis.

Finally, I want to thank Dr. Herb Hess for all his support throughout my education both as an undergraduate and a graduate student. He has always made the University of Idaho like a second home to me.

## Table of Contents

Authorization to Submit Thesis.....	ii
Abstract.....	iii
Acknowledgements.....	iv
Dedication.....	v
List of Tables.....	viii
List of Figures.....	ix
Chapter 1: Introduction.....	1
Research Goals.....	2
Chapter 2: FDTD Method.....	3
Schrödinger Equation Implementation.....	3
FDTD Method to Find Eigenenergies and Eigenstates.....	7
Finding the Eigenenergy.....	7
Finding the Eigenfunction.....	8
Chapter 3: Simulation of a Torus.....	12
Simulation Setup.....	12
Perfectly Matched Layer (PML).....	14
Setting up the Potential.....	16
Determining the Ground State Eigenfunction of the Torus.....	19
Ground Eigenstate Verification.....	28
Chapter 4: Magnetic Dipole Moment Operator and the Simulation of a Magnetic Field.....	31
Calculating the Magnetic Dipole Moment.....	31
Implementing the Magnetic Dipole Moment Operator using the FDTD Method.....	32
Verifying the Accuracy of the Magnetic Dipole Moment Calculation.....	33

Adding a Magnetic Field .....	36
FDTD with a Magnetic Field Applied .....	37
Magnetic Dipole Moment Operator with a Magnetic Field .....	39
Verifying the Accuracy of the Magnetic Dipole Moment Calculation with a Static Magnetic Field .....	39
Chapter 5: Simulation of a Time-Harmonic Magnetic Field .....	42
Effects of a Time-Harmonic Field on a Particle in the Torus.....	42
Magnetic Dipole Moment without a Magnetic Field Applied .....	44
Magnetic Susceptibility.....	45
Grated Torus .....	46
Magnetic Susceptibility Results for a Grated Torus.....	48
Discussion .....	51
Chapter 6: Summary.....	53
Summary .....	53
Conclusions for Implementation in Python .....	54
References .....	55

**List of Tables**

Table 1. Eigenenergy compared with the number of iterations of the FDTD simulation.....	23
Table 2. Classical and quantum magnetic dipole moment with a static magnetic field.....	41



## List of Figures

Figure 1. Torus structure.....	12
Figure 2. Problem space (defined in cells and not drawn to scale) in the (a) x-y direction, and (b) x-z direction. ....	14
Figure 3. Illustration of subcells.....	17
Figure 4. Contour plots of the potential (a) without averaging over subcells through center of z-axis, and (b) with averaging over subcells through center of z-axis, (c) without averaging over subcells through center of y-axis, and (d) with averaging over subcells through center of y-axis.....	19
Figure 5. Test pulse initialized within a torus .....	20
Figure 6. $\Psi_{real}$ over time using 10,000 iterations of a test pulse.....	21
Figure 7. Eigenenergy distribution using 10,000 iterations of a test pulse.....	22
Figure 8. Test function initialized within a wire.....	24
Figure 9. $\Psi_{real}$ over time using 10,000 iterations of a test function.....	25
Figure 10. Eigenenergy distribution using 10,000 iterations of a test function .....	25
Figure 11. Ground eigenstate generated over 10,000 iterations (a) using a test pulse, and (b) using a test function.....	27
Figure 12. Ground eigenstate generated over 200,000 iterations using a test function .....	28
Figure 13. Ground eigenstate after (a) 0 iterations (initial state), (b) 106 iterations (quarter cycle), (c) 211 iterations (half cycle), (d) 317 iterations (three-quarters of a cycle), and (e) 422 iterations (full cycle).....	30
Figure 14. Progression of wave packet over time.....	34
Figure 15. The expectation value of the position in the x- and y-directions over time..	35
Figure 16. Direction of magnetic field applied to the torus.....	36
Figure 17. Progression of wave packet over time with a static 25 T magnetic field .....	40
Figure 18. The expectation value of the position in the x- and y-directions over time with a static 25 T magnetic field.....	41
Figure 19. The results of a simulation of a particle in a torus under the influence of a sinusoidal magnetic field with a maximum magnitude of 30 T and a	

frequency of 20 THz. (a) The magnetic field, (b) the magnetic dipole moment, (c) the Fourier transform of the magnetic field, and (d) the Fourier transform of the dipole moment. ....	43
Figure 20. The magnetic dipole moment versus the magnetic field strength for a magnetic field of 20 THz .....	44
Figure 21. Potential showing a grated torus; cross section shown at $z = Z_c$ .....	47
Figure 22. Ground eigenstate for the grated potential; cross section shown at $z = Z_c$ ...	47
Figure 23. The results of a simulation of a particle in a grated torus under the influence of a sinusoidal magnetic field with a maximum magnitude of 30 T and a frequency of 20 THz. (a) The magnetic field, (b) the magnetic dipole moment, (c) the Fourier transform of the magnetic field, and (d) the Fourier transform of the magnetic dipole moment.....	49
Figure 24. The magnetic dipole moment determined at $f_0$ versus magnetic field strength for a particle in a grated torus. $\alpha = 7.74 \times 10^{-3}$ and $\gamma = -6.33 \times 10^{-10}$ .....	50
Figure 25. The magnetic dipole moment determined at $3f_0$ versus magnetic field strength. $\gamma = 2.20 \times 10^{-10}$ .....	51

## Chapter 1: Introduction

Nonlinear optics (NLO) studies the nonlinear response of a material to the strength of an optical field, such as the strength of the electric field [1]. The nonlinear response may be quadratic, cubic, or higher order [2]. Nonlinear properties can lead to many different phenomena such as a change of the refractive index as in the Kerr effect, or in a change of frequency as in second harmonic generation.

Materials with nonlinear properties are used in many ways. Applications include optical switching [1], lasers, photovoltaic cells [3], imaging [3], and cancer therapy [4], [5]. To improve applications, it is important to establish good methods and models to explore the optimization of nonlinear effects by increasing the nonlinear response. This work provides a way of calculating the nonlinear response of nanostructures. This helps in designing new systems without the need for doing difficult and time-consuming measurements.

The Finite-Difference Time Domain (FDTD) method offers a method of directly implementing the Schrödinger equation in a three-dimensional structure. This method has been described for use in quantum mechanical simulation to find eigenfunctions of arbitrarily shaped structures [6]. The hyperpolarizability of a nanostructure near an electric dipole has also been investigated [7]. This work develops the FDTD method to calculate the magnetic dipole moment in a torus structure with a time-varying magnetic field. This allows the magnetic susceptibilities to be found and will allow a way to simulate and optimize structures to enhance the nonlinear properties in the presence of a magnetic field.

This work implements the FDTD simulation in Python. Python is a widely-used free open source programming language that focuses on clean, high-level, readable syntax.

Python has been widely adopted for use in scientific computing and visualization due to extensive libraries such as SciPy, Matplotlib, NumPy, Seaborn, and Jupyter Notebook. Python additionally has easy interoperability with other languages such as C and Fortran. The Numba library takes advantage of this, automatically compiling and optimizing time-consuming Python functions into machine code with similar performance to low level languages such as C, C++, or Fortran.

### **Research Goals**

1. Implement the FDTD method in Python
2. Simulate and verify the behavior of the ground eigenstate of a three-dimensional torus
3. Find the magnetic dipole moment operator, implement it in the FDTD simulation and verify its accuracy
4. Implement an applied magnetic field in the FDTD method, find the magnetic dipole operator used with a magnetic field, and verify its accuracy
5. Calculate the magnetic susceptibility of a structure at a given frequency

## Chapter 2: FDTD Method

The FDTD method has been developed for use in quantum mechanics, and this chapter draws heavily from Sullivan [8], [9]. The FDTD method is used in this work to implement the Schrödinger equation. The first section describes the general implementation in simulations for three-dimensional problems. The second section describes how it is used to find eigenenergies and eigenstates.

### Schrödinger Equation Implementation

The time dependent Schrödinger equation is

$$i\hbar \frac{\partial \Psi(x, y, z, t)}{\partial t} = \left[ -\frac{\hbar^2}{2m_e} \nabla^2 + V(x, y, z) \right] \Psi(x, y, z, t), \quad (1)$$

where  $\hbar$  is Planck's constant ( $1.055 \times 10^{-34}$  J · s),  $\Psi$  is the state variable and is a function of space and time,  $m_e$  is the mass of an electron ( $9.1 \times 10^{-31}$  kg), and  $V(x, y, z)$  is the potential in  $V$  as a function of space. Eq. (1) can be rearranged to obtain

$$\frac{\partial \Psi(x, y, z, t)}{\partial t} = \left[ i \frac{\hbar}{2m_e} \nabla^2 - \frac{i}{\hbar} V(x, y, z) \right] \Psi(x, y, z, t). \quad (2)$$

From [9],  $\Psi$  contains both a real and an imaginary component,

$$\Psi(x, y, z, t) = \Psi_{real}(x, y, z, t) + i \cdot \Psi_{imag}(x, y, z, t). \quad (3)$$

This separation of  $\Psi$  allows Eq. (2) to be written as a set of equations,

$$\frac{\partial \Psi_{real}(x, y, z, t)}{\partial t} = \left[ -\frac{\hbar}{2m_e} \nabla^2 + \frac{1}{\hbar} V(x, y, z) \right] \Psi_{imag}(x, y, z, t), \quad (4a)$$

$$\frac{\partial \Psi_{imag}(x, y, z, t)}{\partial t} = \left[ \frac{\hbar}{2m_e} \nabla^2 - \frac{1}{\hbar} V(x, y, z) \right] \Psi_{real}(x, y, z, t), \quad (4b)$$

eliminating imaginary components [9]. This creates two equations which can be evaluated sequentially.

Eq. (4a) and (4b) can be expanded and rearranged to

$$\begin{aligned} \frac{\partial \Psi_{real}(x, y, z, t)}{\partial t} &= \frac{1}{\hbar} V(x, y, z) \Psi_{imag}(x, y, z, t) \\ &- \frac{\hbar}{2m_e} \left( \frac{\partial^2 \Psi_{imag}(x, y, z, t)}{\partial x^2} + \frac{\partial^2 \Psi_{imag}(x, y, z, t)}{\partial y^2} + \frac{\partial^2 \Psi_{imag}(x, y, z, t)}{\partial z^2} \right), \end{aligned} \quad (5a)$$

$$\begin{aligned} \frac{\partial \Psi_{imag}(x, y, z, t)}{\partial t} &= -\frac{1}{\hbar} V(x, y, z) \Psi_{real}(x, y, z, t) \\ &+ \frac{\hbar}{2m_e} \left( \frac{\partial^2 \Psi_{real}(x, y, z, t)}{\partial x^2} + \frac{\partial^2 \Psi_{real}(x, y, z, t)}{\partial y^2} + \frac{\partial^2 \Psi_{real}(x, y, z, t)}{\partial z^2} \right). \end{aligned} \quad (5b)$$

Time will be discrete with a step size of  $\Delta t$ . A derivative may be approximated by

$$\frac{\partial \Psi(x, y, z, t)}{\partial t} \cong \frac{\Psi(x, y, z, \Delta t \cdot (n + 1)) - \Psi(x, y, z, \Delta t \cdot (n))}{\Delta t}. \quad (6)$$

The time step is indicated by  $n$ .

Physical space can be represented by a three-dimensional array, with each element, or cell, representing a discrete portion of space. Each cell will be given an index of  $i, j$ , and  $k$  for the  $x, y$ , and  $z$  directions, respectively. Each cell has a length, width, and height of  $\Delta x, \Delta y$ , and  $\Delta z$ . From [9], the second derivative in space for  $\Psi_{real}$  may be approximated by

$$\begin{aligned} \frac{\partial^2 \Psi_{real}(i, j, k, t)}{\partial x^2} &+ \frac{\partial^2 \Psi_{real}(i, j, k, t)}{\partial y^2} + \frac{\partial^2 \Psi_{real}(i, j, k, t)}{\partial z^2} \\ &\cong \frac{\Psi_{real}(i + 1, j, k, t) - 2\Psi_{real}(i, j, k, t) + \Psi_{real}(i - 1, j, k, t)}{(\Delta x)^2} \\ &+ \frac{\Psi_{real}(i, j + 1, k, t) - 2\Psi_{real}(i, j, k, t) + \Psi_{real}(i, j - 1, k, t)}{(\Delta y)^2} \\ &+ \frac{\Psi_{real}(i, j, k + 1, t) - 2\Psi_{real}(i, j, k, t) + \Psi_{real}(i, j, k - 1, t)}{(\Delta z)^2} \end{aligned} \quad (7)$$

$i + 1$  and  $i - 1$  refer to the adjacent cells along the  $x$ -axis,  $j + 1$  and  $j - 1$  refer to adjacent cells along the  $y$ -axis, and  $k + 1$  and  $k - 1$  refer to adjacent cells along the  $z$ -axis for a cell with indices  $i, j$ , and  $k$ . A similar equation can be created for  $\Psi_{imag}$ .

For simplicity,  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  will be identical within this work and be called  $\Delta x$ . Eq. (7) can then be approximated as

$$\begin{aligned} & \frac{\partial^2 \Psi_{real}(i, j, k, t)}{\partial x^2} + \frac{\partial^2 \Psi_{real}(i, j, k, t)}{\partial y^2} + \frac{\partial^2 \Psi_{real}(i, j, k, t)}{\partial z^2} \\ & \cong \frac{1}{(\Delta x)^2} \left[ \begin{aligned} & -6\Psi_{real}(i, j, k, t) + \Psi_{real}(i+1, j, k, t) + \Psi_{real}(i-1, j, k, t) \\ & + \Psi_{real}(i, j+1, k, t) + \Psi_{real}(i, j-1, k, t) \\ & + \Psi_{real}(i, j, k+1, t) + \Psi_{real}(i, j, k-1, t) \end{aligned} \right]. \quad (8) \end{aligned}$$

Eq. (6) and (8) can be used with Eq. (5a) and (5b) to create a pair of coupled equations:

$$\begin{aligned} \frac{\Psi_{real}^{n+1}(i, j, k) - \Psi_{real}^n(i, j, k)}{\Delta t} &= \frac{1}{\hbar} V(i, j, k) \Psi_{imag}^{n+1/2}(i, j, k) \\ & - \frac{\hbar}{2m_e(\Delta x)^2} \left[ \begin{aligned} & -6\Psi_{imag}^{n+1/2}(i, j, k) + \Psi_{imag}^{n+1/2}(i+1, j, k) + \Psi_{imag}^{n+1/2}(i-1, j, k) \\ & + \Psi_{imag}^{n+1/2}(i, j+1, k) + \Psi_{imag}^{n+1/2}(i, j-1, k) \\ & + \Psi_{imag}^{n+1/2}(i, j, k+1) + \Psi_{imag}^{n+1/2}(i, j, k-1) \end{aligned} \right], \quad (9a) \end{aligned}$$

$$\begin{aligned} \frac{\Psi_{imag}^{n+3/2}(i, j, k) - \Psi_{imag}^{n+1/2}(i, j, k)}{\Delta t} &= -\frac{1}{\hbar} V(i, j, k) \Psi_{real}^{n+1}(i, j, k) \\ & + \frac{\hbar}{2m_e(\Delta x)^2} \left[ \begin{aligned} & -6\Psi_{real}^{n+1}(i, j, k) + \Psi_{real}^{n+1}(i+1, j, k) + \Psi_{real}^{n+1}(i-1, j, k) \\ & + \Psi_{real}^{n+1}(i, j+1, k) + \Psi_{real}^{n+1}(i, j-1, k) \\ & + \Psi_{real}^{n+1}(i, j, k+1) + \Psi_{real}^{n+1}(i, j, k-1) \end{aligned} \right]. \quad (9b) \end{aligned}$$

The  $n$  representing the time step is now in the superscript. The  $1/2$  in the superscript indicates a half step offset. This offset of the real and imaginary parts will allow the equations to be run sequentially.

This can be rearranged into

$$\Psi_{real}^{n+1}(i, j, k) = \Psi_{real}^n(i, j, k) + \frac{\Delta t}{\hbar} V(i, j, k) \Psi_{imag}^{n+1/2}(i, j, k) - \frac{\hbar \Delta t}{2m_e(\Delta x)^2} \left[ \begin{aligned} &-6\Psi_{imag}^{n+1/2}(i, j, k) + \Psi_{imag}^{n+1/2}(i+1, j, k) + \Psi_{imag}^{n+1/2}(i-1, j, k) \\ &+ \Psi_{imag}^{n+1/2}(i, j+1, k) + \Psi_{imag}^{n+1/2}(i, j-1, k) \\ &+ \Psi_{imag}^{n+1/2}(i, j, k+1) + \Psi_{imag}^{n+1/2}(i, j, k-1) \end{aligned} \right], \quad (10a)$$

$$\Psi_{imag}^{n+3/2}(i, j, k) = \Psi_{imag}^{n+1/2}(i, j, k) - \frac{\Delta t}{\hbar} V(i, j, k) \Psi_{real}^{n+1}(i, j, k) + \frac{\hbar \Delta t}{2m_e(\Delta x)^2} \left[ \begin{aligned} &-6\Psi_{real}^{n+1}(i, j, k) + \Psi_{real}^{n+1}(i+1, j, k) + \Psi_{real}^{n+1}(i-1, j, k) \\ &+ \Psi_{real}^{n+1}(i, j+1, k) + \Psi_{real}^{n+1}(i, j-1, k) \\ &+ \Psi_{real}^{n+1}(i, j, k+1) + \Psi_{real}^{n+1}(i, j, k-1) \end{aligned} \right]. \quad (10b)$$

Each time step iteration will run the equation for the new  $\Psi_{real}$  followed by the equation for the new  $\Psi_{imag}$ . This process may be repeated for the desired number of time steps. These equations must be evaluated for each cell in the three-dimensional array at each time step. Eq. (10a) and (10b) are easily programmable.

To ensure stability, the conditions

$$\frac{\hbar \Delta t}{2m_e(\Delta x)^2} \leq 0.15, \quad (11a)$$

$$\frac{\Delta t}{\hbar} V(i, j, k) \leq 0.15 \quad (11b)$$

must be met [9]. Otherwise small values can quickly reach numbers large enough to cause memory overflow from stability problems. For example, if a large potential barrier is used,  $\Psi$  is initially contained in the low potential. However, Eq. (10a) and (10b) will cause the cells in the high potential area, but adjacent to the low potential, to have a small value for  $\Psi$ . If the potential of the barrier is large enough, this value will get larger on each successive iteration and eventually cause problems.



### FDTD Method to Find Eigenenergies and Eigenstates

The method used to find the eigenenergies and eigenfunctions using the FDTD implementation follows the methodology described by Sullivan [9], [10].

#### *Finding the Eigenenergy*

The most general wavefunction can be expressed as a summation of the eigenstates and sinusoids with frequencies corresponding to the eigenenergies, as in

$$\Psi(x, y, z, t) = \sum_{n=1}^{\infty} c_n \phi_n(x, y, z) e^{-i\omega_n t}. \quad (12)$$

The relationship between frequency and energy is

$$\epsilon_n = hf_n = \hbar(2\pi f_n) = \hbar\omega_n. \quad (13)$$

Propagating a test function with the form given by Eq. (12) can be used to find the eigenstates. The main consideration for the shape is whether it contains the desired eigenstate. A narrow test pulse will contain components of many eigenstates, allowing each eigenenergy associated with those eigenstates to be found.

If  $\Psi$  is monitored only at one specific test location, Eq. (12) becomes,

$$\Psi(x_0, y_0, z_0, t) = \sum_{n=1}^{\infty} c_n \phi_n(x_0, y_0, z_0) e^{-i\omega_n t}, \quad (14)$$

which is only time dependent. This will create a one-dimensional array if  $\Psi$  is recorded over time.

Eq. (14) is still a summation of sinusoids, each at a frequency corresponding to an eigenenergy. These frequency-dependent components can be examined without any

knowledge of the eigenstates simply by examining the value of  $\Psi$  at a specific point over time. The  $\Psi$  over time values are multiplied by a Hanning window of the form of

$$H(t) = \frac{1}{2} \left( 1 - \cos \left( \frac{2\pi \cdot t}{N} \right) \right), \quad (15)$$

where  $N$  is the number of iterations. This is to reduce errors introduced from the abrupt start of the simulation.

The frequency components of  $\Psi$  over time can be extracted using a Fourier transform. Since the frequency can be converted to energy using Eq. (13), the Fourier transform can be displayed in terms of energy. This can be used to find the eigenenergies. The "spike" at the lowest energy will be the ground state eigenenergy (if the test pulse contained a component of the ground eigenstate).

#### *Finding the Eigenfunction*

The eigenstate can be found with a corresponding known eigenenergy. The ground eigenfunction is of the form:

$$\phi_0(x, y, z, t) = c_0 \phi_0(x, y, z) e^{-i \frac{\epsilon_0}{\hbar} t}. \quad (16)$$

Instead of  $\omega_0$ , the frequency associated with the ground eigenstate,  $\frac{\epsilon_0}{\hbar}$  is used since the ground eigenenergy,  $\epsilon_0$ , is known.

The function  $\phi_0(x, y, z)$  is the ground eigenstate. In order to isolate this from the test pulse that was shown in Eq. (12), the following equation is used:

$$\int_{-\infty}^{\infty} \left[ \sum_{n=1}^N c_n \phi_n(x, y, z) e^{-i \frac{\epsilon_n}{\hbar} t} \right] e^{i \frac{\epsilon_0}{\hbar} t} dt = 2\pi c_0 \phi_0(x, y, z), \quad (17)$$

where only the first  $N$  are relevant. This is essentially a Fourier transform to isolate the eigenstate corresponding with the applied eigenenergy. This result can be implemented with a computer program by approximating a Fourier transform by using a summation over a large number of points. A longer summation running a larger number of iterations will be more accurate. The summation equivalent is

$$\phi_0(x, y, z) = \int_{-\infty}^{\infty} \Psi(x, y, z, t) e^{i\frac{\epsilon_0}{\hbar}t} dt \cong \sum_{m=1}^M \Psi(x, y, z, \Delta t \cdot m) e^{i\frac{\epsilon_0}{\hbar}\Delta t \cdot m}. \quad (18)$$

For simplicity, the  $2\pi c_0$  is dropped from the left-hand term at this point. This will be accounted for at the end of this section. Eq. (18) would need to be calculated at every iteration, resulting in large computational time. However, it can be observed that each iteration will simply add a new term to the summation. This way a running total of the Fourier transform can be saved, and only one additional term is added per cycle as in

$$\begin{aligned} \phi_0^M(x, y, z) &= \sum_{m=1}^{M-1} \Psi(x, y, z, \Delta t \cdot m) e^{i\frac{\epsilon_0}{\hbar}\Delta t \cdot m} + \Psi(x, y, z, \Delta t \cdot M) e^{i\frac{\epsilon_0}{\hbar}\Delta t \cdot M} \\ &= \phi_0^{M-1}(x, y, z) + \Psi(x, y, z, \Delta t \cdot M) e^{i\frac{\epsilon_0}{\hbar}\Delta t \cdot M} \end{aligned} \quad (19)$$

The identity

$$e^{ix} = \cos(x) + i \cdot \sin(x) \quad (20)$$

can be used with Eq. (19) to create

$$\begin{aligned} \phi_0^M(x, y, z) &= \phi_0^{M-1}(x, y, z) \\ &+ \Psi(x, y, z, \Delta t \cdot M) \left[ \cos\left(\frac{\epsilon_0}{\hbar} \Delta t \cdot M\right) + i \cdot \sin\left(\frac{\epsilon_0}{\hbar} \Delta t \cdot M\right) \right]. \end{aligned} \quad (21)$$

The separation of  $\Psi$  into real and imaginary parts, as was done in Eq. (3), can be used to create a function ready to be implemented in the computer program using:

$$\begin{aligned} \phi_{real,0}^M(x, y, z) = & \phi_{real,0}^{M-1}(x, y, z) + \Psi_{real}(x, y, z, \Delta t \cdot M) \cos\left(\frac{\epsilon_0}{\hbar} \Delta t \cdot M\right) \\ & - \Psi_{imag}(x, y, z, \Delta t \cdot M) \sin\left(\frac{\epsilon_0}{\hbar} \Delta t \cdot M\right) \end{aligned} \quad (22a)$$

$$\begin{aligned} \phi_{imag,0}^M(x, y, z) = & \phi_{imag,0}^{M-1}(x, y, z) + \Psi_{imag}(x, y, z, \Delta t \cdot M) \cos\left(\frac{\epsilon_0}{\hbar} \Delta t \cdot M\right) \\ & + \Psi_{real}(x, y, z, \Delta t \cdot M) \sin\left(\frac{\epsilon_0}{\hbar} \Delta t \cdot M\right) \end{aligned} \quad (22b)$$

Further details regarding this implementation can be found in Sullivan [11]. In a similar manner to the way the Hanning window in Eq. (15) was used in the eigenenergy calculation, the Hanning window  $H(t)$  is also used to reduce some of the dependency on when the simulation ends and removes effects of an abrupt start. Thus, the simulation's length is defined at the beginning of the simulation. A simulation cannot simply be run for additional time for a more accurate eigenstate once a simulation ends. The Hanning window is applied as in

$$\begin{aligned} \phi_{real,0}^M(x, y, z) = & \phi_{real,0}^{M-1}(x, y, z) \\ & + H(\Delta t \cdot M) \Psi_{real}(x, y, z, \Delta t \cdot M) \cos\left(\frac{\epsilon_0}{\hbar} \Delta t \cdot M\right), \\ & - H(\Delta t \cdot M) \Psi_{imag}(x, y, z, \Delta t \cdot M) \sin\left(\frac{\epsilon_0}{\hbar} \Delta t \cdot M\right) \end{aligned} \quad (23a)$$

$$\begin{aligned} \phi_{imag,0}^M(x, y, z) = & \phi_{imag,0}^{M-1}(x, y, z) \\ & + H(\Delta t \cdot M) \Psi_{imag}(x, y, z, \Delta t \cdot M) \cos\left(\frac{\epsilon_0}{\hbar} \Delta t \cdot M\right), \\ & + H(\Delta t \cdot M) \Psi_{real}(x, y, z, \Delta t \cdot M) \sin\left(\frac{\epsilon_0}{\hbar} \Delta t \cdot M\right) \end{aligned} \quad (23b)$$

giving the final version of the equation used for finding the ground state's shape.

One final adjustment is needed. Eq. (17) showed the Fourier transform at a specific frequency (associated with a specific eigenenergy) resulted in  $2\pi c_0 \phi_0(x, y, z)$ . The  $2\pi c_0$  was dropped for the above evaluation. Eq. (23a) and (23b) then ignore the effects of this term. Since it is known that eigenstates are orthonormal, and

$$\int_{-\infty}^{\infty} \phi_0^*(x, y, z) \phi_0(x, y, z) dx dy dz = 1 \quad (24)$$

holds, the eigenstate found using Eq. (23a) and (23b) must be normalized. This will allow a true eigenstate to be found. This normalization can be done using the procedure in

$$\phi_{real,0,normalized}(x,y,z) = \frac{\phi_{real,0}(x,y,z)}{\sqrt{(\phi_{real,0}(x,y,z))^2 + (\phi_{imag,0}(x,y,z))^2}}, \quad (25a)$$

$$\phi_{imag,0,normalized}(x,y,z) = \frac{\phi_{imag,0}(x,y,z)}{\sqrt{(\phi_{real,0}(x,y,z))^2 + (\phi_{imag,0}(x,y,z))^2}}. \quad (25b)$$

Other ground states may be found simply by using a different eigenenergy,  $\epsilon_n$ .

### Chapter 3: Simulation of a Torus

This chapter details the simulation of a torus, as in Fig. 1, using the FDTD method described in Chapter 2. The torus is chosen in this work as a model for a loop of wire.

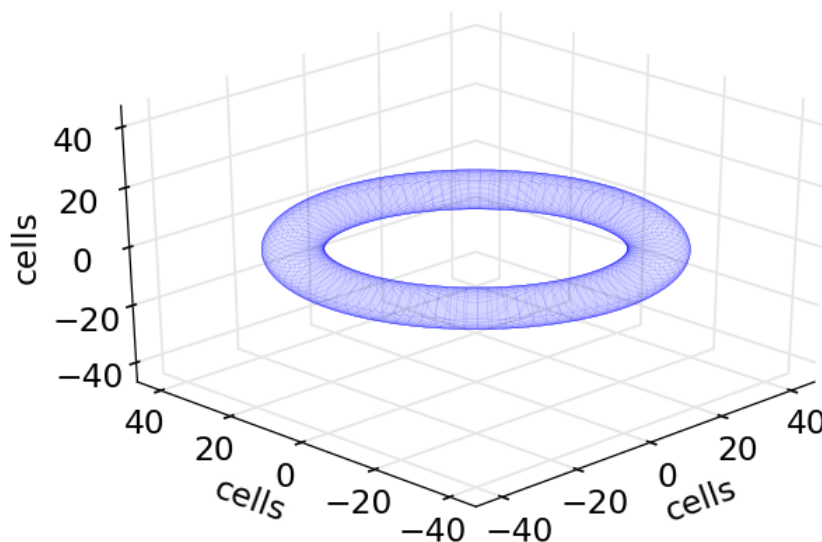


Figure 1. Torus structure

The first section will detail the setup of the simulation, including the perfectly matched layer (PML) that is used to dampen the discontinuity at the edge of the simulation space. The second section describes the setup of the potential to create a torus. The third section will address details of the eigenenergy and eigenstate generation. The final section will verify the behavior of the ground eigenstate over time.

#### Simulation Setup

The simulation is set up in three dimensions. The problem space is  $100 \times 100 \times 30$  cells, with each cell representing one  $\text{\AA}^3$ . The number of cells is chosen as a balance between

computational time and accuracy. Simulation time is drastically increased with additional cells, but more granularity leads to more precise results.

Each dimension of the cell ( $\Delta x$ ) measures one Å so the time step size ( $\Delta t$ ) is chosen to be  $1.09 \times 10^{-17}$  s based on Eq. (11a). A smaller  $\Delta t$  would also work.

The torus has a radius of 35 Å and a tube radius of six Å. The goal is to create a torus with a high torus radius to tube radius ratio, while maintaining good behavior from the simulation. The torus is defined according to

$$r_{tube}^2 = \left( r_{torus} - \sqrt{x^2 + y^2} \right)^2 + z^2, \quad (26)$$

with  $r_{torus}$  as the torus radius and  $r_{tube}$  as the tube radius.

The problem space is illustrated in Fig. 2 and is split as detailed below.

- Five cells on each side are a PML (detailed in subsequent section). This leaves 90x90x20 cells for the torus structure.
- The torus radius is 35 cells and the tube radius is six cells, meaning it occupies a maximum of 82x82x12 cells.
- A minimum of four cells are between the edge of the torus and the PML on each side, in each dimension.

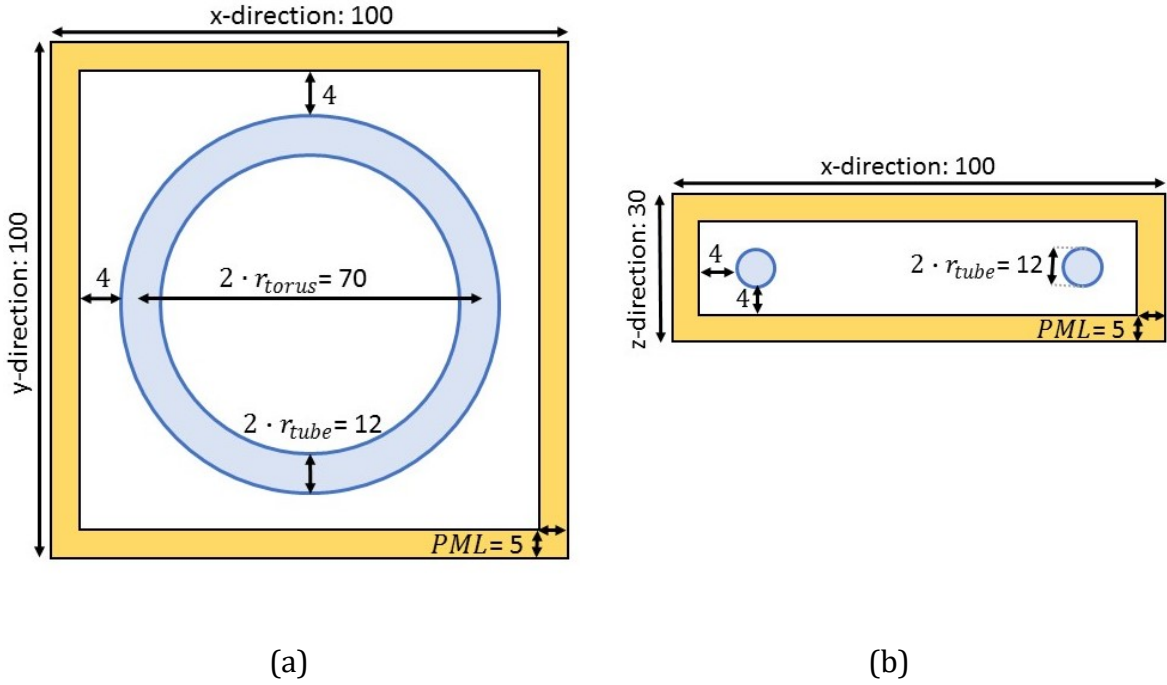


Figure 2. Problem space (defined in cells and not drawn to scale) in the (a) x-y direction, and (b) x-z direction.

### Perfectly Matched Layer (PML)

A five-cell boundary is used for a PML to absorb outgoing waves around the periphery. This is necessary to avoid waveforms reflecting at the boundary of the problem space and creating errors. The PML was originally developed for FDTD electromagnetic simulations [12]. The PML was subsequently modified for the Schrödinger equation [13]. The version used in these simulations was previously described [14]. The cells at the lower boundary of each direction are described by

$$PML_x = 1 - 0.5 \cdot \left( \frac{PML_{cells} - i}{PML_{cells}} \right)^3, \quad (27a)$$

$$PML_y = 1 - 0.5 \cdot \left( \frac{PML_{cells} - j}{PML_{cells}} \right)^3, \quad (27b)$$

$$PML_z = 1 - 0.5 \cdot \left( \frac{PML_{cells} - k}{PML_{cells}} \right)^3. \quad (27c)$$



A similar set of equations is used at the upper boundaries. The cell's index in the array is defined as  $i, j$ , and  $k$ , and the number of cells in the PML is  $PML_{cells}$ . The PML applies to cells 0 to 4 at the lower boundary, and to cells 95 to 99 at the upper boundary in the x- or y-directions. The PML applies to cells 0 to 4 at the lower boundary and cells 25 to 29 at the upper boundary in the z-direction.

$PML_x, PML_y$ , and  $PML_z$  are each three-dimensional arrays of size 100x100x30 cells. Any cells outside the PML are set to one. All three arrays are then multiplied together so

$$PML_{total} = PML_x \cdot PML_y \cdot PML_z. \quad (28)$$

The result is an array with the same dimensions as the problem space. It has a value of one assigned to the majority of the cells. The five cells along the peripheral in any direction show a drop from one to zero. Cells in the corners show a sharper drop.

This complete PML layer,  $PML_{total}$ , is then used in the FDTD simulation as in

$$\begin{aligned} \Psi_{real}^{n+1}(i, j, k) = & PML_{total}(i, j, k) \cdot \Psi_{real}^n(i, j, k) + \frac{\Delta t}{\hbar} V(i, j, k) \Psi_{imag}^{n+1/2}(i, j, k) \\ & - \frac{\hbar \Delta t}{2m_e(\Delta x)^2} \left[ \begin{aligned} & -6\Psi_{imag}^{n+1/2}(i, j, k) + \Psi_{imag}^{n+1/2}(i+1, j, k) + \Psi_{imag}^{n+1/2}(i-1, j, k) \\ & + \Psi_{imag}^{n+1/2}(i, j+1, k) + \Psi_{imag}^{n+1/2}(i, j-1, k) \\ & + \Psi_{imag}^{n+1/2}(i, j, k+1) + \Psi_{imag}^{n+1/2}(i, j, k-1) \end{aligned} \right], \end{aligned} \quad (29a)$$

$$\begin{aligned} \Psi_{imag}^{n+3/2}(i, j, k) = & PML_{total}(i, j, k) \cdot \Psi_{imag}^{n+1/2}(i, j, k) - \frac{\Delta t}{\hbar} V(i, j, k) \Psi_{real}^{n+1}(i, j, k) \\ & + \frac{\hbar \Delta t}{2m_e(\Delta x)^2} \left[ \begin{aligned} & -6\Psi_{real}^{n+1}(i, j, k) + \Psi_{real}^{n+1}(i+1, j, k) + \Psi_{real}^{n+1}(i-1, j, k) \\ & + \Psi_{real}^{n+1}(i, j+1, k) + \Psi_{real}^{n+1}(i, j-1, k) \\ & + \Psi_{real}^{n+1}(i, j, k+1) + \Psi_{real}^{n+1}(i, j, k-1) \end{aligned} \right], \end{aligned} \quad (29b)$$

a modification of Eq. (10a) and (10b) that were previously derived.

### Setting up the Potential

The basic idea when setting up the potential is to create a low potential within the torus, and a high potential outside the torus. This confines the electron to the torus, which allows it to move freely within. The background potential is 4.6 eV, which is the work function of silver. It is high enough to contain the electron, while still allowing stability from Eq. (11b). The goal will be to create a three-dimensional array to represent the potential,  $V(i, j, k)$ .

One method of creating a torus is to look at each cell, and to calculate the results of the inequality

$$r_{tube}^2 > \left( r_{torus} - \sqrt{(i - X_C)^2 + (j - Y_C)^2} \right)^2 + (k - Z_C)^2, \quad (30)$$

where  $X_C$ ,  $Y_C$ , and  $Z_C$  are the centers of the problem space in the x, y and z directions, respectively. The cells are indexed with  $i$ ,  $j$ , and  $k$ , and  $r_{tube}$  and  $r_{torus}$  are given in cells. This inequality will allow the formation of a solid torus. Eq. (26) is the equation for the surface of a torus.

The potential is then set to 0 eV if the expression is true, and 4.6 eV if it is not true. Each cell will then have one of two possible values. This method can work well for a large problem space. However, in a limited problem space, the individual cells will cause irregularities in the torus. This in turn creates irregularities in the results, particularly in the eigenstates.

Another option is to calculate an average potential as described in [15]. By averaging, a larger problem space can be imitated. This can be done by dividing each cell into subcells, and calculating what the potential should be at the position of the subcell. The final cell's potential is assigned a percentage of the full surrounding potential. For example, if half the subcells of a given cell are contained within the torus and half are

outside the torus, the cell is assigned half the surrounding potential. An illustration of this concept in two dimensions for a circle is shown in Fig. 3.

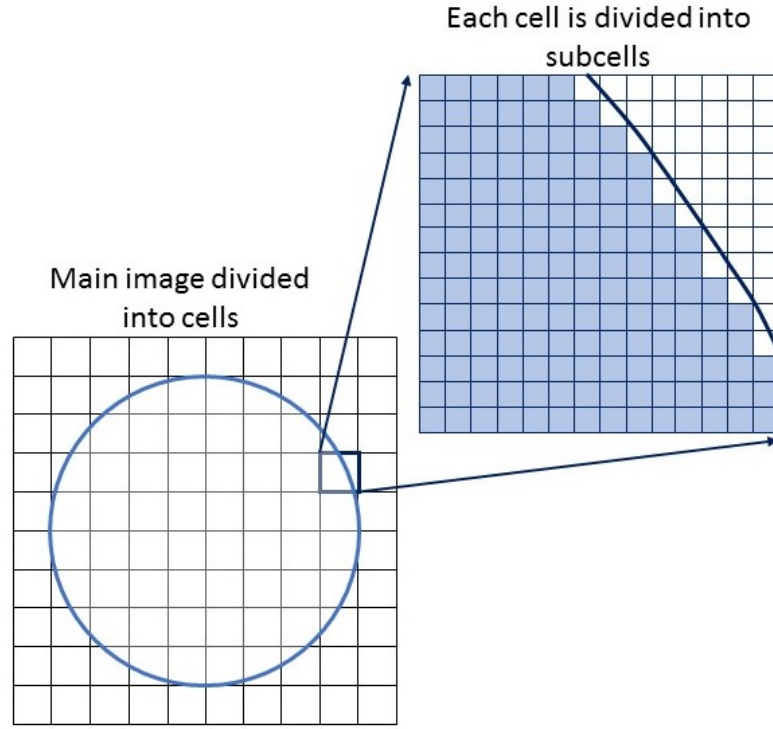


Figure 3. Illustration of subcells

The potential for a torus is calculated by

$$r_{tube}^2 > \left( r_{torus} - \sqrt{\left( i - \frac{i_{sub}}{N} - XC \right)^2 + \left( j - \frac{j_{sub}}{N} - YC \right)^2} \right)^2 + \left( k - \frac{k_{sub}}{N} - ZC \right)^2, \quad (31)$$

where  $N$  indicates the number of subcells and  $i_{sub}$ ,  $j_{sub}$ , and  $k_{sub}$  refer to the subcell (indexed from  $-(N-1)/2$  to  $(N-1)/2$ ). Again  $i$ ,  $j$ , and  $k$  are the cell indices and all values are in terms of cells. Twenty-nine subcells were used in this work to determine the torus potential used in calculations. This means the expression had to be calculated  $29^3=24389$  more times than the method without averaging. Since the potential only

needs to be calculated once for a given structure, the extra computational time is not problematic.

Fig. 4 illustrates the difference between the two methods of calculating the potential of the torus. The first set of images shows a cross section through the center of the z-axis, and the second set shows a cross section through the center of the y-axis. The torus looks significantly smoother using the averaging method, and the averaging method was used in this work.

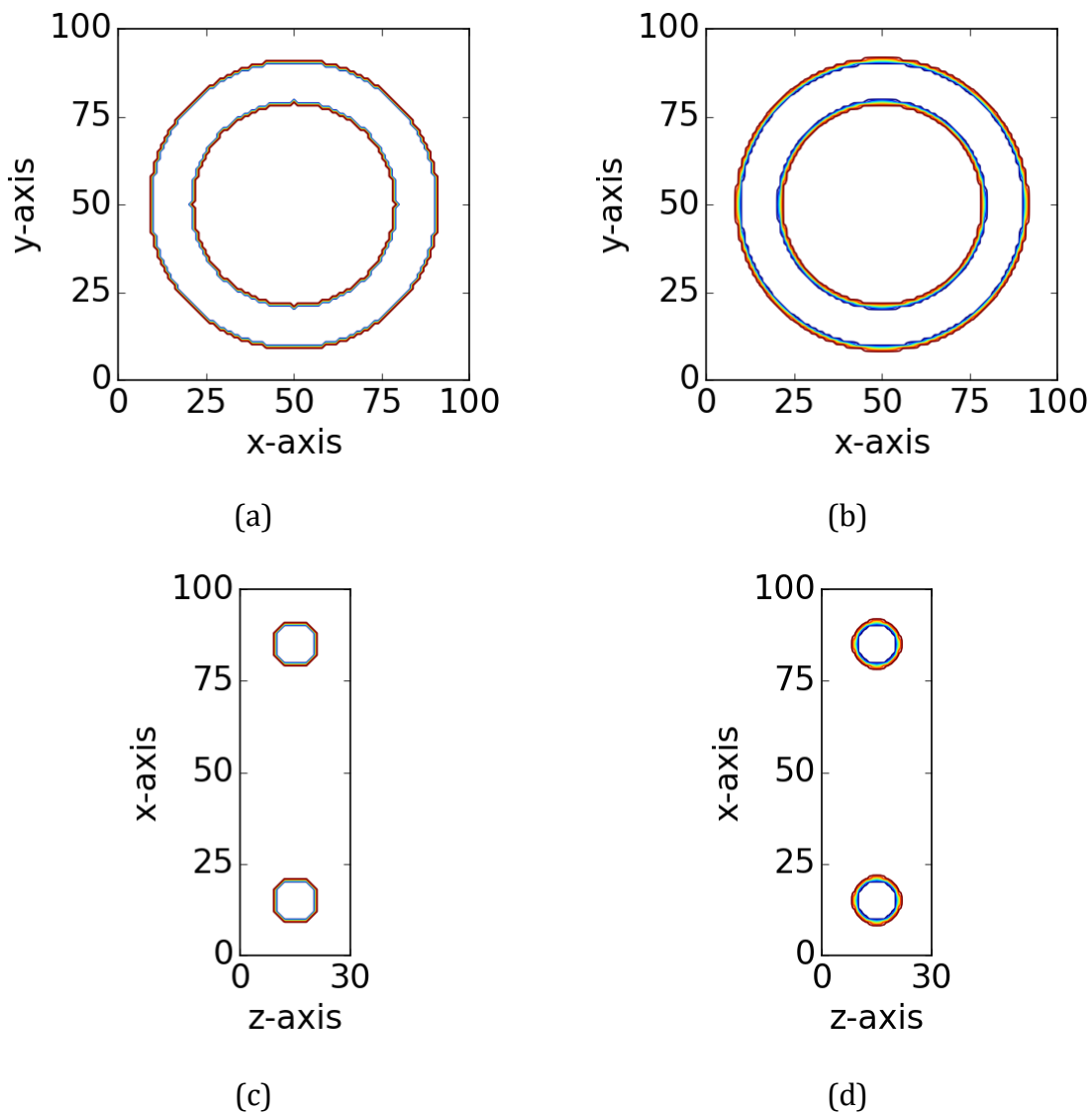


Figure 4. Contour plots of the potential (a) without averaging over subcells through center of z-axis, and (b) with averaging over subcells through center of z-axis, (c) without averaging over subcells through center of y-axis, and (d) with averaging over subcells through center of y-axis

### Determining the Ground State Eigenfunction of the Torus

Before finding the ground eigenstate, the ground state eigenenergy is found. First a small pulse is initialized within the torus, as is shown in Fig. 5. This pulse is a three-dimensional sine wave contained within a Gaussian envelope. This will be a superposition of many eigenstates.

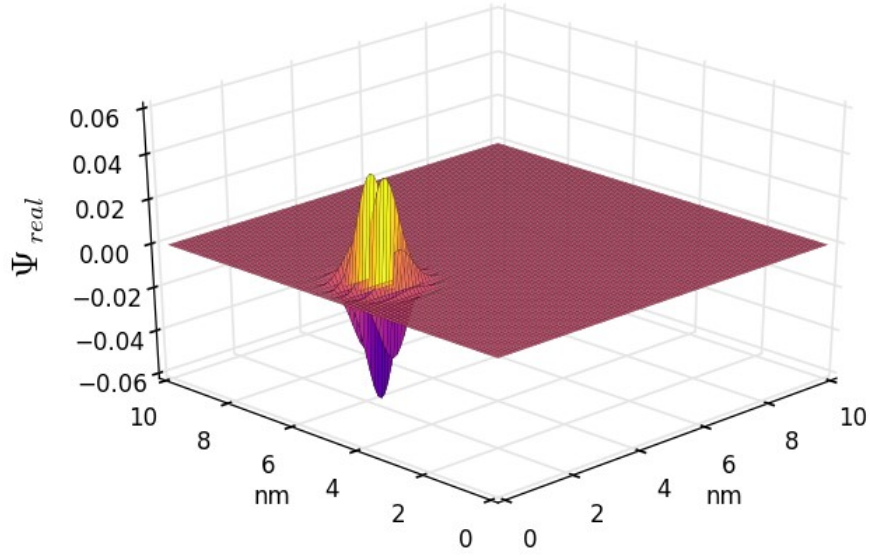


Figure 5. Test pulse initialized within a torus

The ground state's eigenenergy is found as described in Chapter 2. The test pulse is the initial  $\Psi_{real}$  used in Eq. (29a) and (29b). A similar pulse generated with a cosine wave within a Gaussian envelope is used for  $\Psi_{imag}$ . At this point, each term in Eq. (29a) and (29b) is defined.

At each iteration,  $\Psi_{real}$  is monitored at a test location within the torus at  $(X_c + r_{torus}, Y_c, Z_c)$  where  $X_c$ ,  $Y_c$ , and  $Z_c$  represent the center array index along the x-, y-, and z-directions. This is multiplied by the Hanning window of Eq. (15), and the Fourier transform is taken. The x-axis of the Fourier transform has the step size,  $\Delta E$ , shown in

$$\Delta E = \frac{2\pi\hbar}{\Delta t} \cdot \frac{1}{NN}, \quad (32)$$

where  $NN$  is the number of time steps for the simulation. The equation is based off the relationship between frequency and energy in Eq. (13). The spike at the lowest energy is the ground state eigenenergy.

An example of  $\Psi_{real}$  over time is shown in Fig. 6, and the output of the Fourier transform is shown in Fig. 7. The output in Fig. 7 was created with only 10,000 iterations so the resolution in the Fourier transform graph is fairly low. To obtain an accurate eigenenergy, far more iterations are necessary (on the order of 100,000).

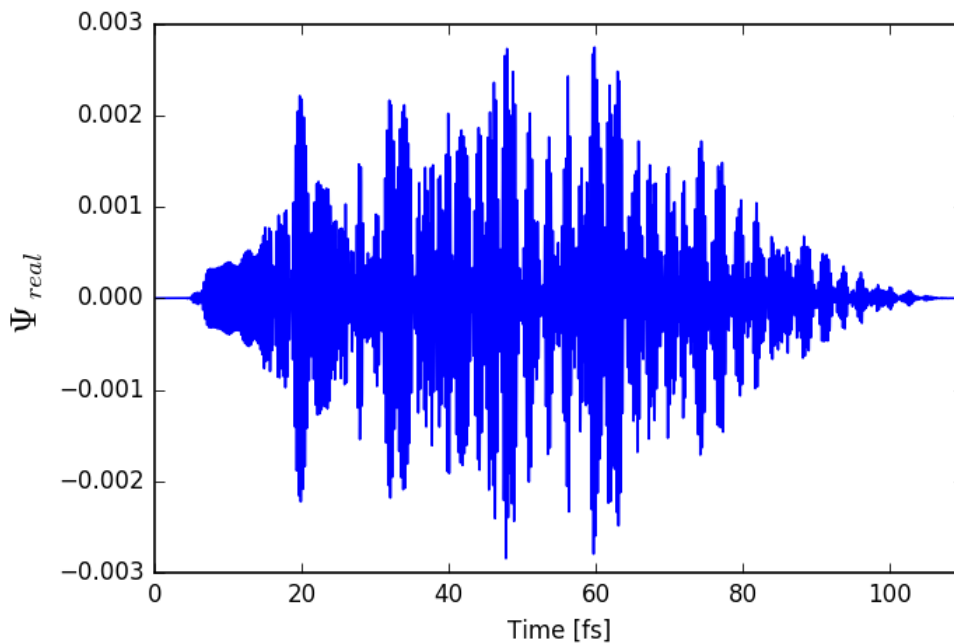


Figure 6.  $\Psi_{real}$  over time using 10,000 iterations of a test pulse

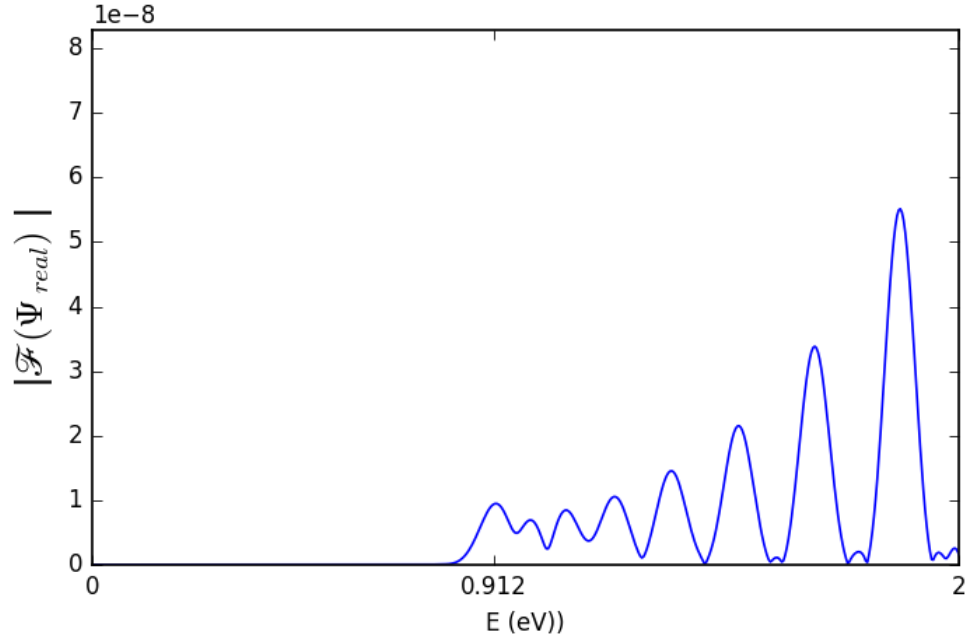


Figure 7. Eigenenergy distribution using 10,000 iterations of a test pulse

Table 1 shows the number of iterations run ( $N$  in Eq. (32)), with the calculated ground state eigenenergy and the eigenenergy step size ( $\Delta E$  in Eq. (32)). This shows the relationship between the eigenenergy accuracy and the number of iterations run. 200,000 iterations was chosen for this work since there was little difference between eigenenergies at 200,000 and 300,000 iterations, and the eigenenergy step size was less than 2 meV, which is adequate resolution.



*Table 1. Eigenenergy compared with the number of iterations of the FDTD simulation*

Number of Iterations	Ground State Eigenenergy [eV]	Eigenenergy Step Size [meV]
10,000	0.912	38.0
25,000	0.912	15.2
50,000	0.912	7.6
100,000	0.897	3.8
150,000	0.900	2.5
200,000	0.899	1.9
300,000	0.898	1.3

A test pulse will work well for finding the eigenenergies, since it will contain a combination of many eigenstates. This ideally will allow the Fourier transform to reveal a complete set of eigenenergies. However, a test pulse requires many iterations to calculate a stable ground eigenstate. It would take several hundred thousand iterations to obtain an accurate ground eigenstate.

An alternative method is to create a test structure that mimics the desired eigenstate. Since the final shape of the ground eigenstate of the torus is generally known, it is possible to estimate what the shape of the ground eigenstate will be. This shape can be used for the initial test function instead of a pulse. Fig. 8 shows the initial shape used to estimate the ground eigenstate of a torus.

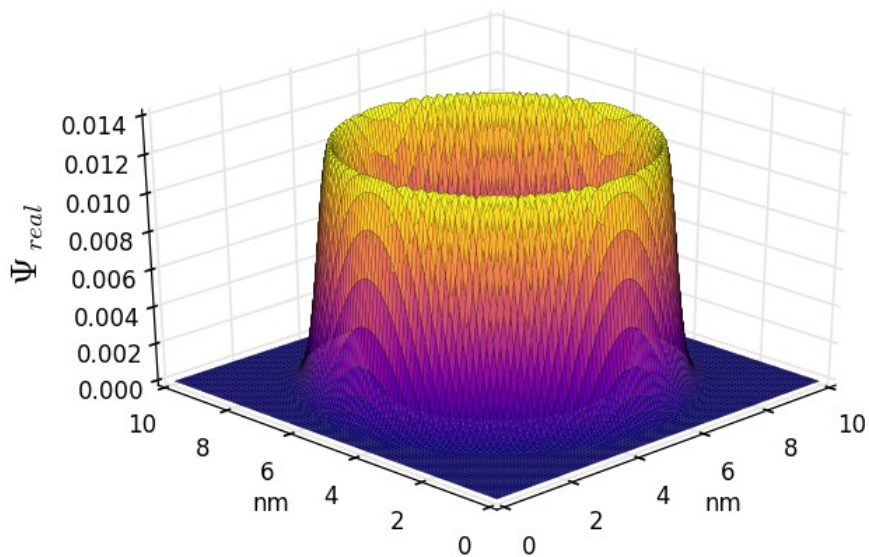


Figure 8. Test function initialized within a wire

The values of  $\Psi_{real}$  over time within a Hanning window are shown in Fig. 9, and the corresponding Fourier transform is shown in Fig. 10. Both graphs use a test function as shown in Fig. 8. These results were again created with only 10,000 iterations to allow for a clear plot of  $\Psi_{real}$  over time; a larger number of iterations would yield a higher resolution plot of the eigenenergies with a more distinct peak. These graphs can be compared with Fig. 6 and Fig. 7, which showed the results using a test pulse. It is clear the test function only allows the ground eigenenergy to be found, but the test pulse allows many eigenenergies to be found. The test function is a good estimate for the final shape of the ground eigenstate, since only components of the ground eigenenergy were prominent in the Fourier transform's single, distinct peak.

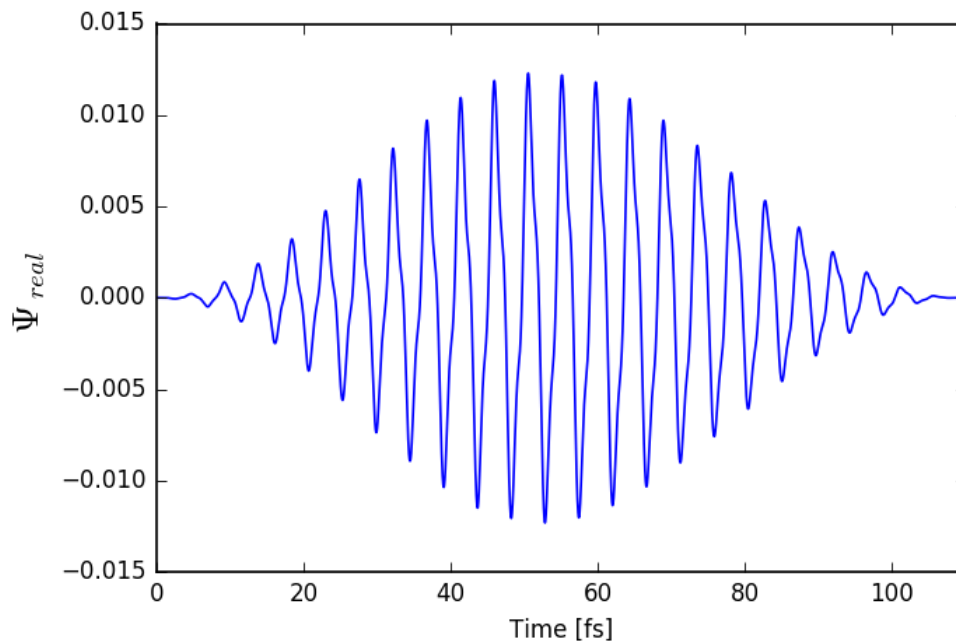


Figure 9.  $\Psi_{real}$  over time using 10,000 iterations of a test function

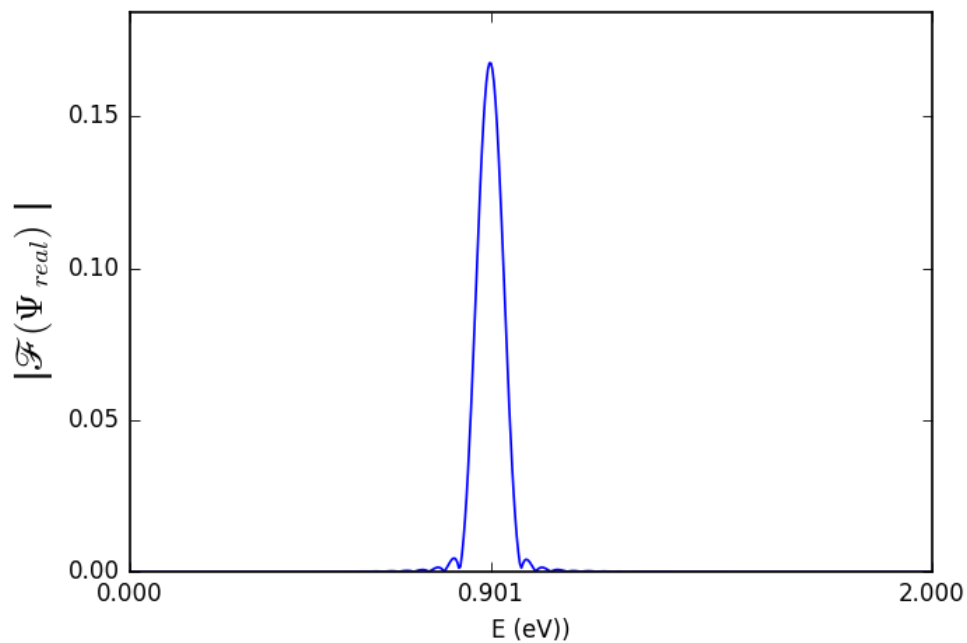
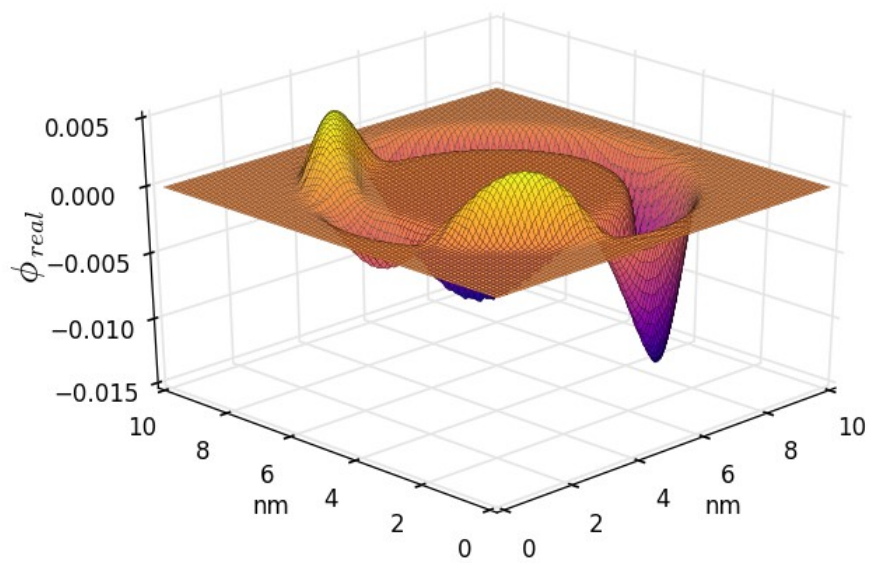


Figure 10. Eigenenergy distribution using 10,000 iterations of a test function

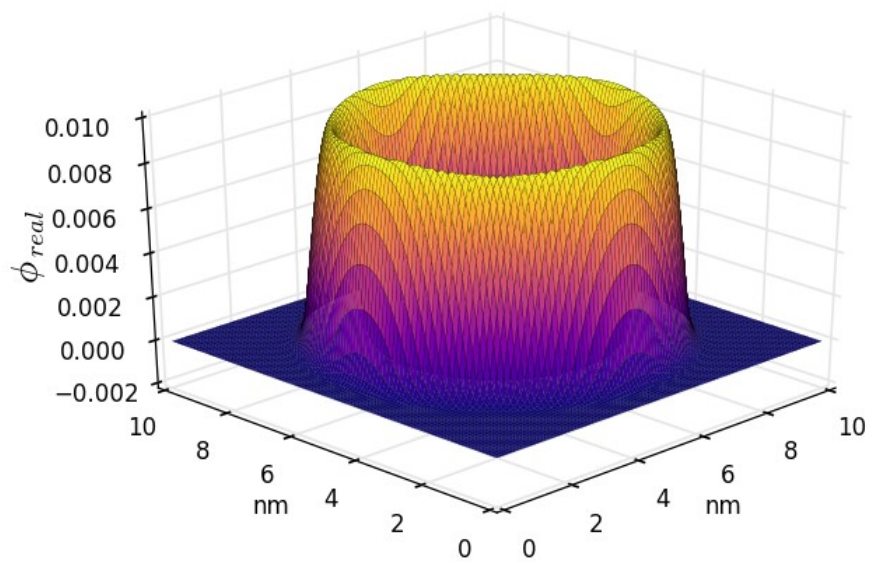
The ground eigenstates are generated by running the FDTD simulation with Eq. (29a) and (29b) again, with a known ground state eigenenergy, and calculating Eq. (23a) and

(23b) on each iteration. At the end of the simulation,  $\phi_{real,0}(x, y, z)$  and  $\phi_{imag,0}(x, y, z)$  are normalized to yield the final ground eigenstate.

The test pulse is effective in generating the ground eigenstate for certain shapes, but this method takes a long time for a torus. Fig. 11 shows a comparison of the ground eigenstates generated after 10,000 iterations using the test pulse compared with the test function. The ground eigenstate generated by the test function has settled into a fairly even ground eigenstate after 10,000 iterations, but the ground eigenstate generated using the test pulse is still uneven.



(a)



(b)

Figure 11. Ground eigenstate generated over 10,000 iterations (a) using a test pulse, and (b) using a test function

An accurate ground eigenstate is necessary, so the test function is far more efficient. 200,000 iterations created an even ground eigenstate using the test function; this number was chosen based on the eigenenergy accuracy. This was later verified using dipole moment calculations, which are described in Chapter 4. The ground eigenstate used in this work is shown in Fig. 12.

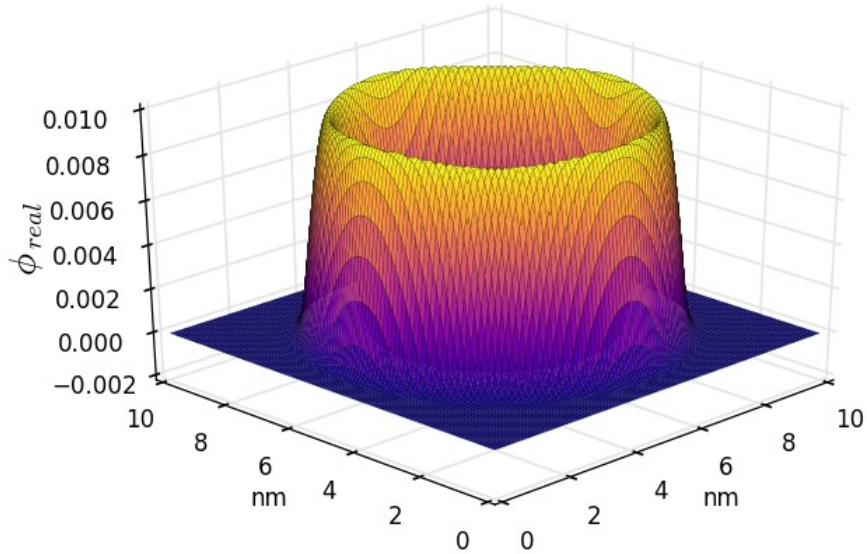


Figure 12. Ground eigenstate generated over 200,000 iterations using a test function

### Ground Eigenstate Verification

The ground eigenstate must oscillate with a period given by

$$T = \frac{h}{\epsilon_0}, \quad (33)$$

and it must maintain normalization [9].  $\epsilon_0$  is the ground state eigenenergy. The period can be a way to verify the eigenstate and eigenenergy are accurate.  $\Psi_{real}$ , initially in the generated ground eigenstate, can be monitored during an FDTD simulation to ensure it retains its shape and revives after the correct number of cycles predicted by the equation.

The ground eigenenergy of the torus described in this chapter is 0.899 eV when generated over 200,000 iterations. This means the revival time will be 4.601 fs from

$$T = \frac{4.136 \times 10^{-15} \text{eV} \cdot \text{s}}{0.899 \text{eV}} = 4.601 \text{fs}. \quad (34)$$

With a  $\Delta t = 1.09 \times 10^{-17} \text{s}$ , the revival time will be approximately 422 time steps. This can be verified by loading the ground eigenstate as  $\Psi_{real}$  and  $\Psi_{imag}$ , and then running the FDTD program a designated number of iterations to ensure that the result is the expected behavior, as shown in Fig. 13. A full period was shown to be 422 time steps, with the ground eigenstate inverted after 211 time steps.

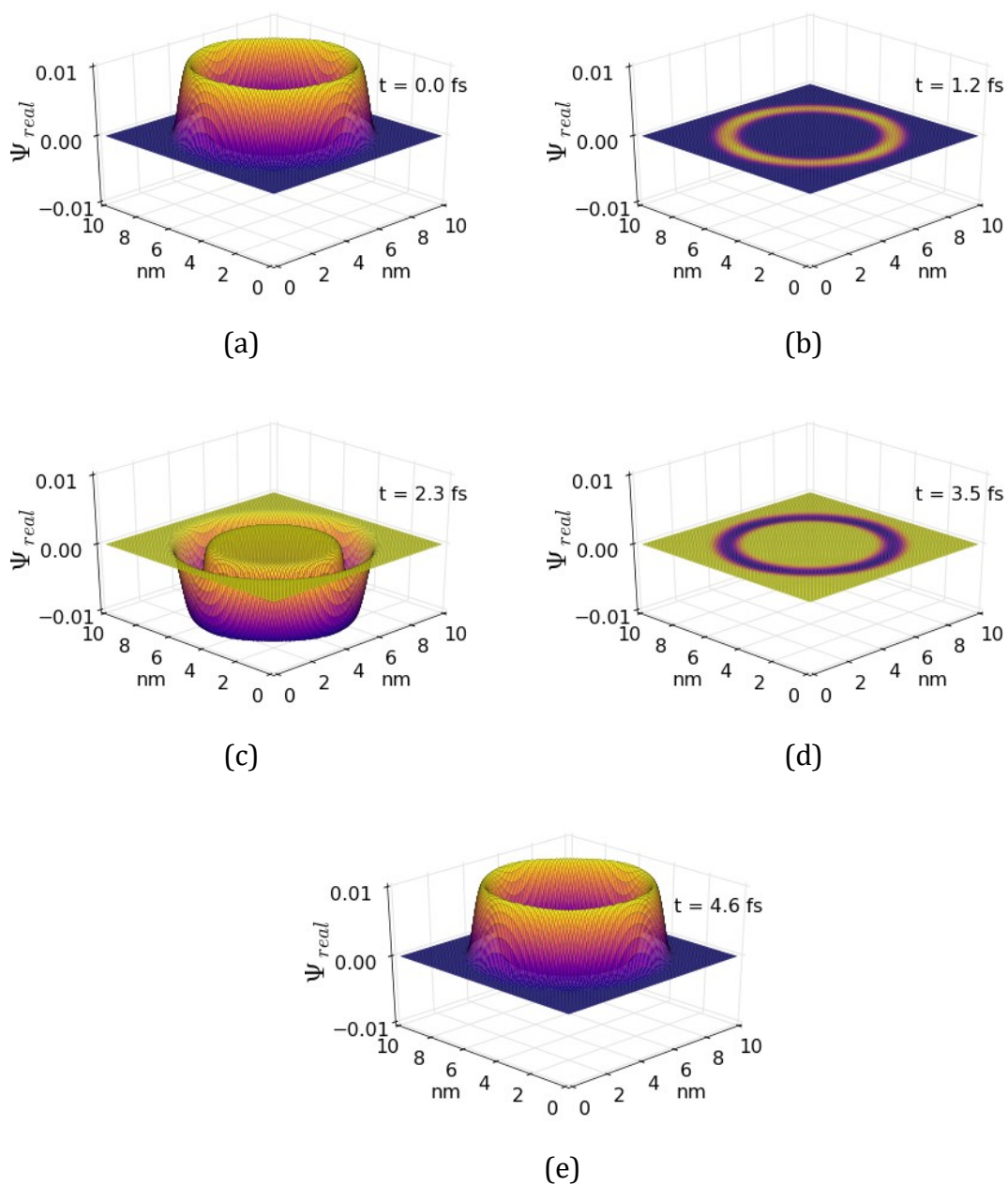


Figure 13. Ground eigenstate after (a) 0 iterations (initial state), (b) 106 iterations (quarter cycle), (c) 211 iterations (half cycle), (d) 317 iterations (three-quarters of a cycle), and (e) 422 iterations (full cycle).



## Chapter 4: Magnetic Dipole Moment Operator and the Simulation of a Magnetic Field

This chapter describes the magnetic dipole moment operator and the effects of a magnetic field on the torus.

### Calculating the Magnetic Dipole Moment

The magnetic dipole moment operator ( $\mathbf{m}$ ) is

$$\mathbf{m} = \frac{q}{2m_e} \mathbf{l}, \quad (35)$$

where  $\mathbf{l}$  is the angular momentum,  $q$  is the charge of an electron, and  $m_e$  is the mass of an electron [16]. This equation uses MKS units. The canonical angular momentum is defined as

$$\mathbf{l} = \mathbf{r} \times \mathbf{p}, \quad (36)$$

where  $\mathbf{r}$  is the position and  $\mathbf{p}$  is the linear momentum.

The position,  $\mathbf{r}$ , is defined in three dimensions. In the case of the torus, the position  $\mathbf{r}$  will be referenced from the center of the torus as

$$\mathbf{r} = (x - X_c) \hat{\mathbf{x}} + (y - Y_c) \hat{\mathbf{y}} + (z - Z_c) \hat{\mathbf{z}}, \quad (37)$$

where  $X_c$ ,  $Y_c$ , and  $Z_c$  represent the center array index along the x-, y-, and z-directions.

The momentum operator in quantum mechanics,  $\mathbf{p}$ , is defined as

$$\mathbf{p} = -i\hbar\nabla = -i\hbar \left( \frac{\partial}{\partial x} \hat{\mathbf{x}} + \frac{\partial}{\partial y} \hat{\mathbf{y}} + \frac{\partial}{\partial z} \hat{\mathbf{z}} \right). \quad (38)$$

Eq. (37) and (38) can be combined to determine the angular momentum operator,  $\mathbf{l}$ , using Eq. (36). The result is

$$\mathbf{l} = [(x - X_c) \hat{\mathbf{x}} + (y - Y_c) \hat{\mathbf{y}} + (z - Z_c) \hat{\mathbf{z}}] \times \left[ -i\hbar \left( \frac{\partial}{\partial x} \hat{\mathbf{x}} + \frac{\partial}{\partial y} \hat{\mathbf{y}} + \frac{\partial}{\partial z} \hat{\mathbf{z}} \right) \right]. \quad (39)$$

Since an electron in a torus will move in the x-y plane, only the z components of the angular momentum will be significant. Eq. (39) can then be reduced to

$$\mathbf{l} = -i\hbar \left[ (x - X_c) \frac{\partial}{\partial y} \hat{\mathbf{z}} - (y - Y_c) \frac{\partial}{\partial x} \hat{\mathbf{z}} \right]. \quad (40)$$

To get the expectation value of the magnetic dipole moment, the operator must be integrated as shown:

$$\langle \mathbf{m} \rangle = \iiint_{-\infty}^{\infty} \Psi^* \cdot \mathbf{m} \cdot \Psi dx dy dz, \quad (41)$$

where  $\Psi^*$  is the complex conjugate of  $\Psi$ . Combining Eq. (35), (40), and (41) result in

$$\langle \mathbf{m} \rangle = \left[ -\frac{iq\hbar}{2m_e} \iiint_{-\infty}^{\infty} \Psi^* \cdot \left( (x - X_c) \frac{\partial \Psi}{\partial y} - (y - Y_c) \frac{\partial \Psi}{\partial x} \right) dx dy dz \right] \hat{\mathbf{z}} \quad (42)$$

to find the expectation value of the magnetic dipole moment.

#### *Implementing the Magnetic Dipole Moment Operator using the FDTD Method*

Eq. (42) can be implemented using the FDTD method and be rewritten as

$$\langle \mathbf{m} \rangle = -\frac{iq\hbar}{2m_e} \iiint_{-\infty}^{\infty} \Psi^* \cdot (der_Y - der_X) dx dy dz, \quad (43)$$

with

$$der_Y = (x - X_c) \frac{\partial \Psi}{\partial y}, \quad (44a)$$

$$der_X = (y - Y_c) \frac{\partial \Psi}{\partial x}. \quad (44b)$$

The direction of the magnetic dipole moment will be along the z-axis.

By dividing  $\Psi$  into real and imaginary components as in Eq. (3),  $der_Y$  and  $der_X$  can be approximated as

$$der_Y(i, j, k) = (x - X_C) \cdot \Delta x \left[ \begin{array}{c} \frac{\Psi_{real}(i, j + 1, k) - \Psi_{real}(i, j - 1, k)}{2 \cdot \Delta y} \\ +i \cdot \frac{\Psi_{imag}(i, j + 1, k) - \Psi_{imag}(i, j - 1, k)}{2 \cdot \Delta y} \end{array} \right], \quad (45a)$$

$$der_X(i, j, k) = (y - Y_C) \cdot \Delta y \left[ \begin{array}{c} \frac{\Psi_{real}(i + 1, j, k) - \Psi_{real}(i - 1, j, k)}{2 \cdot \Delta x} \\ +i \cdot \frac{\Psi_{imag}(i + 1, j, k) - \Psi_{imag}(i - 1, j, k)}{2 \cdot \Delta x} \end{array} \right]. \quad (45b)$$

As mentioned in Chapter 2,  $\Delta x$  and  $\Delta y$  are equal. The derivative was approximated using the two adjacent cells to provide symmetry.

Therefore, the expectation value of the magnetic dipole moment is given by

$$\langle \mathbf{m}(i, j, k) \rangle = -\frac{iq\hbar}{2m_e} \Psi^*(i, j, k) \cdot [der_Y(i, j, k) - der_X(i, j, k)]. \quad (46)$$

The integral must cover the total problem space, from the first cell to the total number of cells,  $X_{max}$ ,  $Y_{max}$ , and  $Z_{max}$ . This can be done by using a summation,

$$\langle \mathbf{m} \rangle = \sum_{k=1}^{Z_{max}} \sum_{j=1}^{Y_{max}} \sum_{i=1}^{X_{max}} \langle \mathbf{m}(i, j, k) \rangle, \quad (47)$$

with  $\langle \mathbf{m}(i, j, k) \rangle$  defined in Eq. (46).

### Verifying the Accuracy of the Magnetic Dipole Moment Calculation

The accuracy of the expectation value of the magnetic dipole moment, described in Eq. (47), can be verified by comparing it with the classical magnetic dipole moment given by

$$\mathbf{m} = I\pi \cdot r_{torus}^2 \hat{\mathbf{z}}, \quad (48)$$

where  $I$  is the current and  $r_{torus}$  is the radius of the torus [17].

In order to simulate the current in a loop, a wavepacket is used to indicate the position of an electron. This is initialized as shown in Fig. 14 at  $t = 0$  fs. The FDTD simulation is allowed to run, and the wavepacket will travel around the torus. The progression of the wavepacket in a complete loop is shown in Fig. 14.

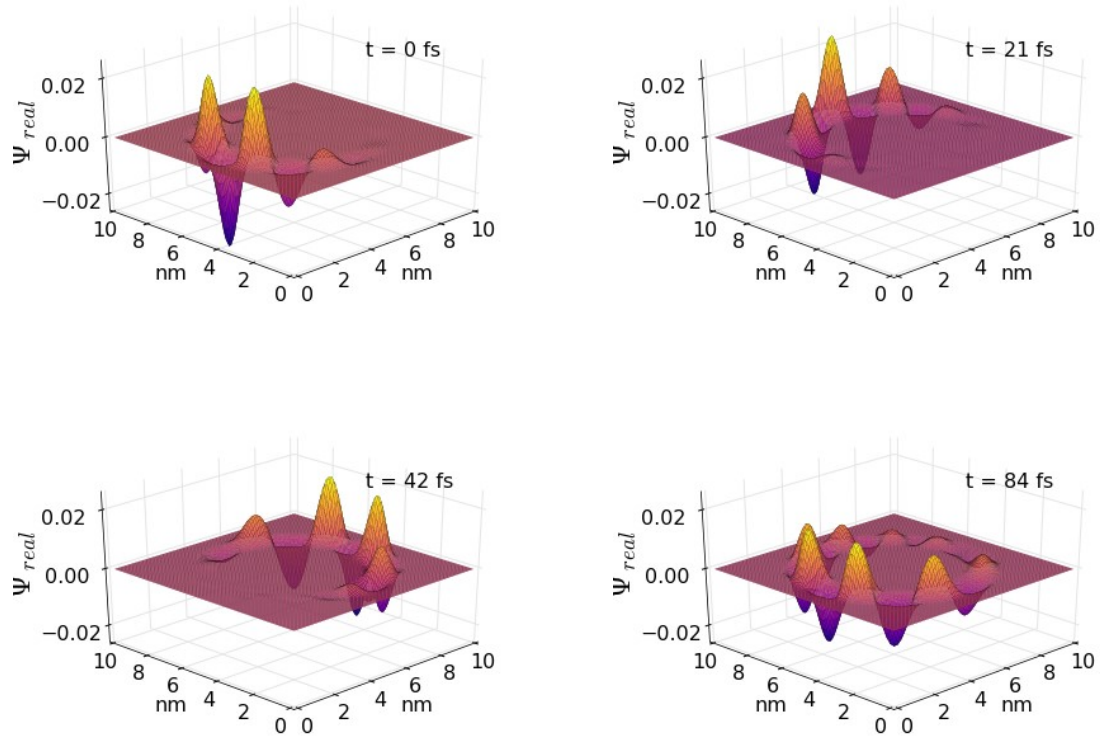


Figure 14. Progression of wave packet over time

From [9], the expectation value of the position is

$$\langle x \rangle = \int_{-\infty}^{\infty} \Psi^* x \Psi dx, \quad (49a)$$

$$\langle y \rangle = \int_{-\infty}^{\infty} \Psi^* y \Psi dy, \quad (49b)$$

where  $\Psi^*$  is the complex conjugate. Since the wavepacket is initiated to move around the torus, it has no initial momentum in the z-direction and only the x and y locations are monitored. The expectation value of the position over time is shown in Fig. 15. Because the wavelengths of the wavepacket have different momenta, the peaks and troughs of the wavelength do not reach the same values after a complete cycle. The amount of time taken to complete a cycle (84.0 fs--8,400 iterations) was measured from the initial trough of the expectation value of the y position to the next trough.

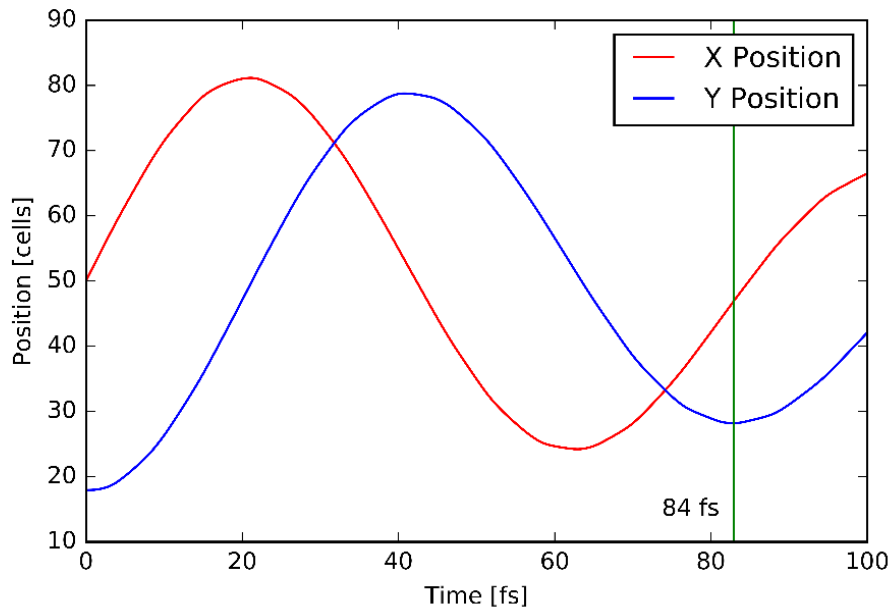


Figure 15. The expectation value of the position in the x- and y-directions over time

The wavepacket represents a single electron, so the charge of the wavepacket is approximately  $e = 1.6 \times 10^{-19} \text{ C}$ . The current may be determined from,

$$I = \frac{e}{\text{Period}} = \frac{1.6 \times 10^{-19} \text{ C}}{84.0 \times 10^{-15} \text{ s}} = 1.91 \times 10^{-6} \text{ A}. \quad (50)$$

Using Eq. (48) and the current, the magnetic dipole moment is

$$\mathbf{m} = (1.91 \times 10^{-6} \text{ A}) \cdot \pi \cdot (35 \text{ \AA})^2 = 0.0073 \text{ A} \cdot \text{\AA}^2, \quad (51)$$

where  $A$  represents amperes. Implementing Eq. (47) directly results in a magnetic dipole moment of  $0.0072 A \cdot \text{\AA}^2$ , which is close to the value calculated by using the period of a particle traveling around a torus. This implies that the FDTD implementation is accurate to within about one percent.

### Adding a Magnetic Field

In this section, a static magnetic field  $B_0$  is applied in the z-direction. It is perpendicular to the torus as seen in Fig. 16.

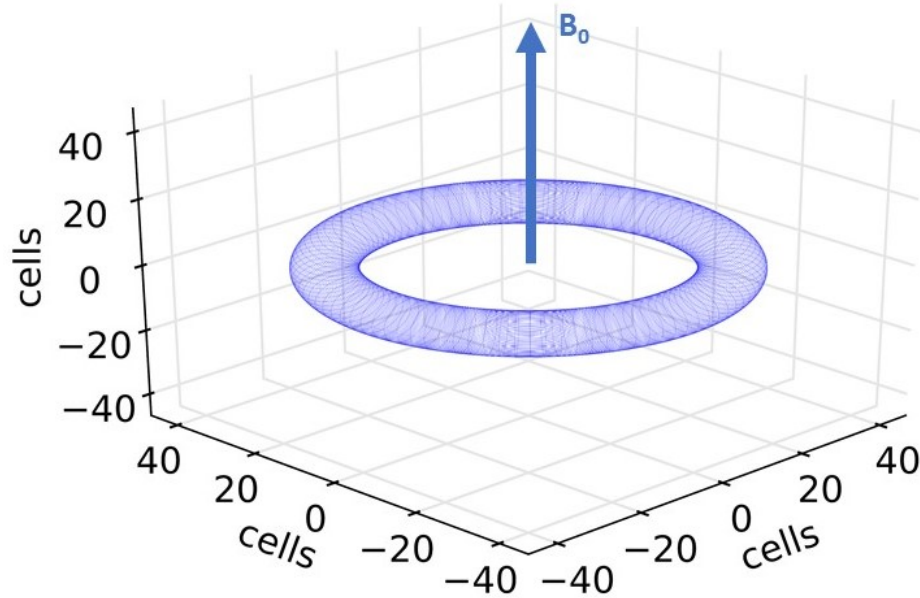


Figure 16. Direction of magnetic field applied to the torus

A magnetic field will complicate the Schrödinger equation shown in Eq. (1) since there are now additional forces acting on the electron. From [9], the Hamiltonian when a magnetic field is applied is given by

$$\begin{aligned}
 H &= \frac{1}{2m_e} \left( \frac{\hbar}{i} \nabla - q\mathbf{A} \right)^2 + V(x, y, z) \\
 &= -\frac{\hbar^2}{2m_e} \nabla^2 - \frac{\hbar q}{im_e} \nabla \cdot \mathbf{A} + \frac{q^2}{2m_e} \mathbf{A} \cdot \mathbf{A} + V(x, y, z)
 \end{aligned} \tag{52}$$

where  $q$  is the charge of an electron and  $\mathbf{A}$  is the vector potential. From [9] and [18],  $\mathbf{A}$  is defined as

$$\mathbf{A} = \frac{1}{2} [-(yB_z - zB_y)\hat{\mathbf{x}} + (xB_z - zB_x)\hat{\mathbf{y}} - (xB_y - yB_x)\hat{\mathbf{z}}], \quad (53)$$

and it is related to the magnetic field by

$$\mathbf{B} = \nabla \times \mathbf{A}. \quad (54)$$

The magnetic field is assumed to be uniform across the x-y plane since the torus structure is much smaller than the wavelength of the electromagnetic field. Since the magnetic field is applied only in the z-direction,  $B_z = B_0$  and Eq. (53) can be reduced to

$$\mathbf{A} = \frac{1}{2} [-(yB_0)\hat{\mathbf{x}} + (xB_0)\hat{\mathbf{y}}]. \quad (55)$$

Eq. (52) and Eq. (55) can be combined to result in

$$H = -\frac{\hbar^2}{2m_e} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) + i \frac{\hbar q B_0}{2m_e} \left( -y \frac{\partial}{\partial x} + x \frac{\partial}{\partial y} \right) + \frac{q^2 B_0^2}{8m_e} (x^2 + y^2) + V(x, y, z), \quad (56)$$

the Hamiltonian used with the magnetic field applied.

#### *FDTD with a Magnetic Field Applied*

The same process described in Chapter 2 can be used to implement the Hamiltonian of Eq. (56) in the FDTD method. The first and last term of the Hamiltonian is the same as that used in Chapter 2. The FDTD implementation including the two new terms is shown as a pair of coupled equations:

$$\begin{aligned}
\Psi_{real}^{n+1}(i, j, k) = & PML_{total}(i, j, k) \cdot \Psi_{real}^n(i, j, k) + \frac{\Delta t}{\hbar} V(i, j, k) \Psi_{imag}^{n+1/2}(i, j, k) \\
& - \frac{\hbar \Delta t}{2m_e(\Delta x)^2} \left[ \begin{aligned} & -6\Psi_{imag}^{n+1/2}(i, j, k) + \Psi_{imag}^{n+1/2}(i+1, j, k) + \Psi_{imag}^{n+1/2}(i-1, j, k) \\ & + \Psi_{imag}^{n+1/2}(i, j+1, k) + \Psi_{imag}^{n+1/2}(i, j-1, k) \\ & + \Psi_{imag}^{n+1/2}(i, j, k+1) + \Psi_{imag}^{n+1/2}(i, j, k-1) \end{aligned} \right], \quad (57a) \\
& + \frac{(\Delta t)(\Delta x)^2 q^2 B_0^2}{8m_e \hbar} [(i - X_C)^2 + (j - Y_C)^2] \Psi_{imag}^{n+1/2}(i, j, k) \\
& + \frac{(\Delta t)qB_0}{4m_e} \left[ \begin{aligned} & -(j - Y_C) \cdot [\Psi_{real}^n(i+1, j, k) - \Psi_{real}^n(i-1, j, k)] \\ & +(i - X_C) \cdot [\Psi_{real}^n(i, j+1, k) - \Psi_{real}^n(i, j-1, k)] \end{aligned} \right]
\end{aligned}$$

$$\begin{aligned}
\Psi_{imag}^{n+3/2}(i, j, k) = & PML_{total}(i, j, k) \cdot \Psi_{imag}^{n+1/2}(i, j, k) - \frac{\Delta t}{\hbar} V(i, j, k) \Psi_{real}^{n+1}(i, j, k) \\
& + \frac{\hbar \Delta t}{2m_e(\Delta x)^2} \left[ \begin{aligned} & -6\Psi_{real}^{n+1}(i, j, k) + \Psi_{real}^{n+1}(i+1, j, k) + \Psi_{real}^{n+1}(i-1, j, k) \\ & + \Psi_{real}^{n+1}(i, j+1, k) + \Psi_{real}^{n+1}(i, j-1, k) \\ & + \Psi_{real}^{n+1}(i, j, k+1) + \Psi_{real}^{n+1}(i, j, k-1) \end{aligned} \right] \quad (57b) \\
& - \frac{(\Delta t)(\Delta x)^2 q^2 B_0^2}{8m_e \hbar} [(i - X_C)^2 + (j - Y_C)^2] \Psi_{real}^{n+1}(i, j, k) \\
& + \frac{(\Delta t)qB_0}{4m_e} \left[ \begin{aligned} & -(j - Y_C) \cdot \Psi_{imag}^{n+1/2}(i+1, j, k) - \Psi_{imag}^{n+1/2}(i-1, j, k) \\ & +(i - X_C) \cdot \Psi_{imag}^{n+1/2}(i, j+1, k) - \Psi_{imag}^{n+1/2}(i, j-1, k) \end{aligned} \right]
\end{aligned}$$

Since  $x$  and  $y$  are measured from the center of the torus,  $x$  is replaced with  $(i - X_C)$  and  $y$  is replaced with  $(j - Y_C)$ , with  $X_C$  and  $Y_C$  representing the midpoint along the x-axis and y-axis, respectively.

Since the new set of equations contains two additional terms, the stability conditions will expand from Eq. (11a) and (11b) to also include

$$\frac{(\Delta t)qB_0}{4m_e} (i - X_C) \leq 0.15, \quad (58a)$$

$$\frac{(\Delta t)(\Delta x)^2 q^2 B_0^2}{8m_e \hbar} [(i - X_C)^2 + (j - Y_C)^2] \leq 0.15. \quad (58b)$$



For the simulation described in this chapter, these values are small and are not a concern. Since the torus is symmetric about the center along the z-axis,  $(i - X_C)$  and  $(j - Y_C)$  will have the same maximum value.

#### *Magnetic Dipole Moment Operator with a Magnetic Field*

The magnetic dipole moment operator previously described in Eq. (35) uses the canonical momentum. When a magnetic field is applied, the effects due to the vector potential must be taken into account. This is accomplished by calculating the magnetic dipole moment using the mechanical momentum. A new term is added to the magnetic dipole moment operator from Eq. (35). From [19], the new operator is

$$\mathbf{m} = \frac{q}{2m_e} \hat{\mathbf{l}} - \frac{q^2}{2m_e} (\mathbf{r} \times \mathbf{A}). \quad (59)$$

The first term in the equation was found in Eq. (35). The second term must be expanded and uses Eq. (37) and (55), to obtain

$$\mathbf{m} = -i \frac{q\hbar}{2m_e} \left[ (x - X_C) \frac{\partial}{\partial y} - (y - Y_C) \frac{\partial}{\partial x} \right] - \frac{q^2 B(t)}{4m_e} [(x - X_C)^2 + (y - Y_C)^2]. \quad (60)$$

This is entirely in the z-direction. Again,  $x$  and  $y$  are measured from the center of the torus. This can be implemented using the FDTD method in the same manner described earlier in this chapter.

#### **Verifying the Accuracy of the Magnetic Dipole Moment Calculation with a Static Magnetic Field**

The accuracy of the new magnetic dipole operator from Eq. (60) must be verified. As was done without a magnetic field, a wavepacket is initialized within the torus and allowed to run. This is shown in Fig. 17 for a complete loop around the torus with a 25 T static magnetic field applied.

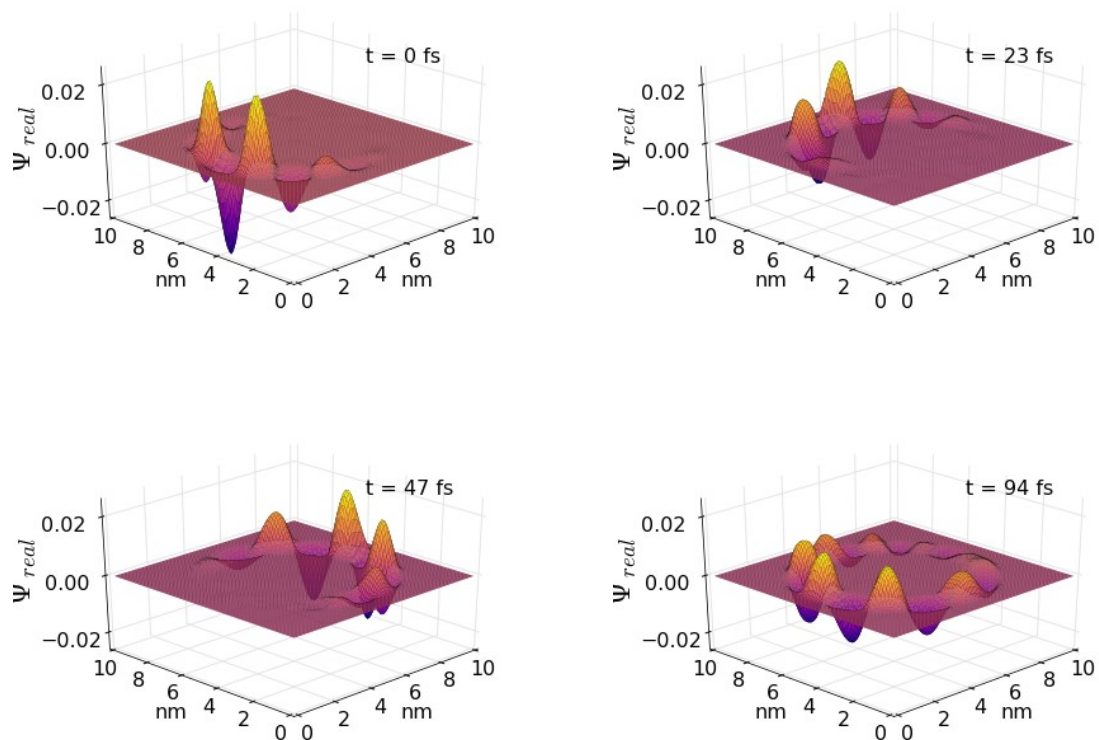


Figure 17. Progression of wave packet over time with a static 25 T magnetic field

The expectation value of the position is again monitored as shown in Fig. 18.

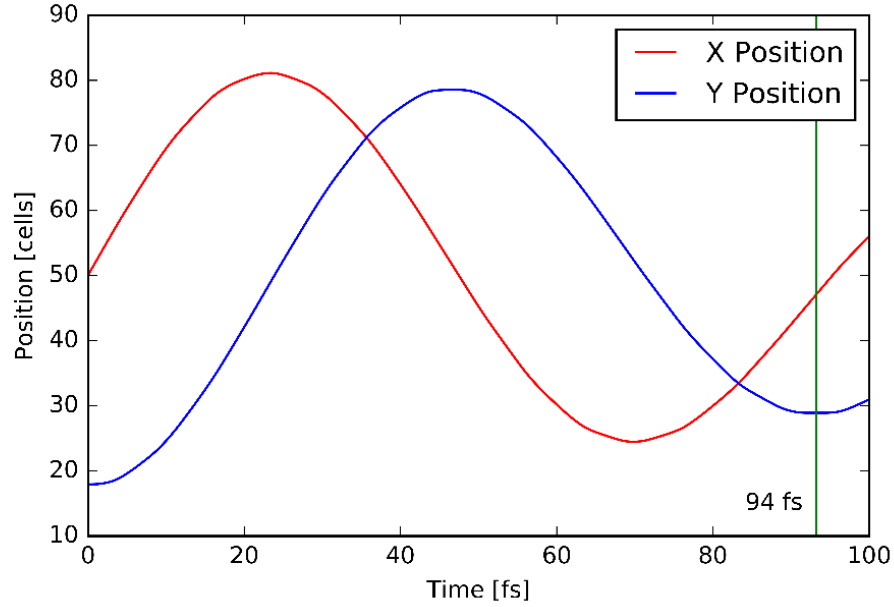


Figure 18. The expectation value of the position in the x- and y-directions over time with a static 25 T magnetic field

The classical magnetic dipole moment can be found using Eq. (48). The result of using Eq. (48) along with using the quantum operator is shown in Table 2. The table includes several magnetic field strengths. The classical and quantum approaches give values that are the same within about three percent, verifying the magnetic dipole moment operator and the magnetic field implementation using the FDTD method.

Table 2. Classical and quantum magnetic dipole moment with a static magnetic field

Magnetic Field [T]	Classical $m$ [ $A \cdot \text{\AA}^2$ ]	Operator $m$ [ $A \cdot \text{\AA}^2$ ]	Difference
-50	0.0078	0.0076	2.6%
-25	0.0075	0.0074	1.3%
0	0.0073	0.0072	1.4 %
25	0.0071	0.0070	1.4%
50	0.0069	0.0068	1.5%

## Chapter 5: Simulation of a Time-Harmonic Magnetic Field

The FDTD method provides a direct way to implement and use the Schrödinger equation to simulate a magnetic field while measuring the magnetic dipole moment. This chapter describes those simulation results and how they can be used to find magnetic susceptibility. It also discusses possibilities for future work and considerations when using the FDTD method.

### Effects of a Time-Harmonic Field on a Particle in the Torus

The FDTD simulation can be run with a magnetic field applied as described, using the ground eigenstate for the initial  $\Psi_{real}$  and  $\Psi_{imag}$ . For this chapter, the applied magnetic field is a sinusoidal function,

$$\mathbf{B}(t) = B_{max} \sin(2\pi f_0 t) \hat{\mathbf{z}}, \quad (61)$$

with  $f_0$  representing the applied frequency and  $t$  the time. This will be multiplied by a Hanning window [Eq. (15)]. The Hanning window is used to reduce the effects of the abrupt addition of a magnetic field. The effects of the magnetic field may be monitored by observing the magnetic dipole moment using the operator in Eq. (60).

Fig. 19(a) shows the applied 30 T, 20 THz magnetic field multiplied by a Hanning window. Fig. 19(b) shows the magnetic dipole moment over time. The magnetic dipole moment oscillates at the same frequency as the applied magnetic field. Fig. 19(c) and Fig. 19(d) show the Fourier transforms of the magnetic field and the magnetic dipole moment, respectively. The Hanning window affects the magnitude of the values generated with the Fourier transforms, so the values in Fig. 19(c) and Fig. 19(d) are multiplied by two. Then the peak for the magnetic field is 30 T in Fig. 19(c), as expected.

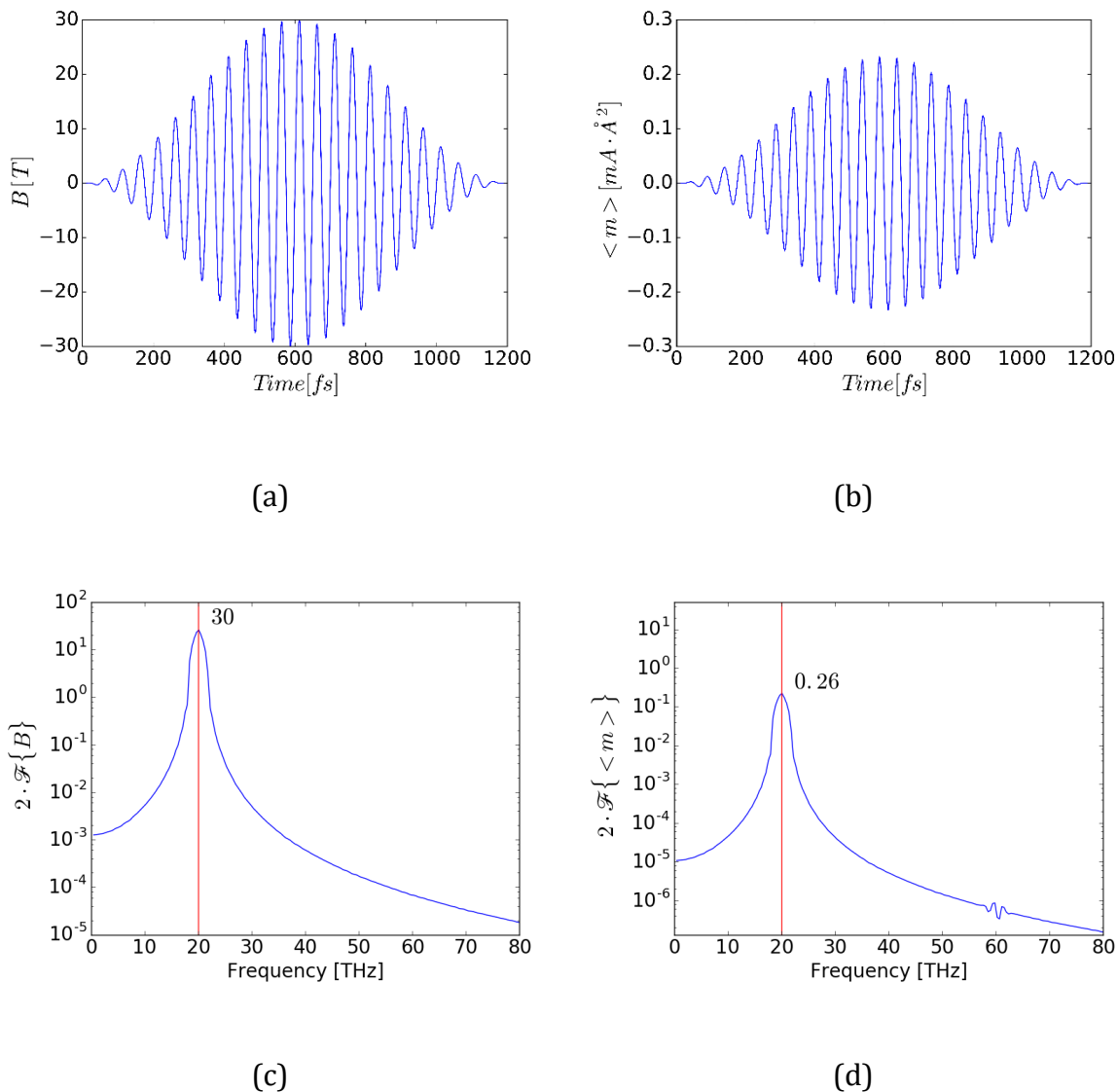


Figure 19. The results of a simulation of a particle in a torus under the influence of a sinusoidal magnetic field with a maximum magnitude of 30 T and a frequency of 20 THz. (a) The magnetic field, (b) the magnetic dipole moment, (c) the Fourier transform of the magnetic field, and (d) the Fourier transform of the dipole moment.

The peak values of the magnetic dipole moment generated with the Fourier transform are determined at  $f_0$ . These are shown to have a linear relationship with the applied magnetic field strength, as shown in Fig. 20.

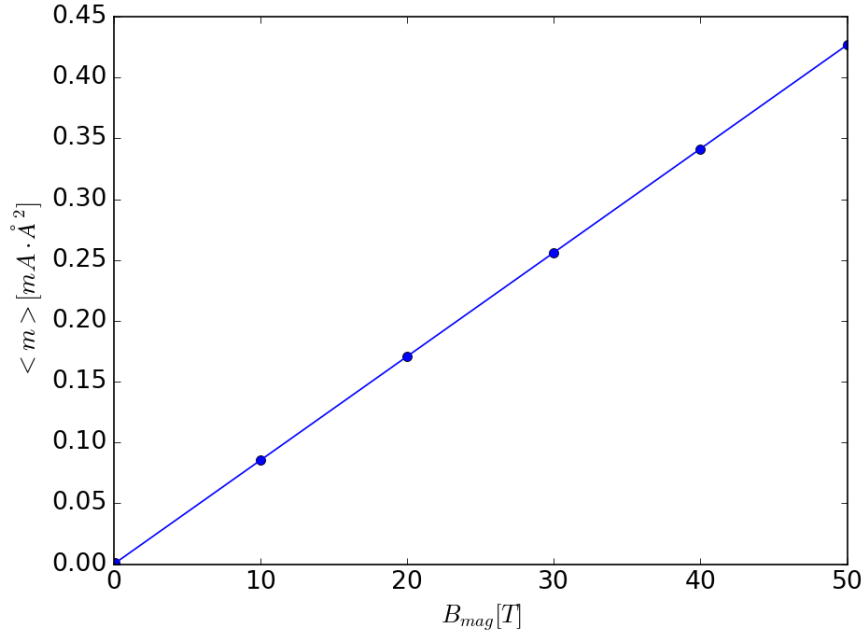


Figure 20. The magnetic dipole moment versus the magnetic field strength for a magnetic field of 20 THz

#### *Magnetic Dipole Moment without a Magnetic Field Applied*

The results of these simulations will have inaccuracies because of the finite number of cells used. The magnetic dipole moment without a magnetic field can be used to find a base level of uncertainty in the calculations and also to examine the accuracy of the ground eigenstate generated. The magnetic dipole moment should ideally be zero for the ground eigenstate of a torus since there is no angular momentum;  $\Psi$  will simply oscillate in a steady manner with the period described in Eq. (33).

For the torus and ground eigenstate used in these simulations, the maximum magnetic dipole without a magnetic field applied is approximately  $2 \times 10^{-15} mA \cdot \text{\AA}^2$ . This value represents an intrinsic magnetic dipole moment due to simulation inaccuracies. The magnetic dipole moment results will be distorted if the magnetic dipole moment caused by the applied magnetic field is on the same scale as the intrinsic magnetic dipole

moment. The intrinsic magnetic dipole moment is not a concern for the results in this chapter.

### Magnetic Susceptibility

The FDTD method described makes it possible to determine the magnetic susceptibility from the magnetic dipole moment. The magnetic dipole moment generated using the Fourier transform, as in Fig. 19(d), can be used to find the magnetic susceptibility.

From [1], the bulk polarizability is given by

$$P(t) = \chi^{(1)}E(t) + \chi^{(2)}E^2(t) + \chi^{(3)}E^3(t) + \dots, \quad (62)$$

where  $\chi^{(1)}$  is the linear susceptibility,  $\chi^{(2)}$  is the second-order nonlinear optical susceptibility, and  $\chi^{(3)}$  is the third-order nonlinear optical susceptibility. This response is in the z-direction only, which is the direction of the applied magnetic field. The bulk polarizability is related to the dipole moment of a molecule by

$$P(t) = Np(t), \quad (63)$$

where  $N$  is the number density and many conditions are met [1]. The dipole moment equation may then be written as

$$p(t) = \alpha E(t) + \beta E^2(t) + \gamma E^3(t) + \dots. \quad (64)$$

An analogous equation for the magnetic dipole moment  $m$  using the magnetic field  $B(t)$  results in

$$m(t) = \alpha B(t) + \beta B^2(t) + \gamma B^3(t) + \dots. \quad (65)$$

The following calculations will focus on  $\alpha$  (the polarizability),  $\beta$  (the hyperpolarizability), and  $\gamma$  (the second hyperpolarizability) [3]. The response is needed in the frequency domain so the magnetic dipole moment is written as

$$m(\omega) = \alpha B(\omega) + \beta B^2(\omega) + \gamma B^3(\omega). \quad (66)$$

A Fourier transform is performed to obtain the magnetic dipole moment in the frequency domain.

The polarizability will occur at the fundamental frequency,  $\omega_0$ . The time varying magnetic field was previously described in Eq. (61). This means the hyperpolarizability will have a magnetic field component

$$B^2(t) = \frac{1}{2}(B_{max}^2 - B_{max}^2 \cos(2\omega_0 t)), \quad (67)$$

which has a frequency component at 0 and at  $2\omega_0$  [2].

Similarly, the second hyperpolarizability will have a magnetic component

$$B^3(t) = \frac{1}{4}(3B_{max}^3 \sin(\omega_0 t) - B_{max}^3 \sin(3\omega_0 t)). \quad (68)$$

This means the second hyperpolarizability will have frequency components at  $\omega_0$  and  $3\omega_0$ .

### *Grated Torus*

The Fourier transform of the torus in Fig. 19(d) showed only a small component at  $3f_0$ . A grated torus, one in which a periodic potential is added, is used in an attempt to find a high nonlinear response. Any structure may be examined using the FDTD method simply by specifying the potential  $V(x, y, z)$ .

The grated torus is shown in Fig. 21. The torus shown has 20 peaks, and the difference between the peaks and the troughs is 0.6 eV.



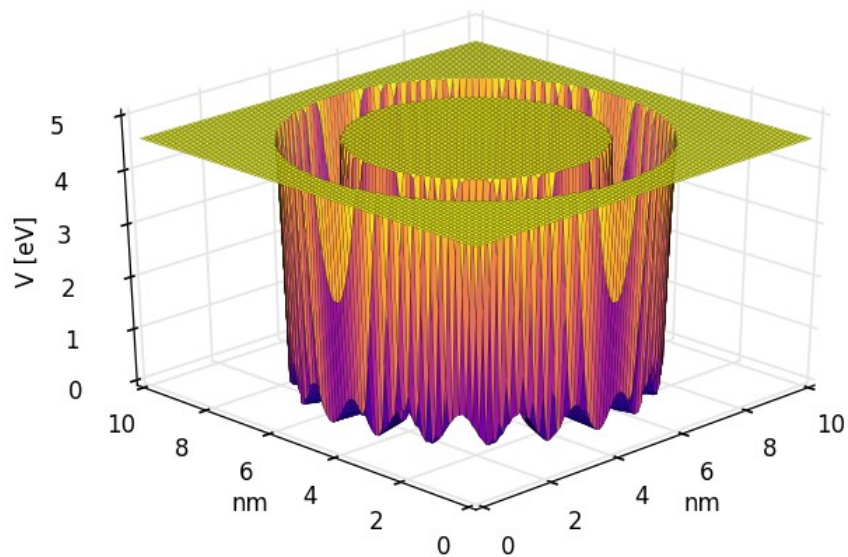


Figure 21. Potential showing a grated torus; cross section shown at  $z = Z_c$

The ground eigenstate of the grated potential is shown in Fig. 22. The irregularities affect the shape of the ground state, as expected.

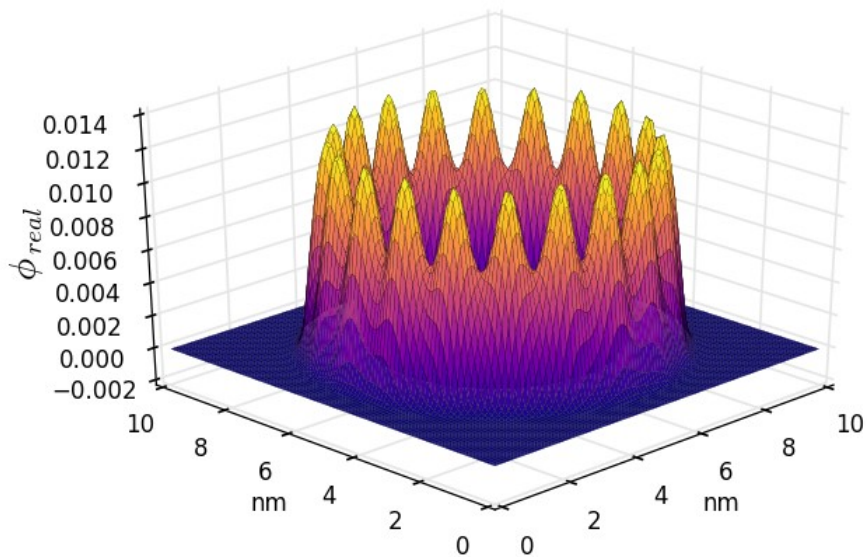


Figure 22. Ground eigenstate for the grated potential; cross section shown at  $z = Z_c$

### *Magnetic Susceptibility Results for a Grated Torus*

In this section the results of applying a sinusoidal magnetic field to a particle in the grated torus are described. The sinusoidal magnetic field is applied so the frequency components of the grated torus can be calculated. The magnetic field is described by Eq. (61) where  $B_{max} = 30 T$  and  $f_0 = 20 THz$ . The results are illustrated in Fig. 23. Fig. 23(a) shows the magnetic field and Fig. 23(b) shows the resulting magnetic dipole moment. Their respective Fourier transforms are shown in Fig. 23(c) and Fig. 23(d). The red vertical lines show  $f_0$ . The majority of the signal for the magnetic dipole moment is at  $f_0$ , but there is also a significant component at  $3f_0$ , as shown in Fig. 23(d). This corresponds with the second hyperpolarizability. This component is much more prominent than that seen in the plain torus.

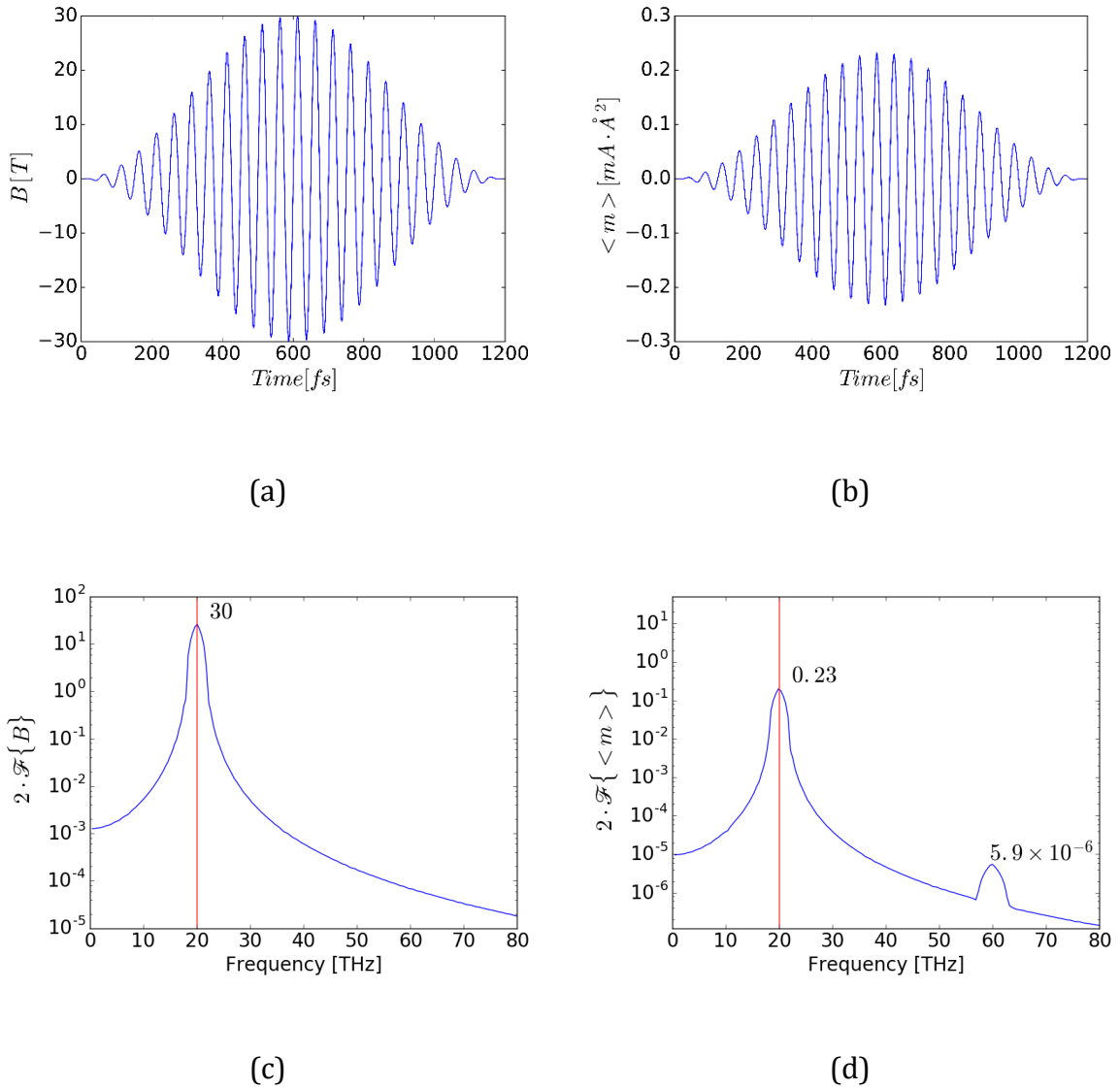


Figure 23. The results of a simulation of a particle in a grated torus under the influence of a sinusoidal magnetic field with a maximum magnitude of 30 T and a frequency of 20 THz. (a) The magnetic field, (b) the magnetic dipole moment, (c) the Fourier transform of the magnetic field, and (d) the Fourier transform of the magnetic dipole moment.

Fig. 24 shows the component of the magnetic dipole moment determined at 20 THz,  $f_0$ , across several magnetic field strengths. Because the hyperpolarizability has a cubic component at  $f_0$ , as shown in Eq. (68), a best fit line is created using a linear and a cubic response to match the calculated values. The linear coefficient,  $\alpha$ , is dominant and gives

the polarizability. The cubic coefficient,  $\gamma$ , is a component of the second hyperpolarizability. These coefficients are used to create the solid line on the graph.

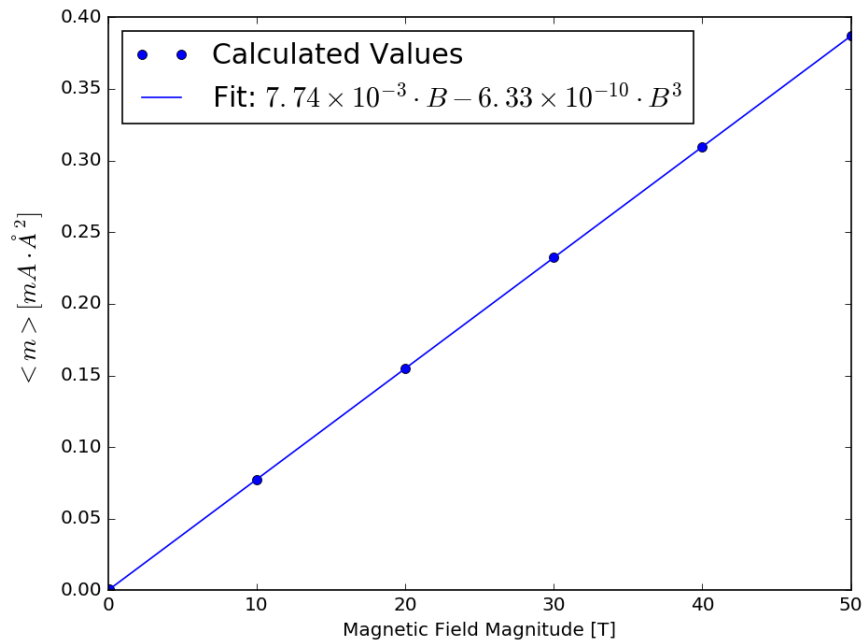


Figure 24. The magnetic dipole moment determined at  $f_0$  versus magnetic field strength for a particle in a grated torus.  $\alpha = 7.74 \times 10^{-3}$  and  $\gamma = -6.33 \times 10^{-10}$

Similarly, the magnetic dipole moment from the Fourier transform can be determined at  $2f_0$ , which is the frequency of the second harmonic. If there is significant hyperpolarizability, the dipole moment plotted across magnetic field magnitudes will show a quadratic response. The coefficient of the quadratic response is the hyperpolarizability. In the case of both the plain and grated torus, this component is not present in any significant amount.

Fig. 25 shows the component of the magnetic dipole moment at  $3f_0$ . The coefficient of the cubic response is a component of  $\gamma$ , the second hyperpolarizability. The cubic fit line is also shown in Fig. 25. The ratio of  $\gamma$  determined at  $3f_0$  to  $\gamma$  determined at  $f_0$  is approximately -0.35, which is close to the ideal value of -1/3. This ratio can be seen in Eq. (68).

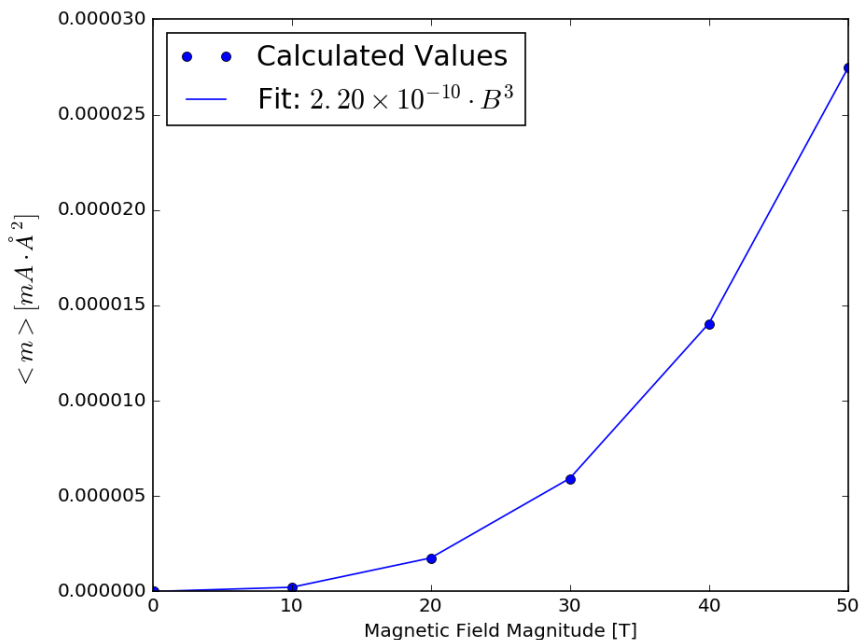


Figure 25. The magnetic dipole moment determined at  $3f_0$  versus magnetic field strength.  $\gamma = 2.20 \times 10^{-10}$

### Discussion

One of the strengths of the FDTD method is that any structure can be simulated, subject only to the resolution of the cell size and the dimensions of the problem space. This allows different nanostructures to be simulated, tested, and modified to obtain optimal nonlinear properties.

It is possible to increase the resolution of a structure by increasing the number of cells used in the problem space. Increasing the resolution allows the discrete cells to be less of a problem and the artificial dipole moment effects to be smaller, even if longer simulations are necessary. The resolution and the types of structures that can be evaluated are dependent on computational power and an acceptable level of accuracy.

It is also important to monitor the shape of  $\Psi_{real}$  and  $\Psi_{imag}$  to ensure they are not distorted. This may indicate perturbation theory is breaking down.  $\Psi_{real}$  and  $\Psi_{imag}$  must remain normalized throughout a simulation just as the initial ground state is normalized as in Eq. (24). As mentioned, the conditions for stability in Eq. (11a) and (11b) and in Eq. (58a) and (58b) must be carefully monitored. This also affects the selection of various parameters. The size of the structure and the time step size may limit the magnetic field strength, for example.

## Chapter 6: Summary

### Summary

The FDTD method, which has been used to implement the Schrödinger equation and find the eigenstates of a wide array of structures, was discussed for a three-dimensional simulation. The primary structure chosen for this work was a torus; this shape behaved in the expected manner when the revival time was simulated. For the sake of efficiency, a test function was used to generate the ground eigenstate as opposed to a test pulse.

The magnetic dipole moment operator was described, and the implementation for measuring the expectation value of the magnetic dipole moment was detailed. The method was verified by using a classical model of an electron moving in a torus in a given time.

This work also used the FDTD method to simulate the Schrödinger equation with an applied magnetic field. The Hamiltonian used to describe the system and the magnetic dipole moment operator were derived. This was again verified by using a classical model of an electron moving in the torus.

The expectation value of the magnetic dipole moment was found with an applied time-varying magnetic field to calculate the magnetic susceptibility. The response of a torus was found to be linear, with no significant frequency components aside from that of the applied frequency.

Since the plain torus did not show many nonlinear characteristics, a grating was added to the torus to obtain second hyperpolarizability characteristics. The magnetic dipole moment at the third harmonic displays a cubic response to magnetic field strength, as expected.

Tradeoffs and considerations were also discussed. For simulation convergence, parameters such as time step size, magnetic field strength, and structure size must be carefully chosen.

### **Conclusions for Implementation in Python**

The implementation of the FDTD formulation of the Schrödinger equation into Python is straight-forward and similar to using MATLAB, the software used in [9]. The Python library NumPy makes for clear and concise programs, eliminating most loops.

However, simulations of the 100x100x30 cell problem space were extremely time-consuming using NumPy functions. When running a hundred thousand iterations, the simulation would take 1.5-2 hours to complete running on a laptop computer.

Python offers a way to speed up certain operations by using the Numba library. This allows the performance of segments of code to be comparable to C and Fortran while keeping a program written in Python. With some minor modifications, the main FDTD process can be implemented efficiently using Numba to reduce the 1.5-2 hour simulation time to about 10 minutes on the laptop. The laptop used in this work has a 2.2 GHz processor and 8 GB of RAM.

Because NumPy has no optimized function for the main FDTD loops, the final program implemented those loops in Numba using simple “for” loops operating on NumPy arrays. All other NumPy calculations operated quickly enough that optimization with Numba was not necessary. This strategy of using high-level library functions and optimizing time-consuming code with Numba was effective in implementing the FDTD method.



## References

- [1] R. W. Boyd, *Nonlinear Optics, Second Edition*, San Diego, CA: Academic Press, 2003.
- [2] M. G. Kuzyk, *Lecture Notes in Nonlinear Optics: A Student's Perspective*, Pullman, WA: NLOsource.com, 2014.
- [3] J. Pérez-Moreno, S. Shafei, and M. G. Kuzyk, "Sum Rules and Scaling in Nonlinear Optics," *Physics Reports*, vol. 524, no. 4, pp. 397-398, 2013.
- [4] M. Firczuk, M. Winiarska, A. Szokalska, M. Jodlowska, M. Swiech, K. Bojarczuk, P. Salwa, and D. Nowis, "Approaches to improve photodynamic therapy of cancer," *Front Biosci*, Landmark 16, pp. 208-224, 2011.
- [5] I. Roy, T. Y. Ohulchanskyy, H. E. Puda, J. Bergey, A. R. Oseroff, J. Morgan, T. J. Dougherty, and P. N. Prasad, "Ceramic-Based Nanoparticles Entrapping Water-Insoluble Photosensitizing Anticancer Drugs: A Novel Drug-Carrier System for Photodynamic Therapy," *J. Am. Chem. Soc.*, vol. 125, no. 26, p. 7860-7865, 2003.
- [6] D. M. Sullivan and D. S. Citrin, "Time-domain simulation of quantum spin," *J. Appl. Phys.*, vol. 94, no. 10, pp. 6518-6524, 2003.
- [7] D. Sullivan, S. Mossman, and M. G. Kuzyk, "Hybrid quantum systems for enhanced nonlinear optical susceptibilities," 21 July 2016. [Online]. Available: <https://arxiv.org/pdf/1607.06431.pdf>. [Accessed 25 March 2017].
- [8] D. M. Sullivan, *Electromagnetic Simulation Using the FDTD Method*, Second Edition, New Jersey: IEEE Press, 2015.
- [9] D. M. Sullivan, *Quantum Mechanics for Electrical Engineers*, New York: IEEE Press, 2012.
- [10] D. M. Sullivan and D. S. Citrin, "Determining quantum eigenfunctions in three-dimensional nanoscale structures," *J. Appl. Phys.*, vol. 97, no. 104305, pp. 1-6, 2005.
- [11] D. M. Sullivan and D. S. Citrin, "Determination of the eigenfunctions of arbitrary nanostructures using time domain simulation," *J. Appl. Phys.*, vol. 91, no. 5, pp. 3219-3226, 2002.
- [12] J. P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, vol. 114, no. 2, pp. 185-200, 1994.

- [13] C. Zheng, "A perfectly matched layer approach to the nonlinear Schrödinger wave equations," *J. Compt. Phys.*, vol. 227, no. 1, pp. 537-556, 2007.
- [14] D. M. Sullivan and P. M. Wilson, "Time-domain determination of transmission in quantum nanostructures," *J. Appl. Phys.*, vol. 112, no. 064325, pp. 1-7, 2012.
- [15] D. M. Sullivan, S. Mossman, and M. G. Kuzyk, "Time-Domain Simulation of Three Dimensional Quantum Wires," 28 April 2016. [Online]. Available: <http://journals.plos.org/>, DOI:10.1371/journal.pone.0153802. [Accessed 26 February 2017].
- [16] D. J. Griffiths, *Introduction to Quantum Mechanics*, Upper Saddle River, NJ: Prentice Hall, 1995.
- [17] D. K. Cheng, *Field and Wave Electromagnetics*, Reading, MA: Addison-Wesley Publishing, 1989.
- [18] C. Cohen-Tannoudji, B. Diu, and F. Lohoe, *Quantum Mechanics*, New York: Wiley, 1977.
- [19] A. L. Fetter and J. D. Walecka, *Theoretical Mechanics of Particles and Continua*, Mineola, NY: Dover Publishing, 2003.