3D Visualization of Light Scattered by a Free Electron

in a Relativistic Laser Focus

Jacob Barker

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Bachelor of Science

Justin Peatross, Advisor

Department of Physics and Astronomy
Brigham Young University

Copyright © 2018 Jacob Barker
All Rights Reserved
ABSTRACT

3D Visualization of Light Scattered by a Free Electron in a Relativistic Laser Focus

Jacob Barker
Department of Physics and Astronomy, BYU
Bachelor of Science

A focused intense laser pulse can rip multiple electrons from atoms. At low density, the liberated electrons move freely in the laser field in response to the Lorentz force. The movement of a charged particle causes electromagnetic radiation to scatter out of the focus. In our lab, we study these laser-particle interactions by looking at the scattered electromagnetic radiation. As the laser intensity increases, electrons can reach relativistic speeds as they oscillate. Few people have intuition for this complex motion and radiation. I developed a computer simulation that animates these interactions in a physically realistic manner. I discuss the physics of these interaction and the development of the simulation. I present several animations of scenarios similar to our experiments.

Keywords: Laser, Particle, Gaussian Beam, Relativistic Electron, Animation
ACKNOWLEDGMENTS

First and foremost, I would like to thank my parents, Mike and Michelle. They were the ones who got me excited about learning and have always supported me and encouraged me to work hard. I would also like to thank my brother, Robbie. Those who have met both of us agree we couldn’t be more different people, but that is what makes us so close. I also would not have been able to complete this without my research adviser, Justin Peatross. He got me excited about research and showed me just how fun and frustrating it can be. I also have to mention my grandpa Rodger. He is the only member of my extended family with a technical background. His advise has helped me navigate the technical world. Last I want to thank my best friend, Isaiah Young. We have been best friends since 9th grade and I can’t wait to see where post-graduation takes both of us.
# Contents

Table of Contents iv

List of Figures vi

1 Introduction 1
   1.1 Advancements in Laser Technology towards Higher Intensities ............... 1
   1.2 Relativistic Regime for Laser-Particle Interactions ............................ 3
   1.3 Electron-Laser Experiments .............................................................. 3
   1.4 The Challenge of Visualizing Interactions .......................................... 4
   1.5 Overview of the Structure of the Thesis ........................................... 5

2 Physical Equations Governing Electron-Laser Interactions 6
   2.1 Singh Model of Laser Field ................................................................. 6
   2.2 Relativistic Equations of Motion for an Electron .................................. 8
   2.3 Electron Trajectories .......................................................................... 8
   2.4 Radiated Fields .................................................................................... 10

3 Visualization Methods 13
   3.1 Virtual Camera ...................................................................................... 13
   3.2 Displaying Laser and Electron Fields .................................................. 14
   3.3 Gridlines ............................................................................................ 15
   3.4 Test Case ............................................................................................ 16
   3.5 Movies of Electron-Laser ..................................................................... 17

Appendix A MATLAB Code 20
   A.1 Moving Electron .................................................................................. 20
   A.2 movieviewer ....................................................................................... 21
   A.3 electronPath ....................................................................................... 24
   A.4 rhs .................................................................................................... 25
   A.5 EBfield ............................................................................................... 26
   A.6 calcMaxs ............................................................................................. 26
   A.7 Rotate ................................................................................................. 28
A.8 calculateWave ................................................................. 28
A.9 calculateS ................................................................. 29
A.10 calculateElectron ....................................................... 30
A.11 calculateSphere ......................................................... 31
A.12 calculateWisp ............................................................ 31
A.13 calculateWispField ..................................................... 32
A.14 fastcross ................................................................. 33
A.15 calculateGrid ............................................................ 33
A.16 calculateGirdLine ....................................................... 34

Bibliography ........................................................................... 35

Index .................................................................................. 37
List of Figures

1.1 Maximum Intensity History ........................................... 2
1.2 Experimental Set Up .................................................. 4
1.3 Dipole Radiation ....................................................... 5

2.1 2D Singh Model ......................................................... 7
2.2 Low Intensity Electron Trajectory ................................. 9
2.3 High Intensity Electron Trajectory .............................. 9
2.4 Vector Definitions .................................................... 11

3.1 Virtual Camera ......................................................... 14
3.2 Gridline Comparison .................................................. 16
3.3 Dipole Comparison .................................................... 16
3.4 Electron Starting at Rest ............................................ 17
3.5 Electron Starting with Small Initial Momentum Upstream .... 18
3.6 Electron Starting with Relativistic Initial Momentum Upstream 18
Chapter 1

Introduction

1.1 Advancements in Laser Technology towards Higher Intensities

Laser intensities have steadily increased since the first laser was built in 1960 by Theodore H. Maiman [1] while he was working at Hughes Research Laboratories. His laser used a ruby crystal to produce a pulsed beam with a wavelength of 694 nm and was based on the theoretical work of Charles Townes and Arthur Schawlow [1]. This laser was only able to reach intensities below $10^{10}$ W·cm$^{-2}$. Since then, the maximum intensity of a laser has increased significantly. Currently, the laser with the largest intensity is the HERCULES Petawatt laser at the University of Michigan. It is able to produce an intensity of $2 \times 10^{22}$ W·cm$^{-2}$. The Guinness Book of World Records notes that this is "equivalent to focusing all the sunlight that hits the Earth onto a single grain of sand." [2] The laser used in our research group is able to reach intensities of $10^{18}$ W·cm$^{-2}$.

All lasers can be classified into two different types: continuous lasers and pulsed lasers. Most people are only familiar with a continuous laser because there are so many common applications; laser pointers, CD/DVD readers, laser tag, and laser light shows are all examples of continuous
lasers. However, for many scientific applications, a pulsed laser is used because it can reach much higher intensities than a continuous laser. A pulsed laser accumulates light energy in a cavity, then releases it periodically. This allows higher intensities to be achieved without requiring additional energy.

In the past 50 years there has been much work done to increase the peak intensity of pulsed lasers. The evolution in time can be seen in Figure 1.1 [3]. Q-switching, mode-locking, and chirped pulse amplification are important innovations that have allowed for an increase in the intensity of a pulsed laser [4]. The importance of this work has recently been acknowledged by the 2018 Nobel prize in physics [5]. Much of the increase in the laser intensity has come through shortening the pulse in time.

![Figure 1.1 The progression of maximum intensity achieved by a laser.][3]
1.2 Relativistic Regime for Laser-Particle Interactions

The first lasers called only for a non-relativistic treatment of a laser-particle interaction. The Lorenz force law tells us that the force exerted on the particle follows the relationship:

\[ \vec{F} = q_e(\vec{E} + \vec{v} \times \vec{B}) \]  

(1.1)

For the first lasers, the strength of the magnetic field can be ignored since \(|\vec{B}| \approx |\vec{E}|/c\) and typically \(v \ll c\). Under these circumstances an electron will oscillate sinusoidally along the polarization direction of the electric field [6]. At higher intensities this non-relativistic treatment begins to break down.

A dramatic increase of laser intensity will increase the electron velocity \(v\) to become an appreciable fraction of \(c\). Once the intensity becomes large enough, the \(\vec{B}\) term in equation (1.1) cannot be ignored. This causes the electron’s trajectory to be significantly different. Intensities around \(10^{18} \text{ W} \cdot \text{cm}^{-2}\) are in the regime where we must also consider the interaction relativistically. This means that we must consider the theory of relativity when writing down our equations of motion.

1.3 Electron-Laser Experiments

In experiments performed by our group, the laser is focused inside a chamber filled with a dilute gas. After ionization, the electromagnetic field of the laser exerts a force on the liberated electrons. This causes the electron to move in space and accelerate. An accelerating charge also radiates an electromagnetic field. A detector is placed perpendicular from the direction that the laser pulse is traveling. This detector measures the number of photons scattered from free electrons within a given frequency bandwidth. This setup can be seen in Figure 1.2.
1.4 The Challenge of Visualizing Interactions

The non-relativistic description of these laser-particle interactions gives us intuition of what is going on at the microscopic level. An oscillating point charge can be treated like a dipole, which radiates light in a doughnut shape as depicted in Figure 1.3 [7], with the motion of the particle moving up and down through the center of the doughnut. This model of interaction gives us a general idea visually of what is happening but breaks down as the electron moves at relativistic speeds.

At higher intensity, the increased $\vec{B}$ field causes the electron to move in the direction of the laser. From equation (1.1) we see that this force is also velocity dependent, but the velocity is changing directions due to the oscillation in the $\vec{E}$ and $\vec{B}$ fields. Because there are so many things going on at once, it is hard to develop a mental picture of these interactions.

For my thesis project, I created a 3D visualization of these interactions to provide intuition to a broad audience. These will be particularly useful in presentations, to give the audience an idea of...
what is happening in the lab and what we are measuring.

1.5 Overview of the Structure of the Thesis

In Chapter 2, I will discuss the equations that govern the laser-particle interaction. This will include a discussion of what models are used. In Chapter 3, I will discuss the visualization methods used and some of the challenges that had to be overcome. Included at the end of Chapter 3 are some of the snapshots of the final movies and a discussion of why they seem to be accurate.
Chapter 2

Physical Equations Governing Electron-Laser Interactions

In this chapter we will look at the well-known equations that govern the laser-particle interaction. This chapter includes a brief overview of each equation and what it means for these interactions.

2.1 Singh Model of Laser Field

We first consider the laser field since it is what is driving the electron. There are several different ways to model a laser field [8]. Many of these can be numerically intensive due to the fact that they require complicated integrals that do not have closed form solutions. The Singh model of a vector field in a laser focus provides a close approximation to these more complicated models. The advantage of the Singh model is that it avoids any integrals.

The formulas for the $\vec{E}$ and $\vec{B}$ fields are

\[
\vec{E}_{\text{Singh}} = E_0 \left[ \hat{x} + \frac{xy}{2\zeta^2} \hat{y} - i \frac{x}{\zeta} \hat{z} \right] \Psi e^{i(kz - \omega t)} \text{ and } \tag{2.1}
\]

6
\[ \vec{B}_{\text{Singh}} = \frac{E_0}{c} \left[ \frac{xy}{2Z^2} \hat{x} + \hat{y} - i \frac{x}{Z} \hat{z} \right] \Psi e^{i(kz - \omega t)}. \] (2.2)

The \( e^{i(kz - \omega t)} \) factor tells us that the \( \vec{E} \) and \( \vec{B} \) fields are oscillating while traveling in the \( z \) direction. The other parameters of these formula are \( Z = z_0 + iz, \ \Psi = \frac{\rho}{Z} \exp \left( -\frac{\rho^2}{Z^2} \right), \ \rho^2 = x^2 + y^2, \) and \( z_0 = \frac{kw_0^2}{2}. \) Where \( w_0 \) is the beam waist. The beauty of these equations is that by knowing the parameters of our laser beam we approximate the vector field around the focus. Figure 2.1 is an image of the magnitude of the Poynting vector near the focus at a single instant of time. We can see the wave fronts narrow and become more intense.

![Figure 2.1](image_url)

**Figure 2.1** A 2D slice of a laser focus calculated using the Singh model.

Since our beam is not continuous, it is necessary to put it in a temporal envelope. We do this by multiplying \( \vec{E}_{\text{Singh}} \) and \( \vec{B}_{\text{Singh}} \) by \( \exp \left[ - \left( t - \frac{z + \rho^2}{\tau} \right) \right]. \) This gives us a packet that also propagates in the \( z \) direction. To save time computationally, we compute the full field if the envelope is above some arbitrarily threshold. Otherwise, we set the entire field to zero.
2.2 Relativistic Equations of Motion for an Electron

Since our laser is powerful enough, we must consider relativistic effects as we are calculating the trajectory of the electron. The equation of motion for the particle is

\[ \frac{d\vec{p}}{dt} = q(\vec{E} + \vec{v} \times \vec{B}). \]  

(2.3)

We can calculate \( \vec{E} \) and \( \vec{B} \) using the equations in the previous section. Since we are making a relativistic argument we will require the relation

\[ \vec{p} = \gamma m\vec{v}. \]  

(2.4)

Now we have introduced a different unknown, \( \gamma \). To calculate this we make use of

\[ \sqrt{(pc)^2 + (mc^2)^2} = \gamma mc^2. \]

Solving for gamma and simplifying yields

\[ \gamma = \sqrt{\left(\frac{p}{mc}\right)^2 + 1}. \]  

(2.5)

Now that we have all the necessary parts that govern the motion of the electron, we can specify some initial conditions and solve for the trajectory numerically.

2.3 Electron Trajectories

At this point it is useful to check the electron trajectories and see if they provide realistic results. We can put the equations of motion into MATLAB’s ode45 function, which uses a fifth order Runge-Kutta method to get an electron trajectory. Figure 2.2 is a plot of the electron trajectory when the peak intensity is low, about \( 10^{16} \text{ W} \cdot \text{cm}^{-2} \). We can see that most of its motion is in the y direction, but it has a small velocity in the z direction due to the small magnetic field.
2.3 Electron Trajectories

![Figure 2.2](image)

**Figure 2.2** Trajectory of a free electron when it is hit by a pulse with peak intensity is $10^{16}$ W cm$^{-2}$.

Since we see that the low intensity case has shown to be correct we change the intensity to $10^{18}$ W cm$^{-2}$ and get Figure 2.3.

![Figure 2.3](image)

**Figure 2.3** Trajectory of a free electron as it is hit by a pulse with the peak intensity is $10^{18}$ W cm$^{-2}$. 
We can see that the electron travels a lot further in the high intensity case. Once it is kicked out of the laser focus, it travels with constant velocity. These graphs are useful in checking the computations and are the foundation of the more informative animations that show how the changing laser field affects the electron.

2.4 Radiated Fields

The radiated fields are based on formulas derived from the Liénart-Wiechert Potentials [9]. The \( \vec{E} \) field of a given point at space at a given point of time is

\[
\vec{E}(\vec{r}, t) = \frac{q}{4\pi \varepsilon_0} \frac{\vec{r}'}{(\vec{r}' \cdot \vec{u})^3} \left[ (c^2 - v^2)\vec{u} + \vec{r}' \times (\vec{u} \times \vec{a}) \right].
\]  

(2.6)

In this we have introduced several new vectors. They are defined as follows:

\[ \vec{u} = c\vec{r}' - \vec{v}, \]

\[ \vec{r}' = \vec{r} - \vec{w}(t_r), \text{ and} \]

\[ \vec{v} = \vec{w}(t_r), \]

where \( \vec{w}(t) \) is the position on the electron at time \( t \). In these definitions we have introduced the idea of the retarded time, \( t_r \). The idea is that the field associated with the position \( \vec{r} \) is determined by what the electron was doing at some previous time since the light takes time to propagate. The retarded time is computed using

\[ |\vec{r} - \vec{w}(t_r)| = c(t - t_r). \]
From this definition of the retarded time we can understand what the physical significance of the vectors $\vec{r}$, $\vec{w}(t_r)$, and $\vec{r}'$. We desire to calculate the electric field at point $P$. As seen in Figure 2.4, $\vec{r}$ is the vector that goes from the origin to point $P$. At the time we are measuring the electromagnetic radiation of point $P$, the position of the electron is $\vec{w}(t_r)$. The position of point $P$ from the electron at the retarded time is given by $\vec{r}'$.

![Particle Trajectory](image)

**Figure 2.4** A visualization of the vectors used to compute the retarded time.

One of the great difficulties of creating a solution to this equation is that the calculation to find the retarded time is transcendental. I overcame this numerically, by subtracting everything to one side yielding
\( | \vec{r} - \vec{w}(t_r) | - c(t - t_r) = 0. \)

We then can input any \( \vec{r} \) and try it for all possible values for \( t_r \). Looking at where the left hand side of this equation is at zero tells us where \( t_r \) should be. If we make our time steps small enough we are able to get very accurate results. It is also worth noting that since an electron cannot travel faster than light, this equation will have only one solution.
Chapter 3

Visualization Methods

In this chapter, we discuss how to project fields in a 3D space onto a 2D computer screen. To conclude, we will look at a few animations that the final code produced and interpret the results.

3.1 Virtual Camera

To display 3D space onto a 2D plane I used a virtual camera. The code for the virtual camera was developed previously by Ryan Sandberg [10]. The way this is done can be seen in Figure 3.1, which was taken from Ryan Sandberg’s thesis where he was calculating the probability densities of quantum electron wave packets in a laser field. For each frame of our movie, we calculate the intensity of each pixel on the screen. A line is drawn from our image plane to a pinhole. That line continues through space. At discrete locations on that line, the program calculates the magnitude of the laser field, electron, or the radiated field. It then sums all of the values and store in an element of a two dimensional array that corresponds to the pixel of our image.
3.2 Displaying Laser and Electron Fields

The laser field is displayed using the Singh Model as discussed in Chapter 2. To display this vector field in space we simply calculate the magnitude. At first it seemed necessary to plot the Poynting vector, but this would take the computer a long time to calculate. I realized that the same amount of visual information would be expressed by plotting the magnitude of the electric field. Doing this allows us to ignore the magnetic field and save time computationally.

The electron is displayed as a small sphere. This is done by modeling it as a field that falls off with $r^K$, where $K$ is some constant. The arbitrary form of the electron used for visual appeal was

![Figure 3.1 Visualization of how the Virtual Camera projects 3D space onto a 2D plane.](image)
\[ a = \frac{1}{|\vec{r} - \vec{r}_0|^{1/4}}. \] (3.1)

Where \( a \) is the "amount" of the electron at the point we are interested, \( \vec{r} \) is the position we are calculating, and \( \vec{r}_0 \) is the current position of the electron. The \( 1/4 \) power was chosen because it gave a smooth look and a nice color gradient to our electron. To get this field to appear as a sphere, we pick a minimum number for \( a \) and anything that is below it we set to zero. This also gives us a convenient way to control the size of the electron that appears on the screen. Raising the value would decrease the size of the electron; lowering the value would increase the size of the electron. To account for the singularity when \( \vec{r} = \vec{r}_0 \), we also include a cap for \( a \).

The final part of the animation to display is the radiation field. It is displayed by calculating the magnitude of the electric field as in Eq. (2.6). This gives us all the components needed for our movie.

### 3.3 Gridlines

The symmetry of the visualization makes it hard to see perspective. The laser focus has cylindrical symmetry, and the electron has spherical symmetry. To help give the viewer additional spatial perspective, I added gridlines to the visualization. This was done by using the same virtual camera. At each point in space the program would see if it was within some some preset distance to a integer line of our axis. If it was then it would be assigned a value of 1. Doing this provided a computationally quick way to display gridlines. Figure 3.2 show how adding gridlines helps give perspective.
3.4 Test Case

With any computational project in any field it is desirable to have something to test code. Some sort of benchmark is needed to see that the code is doing what is expected. At low intensity of the laser we expect the electron to behave like an oscillating point charge. This is due to the fact that the $\vec{B}$ field is negligibly small. Figure 3.3 juxtaposes a single image of this situation with a dipole comparison [7].

![Figure 3.2](image1)  
**Figure 3.2** Comparing how gridlines add perspective to our images.

![Figure 3.3](image2)  
**Figure 3.3** By lowering the intensity we can see that our model is similar to a dipole radiation, as expected.

Keeping in mind that only the electric field of the radiation is being plotted, we see that it closely
matches the dipole case. Having gained the confidence that the code is working properly, we simply turn back up the intensity and see what happens.

### 3.5 Movies of Electron-Laser

The first animation that we consider is given in Figure 3.4. It is a case where the electron starts from rest at the focus of the laser well before the pulse has arrived. When the pulse is far away the electron oscillates mildly, and we get dipole-like radiation. As the pulse envelope gets closer we see that the electron drifts downstream and begins to oscillate more violently. The radiated fields get increasingly more intense. As the center of the laser pulse approaches the focus, the electric and magnetic fields increase with time. This means that the electric field that pushed the electron down is slightly stronger than the field that pushed it up. Eventually, the electron is kicked out of the path of the laser pulse. With no forces acting on it the electron travels with constant velocity. As time progresses, a small Doppler shift, owing to forward drift of the electron, is apparent by the distance between the wave fronts narrowing in one direction and getting further apart in the other. Eventually, the electron moves off camera and we finish watching the laser pulse pass by.

![Figure 3.4](image)

**Figure 3.4** Frames from a movie when the electron had no initial momentum.

One interesting case is when the electron is given a small amount of initial momentum in the
upstream direction. When the electron begins to interact with the laser, the amount of radiation increases dramatically. The electron slows down slightly but quickly gets kicked out of the focus as seen in Figure 3.5. There is also a very distinct Doppler shift in the radiation of the electron.

![Frames from movie when the electron have initial momentum upstream.](image1)

**Figure 3.5** Frames from movie when the electron have initial momentum upstream.

In another example, I gave the electron initial momentum that corresponded to 0.984 c, or to the electron having 1.5 times its rest energy. We can see that there is a build up of electric field in the direction of the electrons motions as shown in Figure 3.6. This is due to the fact that the electron is traveling near the speed of light.

![Frames from movie when the electron have initial relativistic momentum upstream.](image2)

**Figure 3.6** Frames from movie when the electron have initial relativistic momentum upstream.
These animations have helped us gain physical intuition and understanding of what these interactions look like. They demonstrate some very simple effects, such as dipole radiation, or more complex relativistic effects. There are many more possibilities of interaction that what are listed above. In Appendix A, I have included all the code so anyone can modify the initial conditions and constants to make their own animations, and gain intuition about whatever situation fits their needs.
Appendix A

MATLAB Code

To make a movie, first run the program Moving Electron to test initial conditions. The variables you can change are laser intensity I0 in EBfield and the initial position and momentum of the electron given by u0 in Moving Electron. Moving Electron will give you a plot of what your electron will do, and you can use this to ensure that it will be visible in your movie. Then movie the same initial conditions to electronPath. Change the ti and tf to match in moviewer and choose your camera constants. Hit run and in about 8 hours you will have your movie.

A.1 Moving Electron

```matlab
1 clear
close all
% This is used to check the electrons path. It will plot it like it will at
5 % if the camera is at zero angle change
   ti=-20;
tf=20;
nframe=1000;
10 dt=(tf-ti)/nframe; % this just effects the interpelation
   sStep=2;
15 options=odeset('RelTol',1e-8);
   % initial conditions [y,z,x,py,pz,px]
   u0=[0,0,0,0,0,0];
20 [t,u]=ode45(@rhs,[ti,tf],u0,options);
x=u(:,1);
```
A.2 movieviewer

\[
y = u(:, 2);
z = u(:, 3);
px = u(:, 4);
py = u(:, 5);
pz = u(:, 6);
\]
\[
\text{tfine}=t_0:dt:tf;
i = \text{interpl}(t, x, \text{tfine}, 'spline');
iy = \text{interpl}(t, y, \text{tfine}, 'spline');
zi = \text{interpl}(t, z, \text{tfine}, 'spline');
\]
\[
zplot = \text{min}(zi):0.01:\text{max}(zi);
xplot = -\sqrt{1+zplot.^2};
xplot2 = \sqrt{1+zplot.^2};
\]
\[
\text{figure}
\]
\[
\text{plot}(zi, xi, zplot, xplot, '--r', zplot, xplot2, '--r');
\]
\[
\text{axis}('square')
\]
\[
\text{xlabel}('z position (in laser wavelengths')
\]
\[
\text{ylabel}('y position (in laser wavelengths')
\]
\[
\text{legend}('Electron Trajectory', 'Laser Focus')
\]
\[
\text{for } n=1: \text{length}(\text{tfine})
\]
\[
\text{plot3}(zi(1:n), xi(1:n), 'b-', zi(n), yi(n), xi(n), 'y*');
\]
\[
\text{xlabel}('x')
\]
\[
\text{ylabel}('y')
\]
\[
\text{zlabel}('z')
\]
\[
\text{axis}([-5 5 -5 5 -5 5])
\]
\[
\text{grid} on
\]
\[
\text{T}=t_0+(n-1)*dt;
\]
\[
\text{title}(['T= ', \text{num2str}(T, '%.2f')])
\]
\[
\text{pause}(0.1)
\]
\[
\text{end}
\]
\[
c = 3e8;
\]
\[
\text{gamma} = \sqrt{px.^2+py.^2+pz.^2+1};
\]
\[
\text{vx} = px./\text{gamma};
\]
\[
\text{vy} = py./\text{gamma};
\]
\[
\text{vz} = pz./\text{gamma};
\]
\[
\text{v} = \sqrt{vx.^2+vy.^2+vz.^2};
\]
\[
\text{maxVel} = \text{max}(v)
\]

A.2 movieviewer

\[
\text{clear; close all;}
\]
\[
\text{format long;}
\]
\[
\text{%%%%%%%%%%%%%%%%%%%%%%%%} \text{Essential Movie Parameters}%%%%%%%%%%%%%%%%%%%%%%%%
\]
\[
\text{Use these to craft your movie}
\]
\[
\% \text{Don't forget to set 'nframe' to 600 or whatever is appropriate for your movie}
\]
\[
\% and uncomment the commands to save the frames as tifs (found at the end}
\]
\[
\% of the program) when you have the parameters you want
\]
\[
\% \text{Movie title, will appear in frame titles, followed by 'frame'(framenumber)}
\]
\[
\% i.e. if I set moviename to 'slimer', then the first frame will be titled}
\]
\[
\% 'slimerframe1'
\]
\[
\% NOTE: Don't use numbers in moviename!
\]
\[
\text{moviename='Fine2Test'};
\]
\[
\text{Start and stop times}
\]
\[
\% The options to look at a single time is in the main loop
\]
\[
\text{tmin}=-20;
\]
\[
\text{tmax}=20;
\]
% number of frames
nframe = 800;

stepSize = 1;

% pick nframe and stepSize so that dt is about .1
% Both must be integers
dt = (tmax - tmin) / nframe / stepSize;  % Don't change this

% for theta a positive number angles the camera to the right
% for phi a positive number angles the camera down
thetaMin = -pi / 12;
thetaMax = -pi / 12;
phiMin = pi / 6;
phiMax = pi / 6;
dtheta = (thetaMax - thetaMin) / nframe;

dphi = (phiMax - phiMin) / nframe;

% This sets up how the camera moves Positive number in first entry moves the
% camera up. Positive number in second entry move the camera left This is
% where the camera is looking at

thetam = [0, 0, 0];
theatform = [0, 0, 0];

% Number of pixels in final image !!! MUST BE ODD!! (unless you change code
% everywhere)
% Image is an array, so there are mfinal rows by nfinal columns
mfinal = 1001;
nfinal = 1001;

% aspin is the rotation angle of the camera about the camera viewing axis
% (the vector r0). It's better to leave aspin alone.

aspin = 0;

% One way to change the zoom is to change how close the camera is to the
% wave packet. nlr0 is a multiplier that scales this camera-wavepacket
% distance

nlr0 = 5;

% ADDITIONAL CAMERA PARAMETERS IN THE START OF THE MAIN LOOP

% These are the colors of the differt parts of the movie
wavecolor = [.8, 0, 0];
electroncolor = [1, 1, 0];
wispcolor = [0, 0, .2];
gridcolor = [.3, .3, .3];

% ------------------------------------- Development of camera constants -------------------------------------
% You could change these, but I don't recommend it unless you know what
% you're doing. You have already set the important parameters in the
% section above. The only thing I would change is m. If you lower it your
% movie will be made MUCH faster but the quality will go down.

% camera constants
r0 = [0, 0, 2];  % initial vector from rmid to pinhole
c = 1.5;  % distance from pinhole to image plane
camheight = 1.51;  % set size of camera
aratio = mfinal / nfinal;  % ratio of camera width to height
camwidth = aratio * camheight;
m = 151;  % number of rows of pixels !! must be odd!!
nposse = floor(m * aratio);
if (mod(nposse, 2) == 0)
n = nposse + 1;
else
n = nposse;
end
piw=camwidth/m; % pixel width
100 pih=camheight/n; % pixel height
% setting up the initial pixel vectors
camx=[piw,0,0];
camy=[0,pih,0];

% Lateral offset of camera focus from expected center of wave packet
x0=0;
y0=0;

% Get the information about the electron's path
% Change in iteration conditions in the electronPath function
[x0,z0,y0,vx0,vz0,vy0,ax0,az0,ay0]=electronPath(tmin,tmax,dt);
y0=-y0;
vy0=-vy0;

% This sets or coordinate system to be +x is to the right
% +y is up and +z is into the screen
[wavemax,electronmax,waspmx,gridmax]=calcMaxs(c,camx,camy,nframe,m,n,mfinal,
tfinal,tmin,dt,x0,y0,z0,vx0,vz0,vy0,vz0,ax0,ay0,az0);

% ------------ Program starts working here! -----------------------------
for s=1:400:nframe+1 % Loops through each frame of the movie
  tic

  % Camera parameters
  % These two lines allow you to cycle through different times
  t=tmin+(s-1)*dt*stepSize;
s0=(s-1)*stepSize+1;
  % These two lines allow you to look at a single time
  % t=-20;
  % s0=length(tmin:dt:t);

  % These two lines allow for a "smooth" transition
  % This causes the camera to pan between two points
  thetaw=-(thetaMax-thetaMin)/2*cos(pi*(s-1)/(nframe+1))+thetaMax/2;
  phiw=-(phiMax-phiMin)/2*cos(pi*(s-1)/(nframe+1))+phiMax/2;

  % These two lines allow for a "linear" transition
  % This causes the camera to be fixed to the electron
  thetaw=thetaMin+(s-1)*dtheta;
  phiw=phiMax+(s-1)*dphi;

  % Not tested but this should go between two points smoothly
  % This causes camera to be fixed to the electron
  % approx position of focus this will be where the camera is
  % looking at.
  % nrmid=[x0,y0,0]+epos;

  adjr0=r0*n1r0;

  % dx=19/(m-1); % Why 19? It's completely arbitrary right now.
  % It is an artifact of earlier work.
end

% Approximate position of focus this will be where the camera is
% looking at.

nr0=Rotate(0,thetaw,phiw,adjr0);
ncamx = Rotate(aspin,thetaw,phiw,camx);
ncamy = Rotate(aspin,thetaw,phiw,camy);
A.3 electronPath

function \[x0,y0,z0,vx0,vy0,vz0,ax0,ay0,az0\]=electronPath(tmin,tmax,dt)
%This function outputs the electrons position, velocity, and acceleration.
%Because the electron is moving so fast we have to consider it
%relativistically.
Set constants
\( q = 1.60 \times 10^{-19} \);
\( m = 9.11 \times 10^{-31} \);
\( c = 2.99 \times 10^8 \);
\( T = 800 \times 10^{-9} / c \);

ode options
options = odeset('RelTol', 1e-8);

% initial conditions \([y, z, x]\)
\( u_0 = \left[ 0, 0, 0, 0, 0, 0 \right] \);
\([t, u] = \text{ode}45(@rhs, [tmin, tmax], u_0, options)\);

% now that we have the solution we put it to equal time intervals
\( t_{\text{fine}} = \text{tmin} : \text{dt} : \text{tmax} \);

% This accounts for relativity
\( \gamma = \sqrt{\left( p_x(\text{n})^2 + p_y(\text{n})^2 + p_z(\text{n})^2 \right) + 1} \);
\( v_x(\text{n}) = p_x(\text{n}) / \gamma \);
\( v_y(\text{n}) = p_y(\text{n}) / \gamma \);
\( v_z(\text{n}) = p_z(\text{n}) / \gamma \);

% This gives us the accelerations at our time points
\([E, B] = \text{EBfield}(x_0(\text{n}), y_0(\text{n}), z_0(\text{n}), t_{\text{fine}}(\text{n}), 1)\);
\( a_x(\text{n}) = q * T / m / c * (E(1) + v_y(\text{n}) * B(3) - v_z(\text{n}) * B(2)) \);
\( a_y(\text{n}) = q * T / m / c * (E(2) + v_z(\text{n}) * B(1) - v_x(\text{n}) * B(3)) \);
\( a_z(\text{n}) = q * T / m / c * (E(3) + v_x(\text{n}) * B(2) - v_y(\text{n}) * B(1)) \);

\( f = \text{ode}45(@rhs, [tmin, tmax], u_0, options)\);

function \( F = \text{rhs}(t, u) \)
\( q = 1.60 \times 10^{-19} \);
\( m = 9.11 \times 10^{-31} \);
\( c = 2.99 \times 10^8 \);
\( T = 800 \times 10^{-9} / c \);
\( x = u(1) \);
\( y = u(2) \);
\( z = u(3) \);
A.5 EBfield

\[ p_x = u(4); \]
\[ p_y = u(5); \]
\[ p_z = u(6); \]

15 \texttt{gamma} = \texttt{sqrt}((p_x^2 + p_y^2 + p_z^2) + 1);

\[ [E, B] = \texttt{EBfield}(x, y, z, t, 1); \]

\[ vx = p_x * c / \texttt{gamma}; \]
\[ vy = p_y * c / \texttt{gamma}; \]
\[ vz = p_z * c / \texttt{gamma}; \]
\[ F = \texttt{zeros}(6, 1); \]

20 \[ F(1) = p_x / \texttt{gamma}; \]
\[ F(2) = p_y / \texttt{gamma}; \]
\[ F(3) = p_z / \texttt{gamma}; \]
\[ F(4) = q * T / m / c * (E(1) + vy * B(3) - vz * B(2)); \]
\[ F(5) = q * T / m / c * (E(2) + vz * B(1) - vx * B(3)); \]
\[ F(6) = q * T / m / c * (E(3) + vx * B(2) - vy * B(1)); \]

A.6 calcMaxs

\[ \texttt{tic} \]
% Camera parameters -------------------------
\[
t = 0 - dt;
\]
\[
s_0 = \text{floor}\left(\frac{\text{nframe} + 1}{2}\right);
\]
\[
\theta_w = 0;
\]
\[
\phi_w = 0;
\]
\[
\text{epos} = 0;
\]
\[
x_0 = 0;
\]
\[
y_0 = 0;
\]
\[
\text{aspin} = 0;
\]
\[
\text{adjr0} = [0, 0, 10];
\]
\[
\text{nrmid} = [x_0, y_0, 0] + \text{epos};
\]
\[
\text{nrx} = \text{Rotate}(0, \theta_w, \phi_w, \text{adjr0});
\]
\[
\text{ncamx} = \text{Rotate}(\text{aspin}, \theta_w, \phi_w, \text{ncamx});
\]
\[
\text{ncamy} = \text{Rotate}(\text{aspin}, \theta_w, \phi_w, \text{ncamy});
\]
\[
\text{dx} = 19 / (\text{m} - 1); \quad \% \text{Why 19? It's completely arbitrary right now.}
\]
\[
\text{dy} = 19 / (\text{n} - 1);
\]
\[
[\text{Xi}, \text{Yi}] = \text{ndgrid}(1 : \text{dx} : 20, 1 : \text{dy} : 20);
\]
\[
\% \text{grid, waveintensity}
\]
\[
\text{[waveintensity]} = \text{calculateWave}(\text{m}, \text{n}, \text{nr0}, \text{nrmid}, \text{ncamx}, \text{ncamy}, \text{c}, \text{t});
\]
\[
\text{[electronintensity]} = \text{calculateElectron}(\text{m}, \text{n}, \text{nr0}, \text{nrmid}, \text{ncamx}, \text{ncamy}, \text{c}, \text{t}, x_0(s_0), y_0(s_0), z_0(s_0));
\]
\[
\text{[wispintensity]} = \text{calculateWisp}(\text{m}, \text{n}, \text{nr0}, \text{nrmid}, \text{ncamx}, \text{ncamy}, \text{c}, \text{t}, \text{tmin}, \text{dt}, ... x_0(1:s_0), y_0(1:s_0), z_0(1:s_0), v_x(1:s_0), v_y(1:s_0), v_z(1:s_0), a_x(1:s_0), a_y(1:s_0), a_z(1:s_0));
\]
\[
\text{[gridintensity]} = \text{calculateGrid}(\text{m}, \text{n}, \text{nr0}, \text{nrmid}, \text{ncamx}, \text{ncamy}, \text{c}, \text{t});
\]
\[
\% \text{final grid calculation, finalintensity interpolation}
\]
\[
\text{dx}_i = 19 / (\text{mfinal} - 1); \quad \% \text{Deciding interpolation of final image}
\]
\[
\text{dy}_i = 19 / (\text{nfinal} - 1);
\]
\[
[\text{Xi}, \text{Yi}] = \text{ndgrid}(1 : \text{dx}_i : 20, 1 : \text{dy}_i : 20);
\]
\[
\text{finalwaveintensity} = \text{interp2}(\text{Xi}', \text{Yi}', \text{waveintensity}', \text{Xi}', \text{Yi}');
\]
\[
\text{finalelectronintensity} = \text{interp2}(\text{Xi}', \text{Yi}', \text{electronintensity}', \text{Xi}', \text{Yi}');
\]
\[
\text{finalwispintensity} = \text{interp2}(\text{Xi}', \text{Yi}', \text{wispintensity}', \text{Xi}', \text{Yi}');
\]
\[
\text{finalgridintensity} = \text{interp2}(\text{Xi}', \text{Yi}', \text{gridintensity}', \text{Xi}', \text{Yi}');
\]
\[
\text{waveframe} = \text{zeros}(\text{mfinal}, \text{nfinal}, 3);
\]
\[
\text{electronframe} = \text{zeros}(\text{mfinal}, \text{nfinal}, 3);
\]
\[
\text{wispframe} = \text{zeros}(\text{mfinal}, \text{nfinal}, 3);
\]
\[
\text{gridframe} = \text{zeros}(\text{mfinal}, \text{nfinal}, 3);
\]
\[
\% \text{This loop makes the color image 'finalimage' from the calculated projection of the}
\]
\[
\% \text{wave packet in the array 'finalintensity'}
\]
\[
\text{wavecolor} = [.8, 0, 0];
\]
\[
\text{electroncolor} = [1, 1, 0];
\]
\[
\text{wispcolor} = [1, 1, 1];
\]
\[
\text{gridcolor} = [.3, .3, .3];
\]
\[
\% \text{for k=1:mfinal}
\]
\[
\text{for l=1:nfinal}
\]
\[
\text{waveframe(k, l, :) = finalwaveintensity(k, l) * wavecolor;}
\]
\[
\text{electronframe(k, l, :) = finalelectronintensity(k, l) * electroncolor;}
\]
\[
\text{wispframe(k, l, :) = finalwispintensity(k, l) * wispcolor;}
\]
\[
\text{gridframe(k, l, :) = finalgridintensity(k, l) * gridcolor;}
\]
\[
\text{end}
We much divide out the maximum in in order to get a not black image. 
Unfortunately the maximum depends on the individual parameters. I would
suggest uncommenting the four lines below, run your movie with lower
quality then finding the max of each of them

\[
\begin{align*}
\text{wave max} &= \max(\max(\max(\text{wave intensity}))) \\
\text{electron max} &= \max(\max(\max(\text{electron frame}))) \\
\text{wisps max} &= \max(\max(\max(\text{wisps frame}))) \\
\text{grid max} &= \max(\max(\max(\text{grid frame})))
\end{align*}
\]

If the electron is off screen when we take our snapshot we run into 
problems because we are dividing by zero. This value should be close 
%to other instances

\[
\text{if electron max} = 0 \\
\text{electron max} = 0.0025;
\]

These are used to check as you go

\[
\text{imagesc}(\text{final image})
\]

axis off;  
Title = sprintf('t = %.4f', t); 

\[
\text{title}(\text{Title})
\]

drawnow

\[
\text{set}(0, 'DefaultImageCreateFcn', 'axis image')
\]

\[
\text{toc}
\]

A.7 Rotate

\[
\text{function new Vector = Rotate(aspin, theta, phi, vector)}
\]

% Rotates the camera  
\[
R1 = \begin{bmatrix}
\cos(\phi), 0, \sin(\phi); 0, 1, 0; -\sin(\phi), 0, \cos(\phi)
\end{bmatrix};
\]

\[
R2 = \begin{bmatrix}
1, 0, 0; \cos(\theta), \sin(\theta), 0; -\sin(\theta), \cos(\theta)
\end{bmatrix};
\]

\[
R3 = \begin{bmatrix}
\cos(\text{aspin}), -\sin(\text{aspin}), 0; \sin(\text{aspin}), \cos(\text{aspin}), 0; 0, 0, 1
\end{bmatrix};
\]

\[
\text{new Vector} = (R2*R1*R3*vector')';
\]

A.8 calculateWave

\[
\text{function [intensity]} = \text{calculateWave}(m, n, r0, rmid, camx, camy, c, t)
\]

% An expansion of what this does can be found in Ryan Sandburg's thesis 2012  
stepSize = .015; 

delx = norm(camx); 

dely = norm(camy); 

\[
\text{pinhole} = \text{rmid} + r0; % position of the pinhole (for convenience)
\]

\[
\text{intensity} = \text{zeros}(m, n); % initializing image array
\]

\[
\text{r} = .25 + 3*t/200; % approximation to size of electron cloud
\]

\[
r1 = 1; % when pw is small
\]

\[
r2 = \text{norm}(r0); % distance from pinhole to electron
\]

\[
\text{center}
\]

\[
\text{integrating through the wavepacket at a specified time}
\]

\[
\text{for k} = 1:m
\]

\[
\text{test row} = \text{zeros}(1, n);
\]

\[
\text{for l} = 1:n
\]
function [magS] = calculateS(x,y,z,t)
% Calculates magnitude of poynging vector
z0=pi;\%=k*w0^2/2
5 \% w0 is effective radius
c=2.99e8;
T=800e-9/c;
tau=35e-15/T;
10 R=x+z0^2/2;
rhosquared = x^2 + y^2;
% To improve speed we check if this function is above a limit. This is the
15 % gaussian part of the function
envelopefunc=exp(-(t-x-rhosquared/2/R)/tau)^2);
if envelopefunc > .1 \%I like to use .05 but .1 seems to also be good. Higher=
fast/less accurate.
20 [E,\,] = EBfield(x,y,z,t,0);
function [intensity] = calculateElectron(m,n,r0,rmid,camx,camy,c,t,x0,y0,z0)
    % An expansion of what this does can be found in Ryan Sandburg's thesis 2012
    stepSize = .015;
    delx = norm(camx);
    dely = norm(camy);
    pinhole = rmid + r0; % position of the pinhole (for convenience)
    intensity = zeros(m,n); % initializing image array
    r = .25 + 3 * t / 200; % approximation to size of electron cloud
    r2 = norm(r0); % distance from pinhole to electron center
    % integrating through the wavepacket at a specified time
    for k = 1:m
        testrow = zeros(1,n);
        for l = 1:n
            pinhole = pinhole;
            dvec = -1/norm(r0) * c * r0 - (k - (m+1)/2) * camx ... 
                - (1-(n+1)/2) * camy; % dvec is vector from pixel to pinhole
            % calculate the bounds of integration:
            v1 = (r2 - r) / c; % approximate start and stop positions of wavepacket
            v2 = (r2 + r) / c;
            u1 = (r2 - 1) / c; % bounds to the grid calculator
            u2 = (r2 + 1.75) / c;
            U = u1:stepSize:u2;
            indexV = find((U >= v1) & (U <= v2));
            lRho = length(indexV);
            rhoL = zeros(lRho+1,1);
            phaseL = zeros(lRho,1);
            for u = U3:U4
                x = pinhole(1) + U(u) * dvec(1);
                y = pinhole(2) + U(u) * dvec(2);
                z = pinhole(3) + U(u) * dvec(3);
                v = u + 2 - U3; % rhoL(i)=0, then we add 0 to every other result
                rhoL(v) = calculateSphere(x,y,z,x0,y0,z0); % probability calculator that includes volume effects
                dOut = c * U(u); % distance from pinhole
                dElV = 1 / 3 * delx * dely / c^2 * (dOut-c*stepSize)^3; % volume containing the rhoL(v) probability density
                rhoL(v) = rhoL(v) * dElV; % prob. = prob. density * volume
                Add = rhoL(v) + rhoL(v-1);
                rhoL(v) = Add;
            end
            testrow(1) = rhoL(lRho+1); % get last value of rhoL
        end
        intensity(k,:) = testrow;
    end
end
### A.11 calculateSphere

```matlab
function [density] = calculateSphere(x, y, z, x0, y0, z0)
    % We model the electron as a sphere. Most of the values in this were found
    % using trial and error. The power in the first function determines the
    % shading. The if else statements control the size.
    a = 1 / ((x-x0)^2+(y-y0)^2+(z-z0)^2)^(1/4);
    if a > 4 && a < 10
        density = a;
    elseif a >= 10
        density = 10;
    else
        density = 0;
    end
end
```

### A.12 calculateWisp

```matlab
function [intensity] = calculateWisp(m, n, r0, rmid, camx, camy, c, t, tmin, dt, x0, y0, z0, vx0, vy0, vz0, ax0, ay0, az0)
    % An expansion of what this does can be found in Ryan Sandburg's thesis 2012
    stepSize = 0.015;
    delx = norm(camx);
    dely = norm(camy);
    pinhole = rmid + r0; % position of the pinhole (for convenience)
    intensity = zeros(m, n); % initializing image array
    r = 0.25 + 3 * t / 200; % approximation to size of electron cloud
    r = 1; % when pw is small
    r2 = norm(r0); % distance from pinhole to electron
    % center
    % integrating through the wavepacket at a specified time
    for k=1:m
        testrow = zeros(1, n);
        indexV = find((U >= v1) & (U <= v2));
        U = U + stepSize * v1;
        U1 = U + stepSize;
for u=U3:U4
    x=pinhole(1)+U(u)*dvec(1);
    y=pinhole(2)+U(u)*dvec(2);
    z=pinhole(3)+U(u)*dvec(3);
end

v=u+2-U3; % rhoL(1)=0, then we add 0 to every other result
[rhol(v)]=calculateWispField(x,y,z,x0,y0,z0,vx0,vy0,vz0,ax0,ay0,az0,t,
    tmin,dt);
    % probability calculator that includes volume effects
dOut=c*U(u); % distance from pinhole
delV=1/3*delx*delx*delx/c^2*(dOut^3-(dOut+c*stepSize)^3);
    % volume containing the rhoL(v) probability density
    rhoL(v)=rhoL(v)*delV; % prob. = prob. density * volume
    Add=rhoL(v)+rhoL(v-1);
    rhoL(v)=Add;
end
testrow(l)=rhoL(lRho+1); % get last value of rhoL
end

intensity(k,:)=testrow;

% ---------------Testing validity of volume factor in calculating probability
intensitysum=sum(sum(intensity));
fprintf('Time: %.2f\tSum of wisp array: %.3f\n', t, intensitysum)
% % ---------------End

A.13 calculateWispField

function [density]=calculateWispField(x,y,z,x0,y0,z0,vx0,vy0,vz0,ax0,ay0,az0,t,
    tmin,dt)
    %This calculates the field that is caused by the electron

tvec=tmin:dt:t;
    tvec=tvec';
    %To find the restarted time that we must evaluate at we solve the
tresedental equation t=t+norm(1)/c. We do so numerically my taking the
    %minimum. We look at what intex that is at and use that in our
    %calculations.
    [~,I]=min(abs((tvec-t)+sqrt((x-x0).^2+(y-y0).^2+(z-z0).^2)));
    %I believe these equations can be found in any senior level E&M book. I
    %know it is chapter 10 of the griffels book.
rd=[x,y,z];
    r0=[x0(I),y0(I),z0(I)];
u=[vx0(I),vy0(I),vz0(I)];
a=[ax0(I),ay0(I),az0(I)];
l=abs(rd-r0);
    Xi=l/norm(l)-u;
    %3.52e-9 is for 800 nm light
    %The code will still work for other wave lengths since it scales at the end
    %This calculates both fields
    %E=3.52e-9*norm(1)/((norm(dot(1,Xi)))^3*(1-norm(u)^2)*Xi+fastcross(1,fastcross(Xi,
    %a)));
    %This just calculates the far field
E=3.52e-9*norm(1)/((norm(dot(1,Xi)))^3*(fastcross(1,fastcross(Xi,a)));
    density=norm(E);
    %The field is infinity close to the charge. We set a cap to avoid that.
A.14 fastcross

1 function [C] = fastcross(A,B)
% The cross command that MatLab uses is very slow. I got tired of writing
% cross products so I started using this simple function.

C(1)=A(2)*B(3)-A(3)*B(2);
C(2)=A(3)*B(1)-A(1)*B(3);
C(3)=A(1)*B(2)-A(2)*B(1);
end

A.15 calculateGrid

1 function [intensity]=calculateGrid(m,n,r0,rmid,camx,camy,c,t)
% An expansion of what this does can be found in Ryan Sandburgs thesis 2012
stepSize=.016;
delx=norm(camx);
dely=norm(camy);
phole=rmid+r0; % position of the pinhole (for convenience)
intensity=zeros(m,n); % initializing image array
% r=.25+3*t/200; % approximation to size of electron cloud
r=1; % when pw is small
r2=norm(r0); % distance from pinhole to electron
% center
% integrating through the wavepacket at a specified time
for k=1:m
    testrow=zeros(1,n);
    for l=1:n
        pinholet=pinhole;
        dvec=(-1/norm(r0)*c*r0-(k-(m+1)/2)*camx...
             -(-1-(n+1)/2)*camy); % dvec is vector from pixel to pinhole
        % calculate the bounds of integration:
        v1=(r2-r)/c; % approximate start and stop positions of wavepacket
        v2=(r2+r)/c;
        u1=(r2-1)/c; % bounds to the grid calculator
        u2=(r2+1.75)/c;
        U=u1:stepSize:u2;
        indexV=find((U>v1)&(U<v2));
        rhol=length(indexV);
        U3=indexV(1); U4=indexV(rhol); % the indices marking the approximate e-
        rhol=zeros(1,rhol+1,1);
        phas=zeros(1,rhol,1);
        for u=U3:U4
            x=pinhole(1)+U(u)*dvec(1);
            y=pinhole(2)+U(u)*dvec(2);
            z=pinhole(3)+U(u)*dvec(3);
            v=u+2-U3; % rhol(1)=0, then we add 0 to every other result
            rhol(v)=calculateGridLine(x,y,z);
            % probability calculator that includes volume effects
A.16 calculateGirdLine

```matlab
function [magS] = calculateGirdLine(x,y,z)
    n = -50:50;
    cutoff = .05;
    dim = 0;
    if min(abs(x-n)) < cutoff
        dim = dim + 1;
    end
    if min(abs(y-n)) < cutoff
        dim = dim + 1;
    end
    if min(abs(z-n)) < cutoff
        dim = dim + 1;
    end
    if dim > 1
        magS = 1;
    else
        magS = 0;
    end
```

dOut = c*U(u); % distance from pinhole
delV = 1/3*delx*delx/c^2*((dOut)^3-(dOut-c*stepSize)^3);

% volume containing the rhoL(v) probability density
rhoL(v) = rhoL(v)*delV; % prob. = prob. density * volume
Add = rhoL(v)+rhoL(v-1);
rhoL(v) = Add;
end

testrow(1) = rhoL(1Rh0+1); % get last value of rhoL
end

% intensity(k,:) = testrow;

% Testing validity of volume factor in calculating probability
intensitysum = sum(sum(intensity));
fprintf('Time: %2.1f\t Sum of grid array: %2.3f\n', t, intensitysum)
```
Bibliography


Index

Continuous Laser, 1

Dipole Radiation, 4, 16
Doppler shift, 17

Liénart-Wiechert Potentials, 10
Lorenz Force Law, 3

Pulsed Laser, 1

Retarded Time, 10
Ryan Sandberg, 13

Surendra Singh, 6

Virtual Camera, 13