Controlling Phase Mismatch in Second Harmonic Generation with Counterpropagating Light

Senior Thesis

Richard S. Camuccio

Advisor: Amy L. Lytle

Department of Physics & Astronomy

Franklin & Marshall College, Lancaster, PA

25 April 2016

Contents

1 Introduction

One of the primary applications of the field of nonlinear optics is to produce different frequencies of light. Given any wave of light, the frequency of the wave can be transformed into another frequency through nonlinear optical processes. One of the most prevalent and difficult challenges to overcome in nonlinear optics is to produce a bright and coherent harmonic signal. The brightness of the signal can be increased by involving higher-powered light sources. The coherence of the signal, however, is difficult to build up, and the phases of the fundamental and harmonic signals tend to destructively interfere within the nonlinear medium before a coherent signal is achieved. The nonlinear medium that produces optical harmonics can be engineered to correct for the phase mismatch. However, in the case where the material itself is being investigated (like in the field of biological microscopy, which may use nonlinear optics as a tool) [1], manipulating the material would damage the properties being investigated. The goal of this project was to investigate our understanding of phase mismatch in second harmonic generation (SHG), a nonlinear optical process. Using a novel technique called "all-optical quasi-phase matching" (AO QPM), the phase mismatch between the fundamental and harmonic beams can be corrected for using interfering light waves, therefore bypassing the need for engineering the material to correct the phase mismatch between the fundamental and harmonic signals. We present a year's worth of scientific investigation into controlling phase mismatch in SHG using the AO QPM technique.

This thesis is broken up into a few primary sections. Section 2 (Theory) details the background information regarding nonlinear optics, SHG, the nature of phase mismatch, and various techniques of phase matching. Section 3 (Experiment) discusses our procedure for collecting data in order to understand the properties of phase matching, and also describes the experimental setup we used in the investigation. Section 4 (Results and Analysis) discusses several important, post-experimental details: the correlation between the power ratio of two counterpropagating beams and the resulting amplitude of the second harmonic sig-

nal; Fourier analysis that was conducted on the signal to better characterize and model it in comparison with a simulated signal. Section 5 (Conclusion) describes the next planned phase of the project, and what can be improved for better data and analysis.

2 Theory

2.1 Second Harmonic Generation

We begin our discussion with the theory behind nonlinear optics. Imagine that an oscillating, time-dependent electric field, $\vec{E}(r,t)$, is applied to a neutral atom. Through the interaction between the field and the atom, the electrons and nucleus respond by becoming polarized. In the case of a weak field, the response is a linear function that is proportional to the strength and direction of the electric field. It can be expressed as

$$P = \varepsilon_0 \chi^{(1)} E, \tag{1}$$

where P is the polarization response, ε_0 is the permittivity of free space (8.8542×10^{-12} $\mathrm{C}^2/\mathrm{Nm}^2$), $\chi^{(1)}$ is the linear electric susceptibility, and E is the magnitude of the electric field [2]. The linear electric susceptibility is a property of the material, and it is related to the material's index of refraction via

$$\chi^{(1)} = n^2 - 1. \tag{2}$$

where n is the material's index of refraction [3]. Imagine that the electrons and nucleus are attached to each other by a spring that abides by Hooke's Law [4]. The oscillating electric field polarizes the atom, and the atom begins to undergo simple harmonic motion with the same amplitude and frequency as the external field. Given a weak field, this linear behavior is a sufficient approximation [5] [4].

Like the behavior of a Hooke's Law spring, the linear polarization can quickly become an inaccurate description of motion as the external force applied is increased. As the force increases, the response of the spring becomes increasingly nonlinear, thus deviating from

the linear approximation of Hooke's Law [5] [4]. In the case of the spring, it is usually irrevocably stretched and thus damaged. The same deviation from linearity applies to the polarization response of the atom in response to an external force. Given a strong (10^{10} V/m) electric field, which exceeds the interatomic field strength of the Coulomb potential, the linear polarization response must be modified by a power series [6] [4]. The power series is

$$P = \varepsilon_0 \left(\chi^{(1)} E + \chi^{(2)} E^2 + \chi^{(3)} E^3 + \cdots \right). \tag{3}$$

Equation 3 is a series expansion of Equation 1, which allows us to include higher order terms to describe the polarization response as it deviates from the linear function. Assuming that we have a single, sinusoidal electric field, expressed as

$$E(t) = E_0 \sin(\omega t), \tag{4}$$

we can substitute it into our power series [5]. Our series then becomes

$$P = \varepsilon_0 \left(\chi^{(1)} E_0 \sin(\omega t) + \chi^{(2)} E_0^2 \sin^2(\omega t) + \chi^{(3)} E_0^3 \sin^3(\omega t) + \cdots \right), \tag{5}$$

of which the first two terms can be rewritten as

$$P = \varepsilon_0 \left[\chi^{(1)} E_0 \sin(\omega t) + \frac{1}{2} \chi^{(2)} E_0^2 \left(1 - \cos(2\omega t) \right) + \cdots \right]. \tag{6}$$

where the first term in the series is the linear polarization response of the atom to the sinusoidal field, and the second term shown is the first correction introducing optical harmonics related to the polarization response [5]. The higher harmonic terms are paired with increasingly-smaller susceptibilities. For this reason, the higher harmonics are not typically

noticeable in natural optical processes. Given a weak field, the higher terms become negligible, and the expression reduces to Equation 1. Stronger fields are necessary in order to produce bright nonlinear fields. By using a stronger fundamental field, we permit the possibility of producing several different frequencies of light. One of the processes that has the possibility of occurring is SHG.

The second term in the series describes several different types of nonlinear processes. These phenomena are listed in Table 1 [3].

Table 1: Known phenomena associated with second-order nonlinear processes. Adapted from Table 15.1 of Robert Guenther's *Modern Optics*.

Incident Field Frequency	Polarization Frequency	Process
ω_1, ω_2	$\omega_1 = \omega_1 + \omega_2$	Sum Frequency Mixing
ω_1	$\omega = 2\omega$	**Second Harmonic Generation**
ω_1, 0	$\omega = \omega_1$	Pockels Effect
ω_1, ω_2	$\omega = \omega_1$ - ω_2	Difference Frequency Mixing
ω_1	$\omega_1 = 0$	Inverse Electrooptic Effect, DC Rectification

The part of the second term in the series that describes frequency doubling, otherwise known as SHG, involves the cosine term [7]. The argument of the cosine term hints at a frequency that can be produced, which is double the fundamental frequency. After all, SHG is a quantum mechanical process, involving two incident photons, each with an energy $\hbar\omega$. There is a slight chance, given the two interacting photons, that both photons will be destroyed and a single photon will be created with an energy twice that of the original photons, or $2\hbar\omega$ (thus obeying conservation laws) [2] [1]. Since the probability of producing SHG light is small, a strong field is necessary to increase the intensity of the SHG light that happens to be produced, thus making it a more probable event (and, therefore, more noticeable).

Producing a strong field is relatively easy, thanks to the invention of the laser [8]. Lasers are easily tunable, and can yield a narrow band of frequencies that make the beam easier to control and characterize. Furthermore, the laser beam has a narrow cross-section, which

means that the intensity of the laser is focused to a small point. Whatever amount of power that goes into producing the laser is maximized by narrowing the beam cross-section in space. Finally, the laser can also be emitted in periodic bursts, which localize the intensity of the beam in time. Given a focused, pulsed laser, the intensity can be maximized via spatial and temporal localization, making the laser an ideal piece of technology to create noticeable second harmonic events.

However, another ingredient necessary for producing a noticeable SHG (or any nonlinear) signal is coherence. Producing a strong field that interacts with a medium composed of, say, several atoms, comes with a host of other problems. One of the major problems that inhibits the coherent production of SHG light is that due to dispersion, resulting in the mismatching of phase between the fundamental and harmonic signals. The problem of phase mismatch shall be discussed in the following subsection.

2.2 Phase Mismatch

How do we produce a coherent SHG signal? If we consider the example of a single atom responding to an oscillating electric field, there is a chance that the response will be SHG. For a noticeable SHG signal to be produced from a medium, we require several atoms responding to the fundamental field in a similar way. More specifically, all of the responses need to be in phase with each other. In the case of complete phase matching conditions, the total radiated field produced by the atoms is simply the sum of all of their individual contributions – complete constructive interference [1].

However, the conditions for complete phase matching are hard-sought. Dispersion is the main obstacle preventing coherent SHG signals from being produced easily. All materials have an index of refraction, n, which can be viewed as the frequency-dependent response of the material to incident electromagnetic radiation [1]. The relationship between the index of refraction of the fundamental frequency with respect to that of the harmonic can be

expressed as

$$n(\omega) < n(2\omega), \tag{7}$$

where $n(\omega)$ is the index of refraction for a material when responding to the fundamental wave, which has a frequency ω; $n(2\omega)$ is the index of refraction when the material responds to the second harmonic light that is radiated by the responding atoms, which has a frequency 2ω [1].

The frequency-dependent responses of both the fundamental and harmonic fields are destined to naturally fall out of phase with each other, at least periodically, since they respond differently to the nonlinear medium. A phase mismatch between the wave vectors thus becomes an intrinsic problem of producing coherent SHG light. The wave vector, $\vec{k}$, of any wave, in this case an electromagnetic wave, defines the direction of travel. The magnitude of $\vec{k}$ is $\frac{2\pi}{\lambda}$, where λ is the wavelength of light. Phase mismatch can be defined as the difference between the harmonic wave vector, $\vec{k}_{2\omega}$, and the fundamental wave vector, $\vec{k}_{\omega}$, as

$$\Delta\vec{k} = \vec{k}_{2\omega} - 2\vec{k}_{\omega}, \tag{8}$$

where $\Delta\vec{k}$ is the difference in wave vectors, $\vec{k}_{2\omega}$ is the wave number of the SHG light, and $\vec{k}_{\omega}$ is the wave number of the fundamental light source [1].

In the ideal case where we have a coherent SHG signal, the difference between the wave vectors is minimized ($\Delta k = 0$) [1]. In most cases, when there is a difference in wave vectors, the interacting fields periodically go in and out of phase with each other. This corresponds to constructive and destructive interference that periodically occurs over the length of the medium [1]. We can define a coherence length as the distance over which the fundamental and harmonic waves accumulate a phase difference. The phase difference is equal to π, and

the interaction, or coherence, length can be defined mathematically as

$$L_c = \frac{\pi}{\Delta k}, \tag{9}$$

where L_c is the coherence length and Δk is the difference in the fundamental and harmonic wave vectors [1]. Figure 1 demonstrates true phase matching conditions in comparison to the typical condition where the phases periodically build up and decay over several coherence lengths.

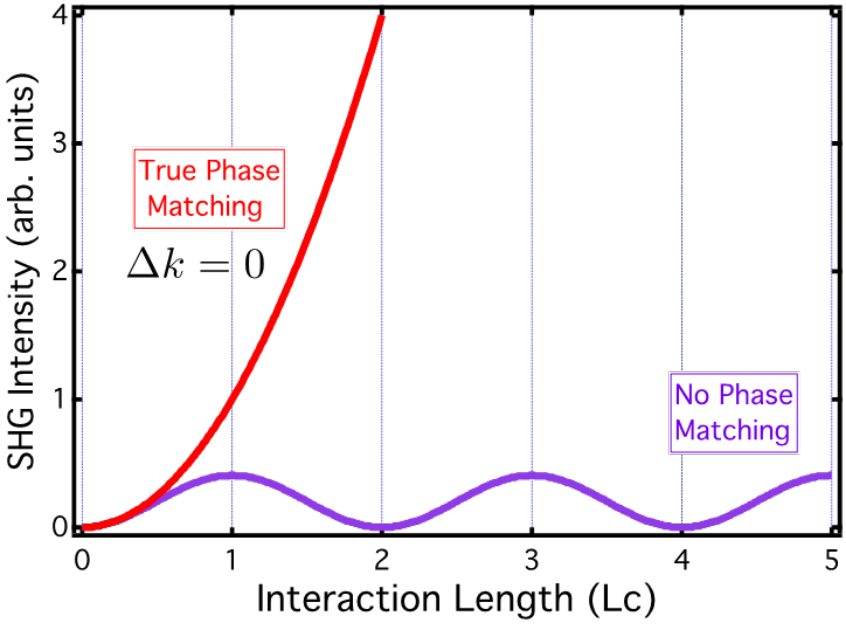

Figure 1: The intensity versus interaction (coherence) length of SHG light. With true phase matching conditions, the intensity increases rapidly. Otherwise, the phases of both the fundamental and harmonic signals will at least partially go out of phase with each other, preventing coherence from being built up. Courtesy A. Lytle.

By using the equation describing a wave vector, $k = \frac{2\pi n}{\lambda}$, where n is the index of refraction of the medium and λ is the wavelength of light in vacuo, and substituting it into our expression for phase mismatch, we get

$$\Delta\vec{k} = \frac{2\pi n(2\omega)}{\lambda_{2\omega}} - \frac{4\pi n(\omega)}{\lambda_{\omega}}, \tag{10}$$

where $n(2\omega)$ is the index of medium of the material responding to second harmonic light, $n(\omega)$ is the index of refraction for the fundamental signal, $\lambda_{2\omega}$ is the vacuum wavelength of the second harmonic signal, and λ_{ω} is the vacuum wavelength of the fundamental signal.

Taking it a step further and noticing that, since we are describing the generation of second harmonic waves, $\lambda_{\omega} = 2\lambda_{2\omega}$, we get:

$$\Delta\vec{k} = \frac{4\pi}{\lambda_{\omega}}\Big(n(2\omega) - n(\omega)\Big) \tag{11}$$

The phase matching condition is dependent upon the difference in the index of refraction both in response to the fundamental light and the second harmonic light. If both indices are equal, the phase mismatch becomes zero, and the waves are both in phase.

2.2.1 Birefringent Phase Matching

If one is not concerned with the properties of the material being used during SHG, then we can take advantage of the index-dependency of the phase matching conditions. There is a special class of material that exhibits two orientation-dependent indices of refraction. Birefringent crystals can be used to exactly match the phases of the fundamental and harmonic signals. The coherence of the SHG signal can be built up exactly in accordance to true phase matching, since the two indices of refraction cause the difference between the phase vectors to become zero [1]. Figure 2 demonstrates an identical situation as Figure 1.

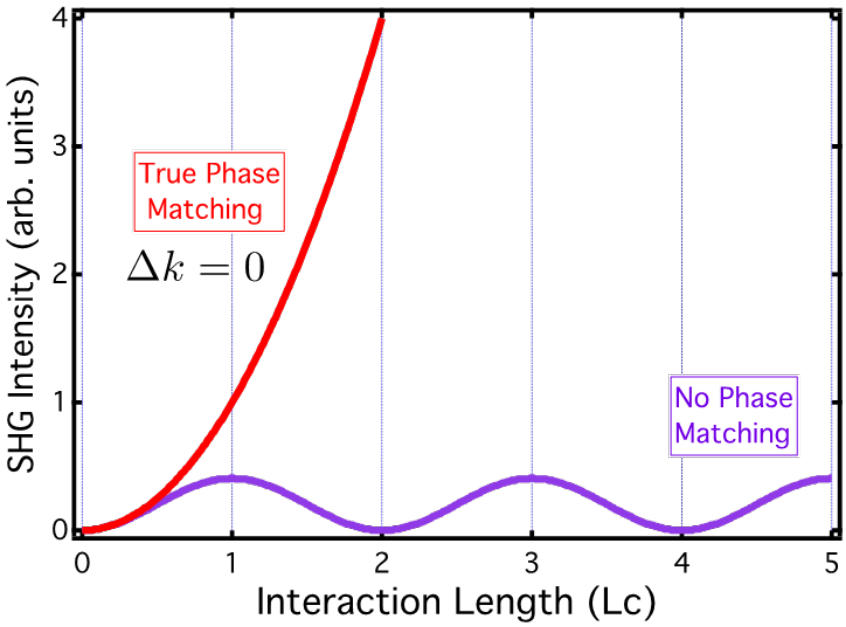

Figure 2: The intensity versus coherence length of an SHG signal. Using birefringent media, the phase matching conditions are optimized so that $\Delta k = 0$, producing an identical curve to true phase matching. Courtesy A. Lytle.

2.3 Quasi-Phase Matching

2.3.1 Periodically-Poled Quasi-Phase Matching

Quasi-phase matching (QPM) is another way of producing coherent second harmonic light. The idea behind QPM is to suppress the destructive interference zones where the phases begin to mismatch. Over several coherence lengths, the intensity can be built up in stages. One method of QPM is called periodically-poled quasi-phase matching (PP QPM). The material is layered in such a way as to produce a flip in the phase of the harmonic light, which is achieved by reversing the sign of the nonlinearity of the material. This, in turn, causes the coherence to be built up over successive interaction lengths [1]. A plot of PP QPM is demonstrated in Figure 3, and is compared to a true phase matching situation, as well as when no phase matching has been achieved.

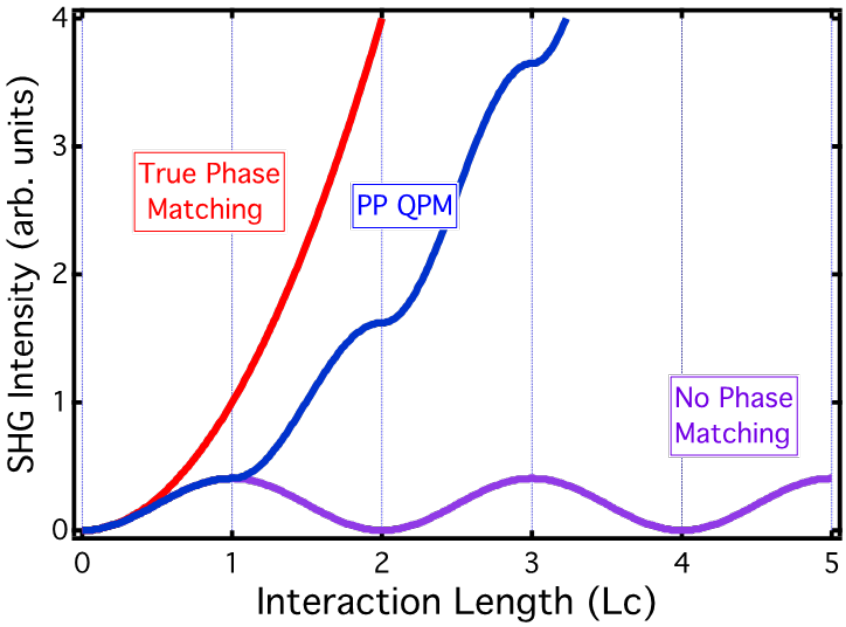

Figure 3: Demonstrating PP QPM in comparison to true phase matching conditions (also the case of using birefringent material) and no phase matching. Courtesy A. Lytle.

2.3.2 All-Optical Quasi-Phase Matching

It has been demonstrated that AO QPM can be achieved without the need to manipulate the medium in which the harmonic light is being generated, both for higher-harmonic generation [9], as well as in theory for SHG [10]. In practice, AO QPM has been demonstrated using counter-propagating beams [11]. If one takes a one-dimensional, forward-propagating electric field, and has it interact with an identical backward-propagating electric field, the phase of the total field can be modulated to periodically build up the coherence of second harmonic light. This is akin to the periodic buildup via PP QPM, but without the need for changing the medium in which the light is being generated. A standing wave can be phase-modulated to prevent the decoherence from occurring [1]. Figure 4 demonstrates the suppression of the destructive interference zones due to pulsed, counter-propagating light.

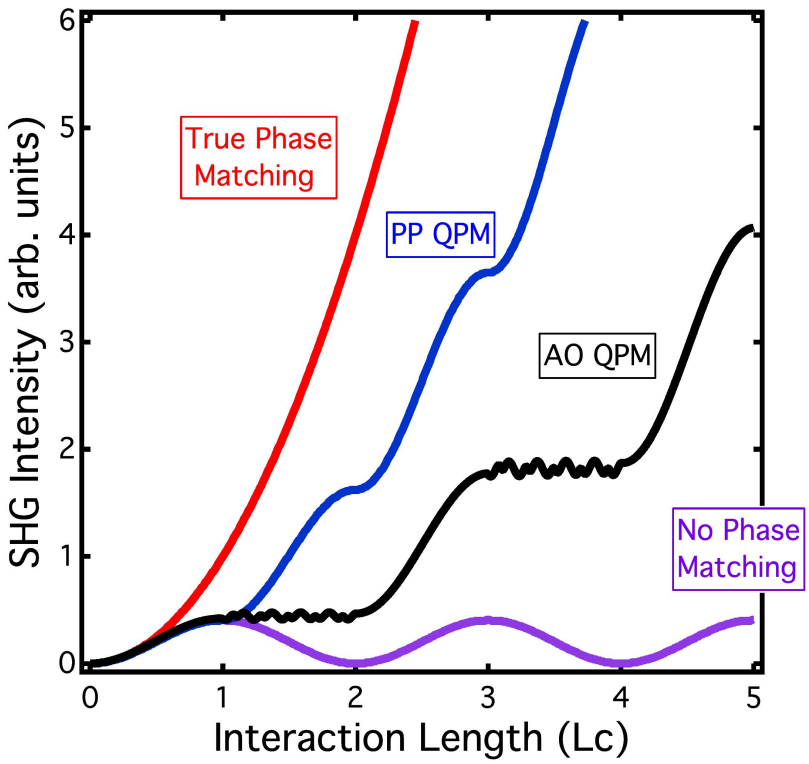

Figure 4: Demonstrating AO QPM in comparison to PP QPM and true phase matching. The regions that lose coherence are sustained using a modulated, counter-propagating laser pulse train that disrupts the regions and allow the overall buildup of coherence. Courtesy A. Lytle.

In order to demonstrate the idea of AO QPM further, we will derive a mathematical expression of the magnitude and phase of the total wave. We begin by considering the equation for a forward propagating, one-dimensional plane wave,

$$E_f(z,t) = E_f e^{i(kz-\omega t)}, \tag{12}$$

as well as the equation for a backward propagating, one-dimensional plane wave,

$$E_b(z,t) = E_b e^{i(kz+\omega t)}, \tag{13}$$

where k is the wave vector, z is the position in the plane, ω is the angular frequency, and t is time.

By taking the sum of the two waves, we get an equation that describes a standing wave. We consider the sum of a forward and backward propagating electric field, E_f and E_b, respectively. The sum can be expressed as the following:

$$E_{tot}(z,t) = E_f(z,t) + E_b(z,t) \tag{14}$$

$$E_{tot}(z,t) = E_f e^{i(kz-\omega t)} + E_b e^{i(kz+\omega t)}. \tag{15}$$

Equation 15 describes the total field of two counterpropagating electric fields. By factoring out the first exponential term, and in turn by splitting up the arguments within all of the exponential terms, we can cancel out some equivalent terms and rewrite the sum:

$$E_{tot}(z,t) = E_f e^{ikz} e^{-i\omega t} + E_b e^{ikz} e^{i\omega t} \tag{16}$$

$$E_{tot}(z,t) = \left[E_f e^{-i\omega t} + E_b e^{i\omega t} \right] e^{ikz} \tag{17}$$

$$E_{tot}(z,t) = \left[E_f + E_b e^{2i\omega t} \right] e^{i(kz-\omega t)}. \tag{18}$$

Equation 18, however, can be simplified further. If we treat the argument within the brackets as the magnitude, E_T, of the total field, the equation can be written as

$$E_{tot}(z,t) = E_T e^{i(kz-\omega t)} \tag{19}$$

where we have written the total field as an amplitude and a forward-propagating field. For the remainder of this discussion, we will look at just the amplitude to learn more about

the interacting fields. We note that the amplitude, E_T, is a complex number of the form $z = e^{i\theta}$:

$$E_T = E_f + E_b e^{2i\omega t}. \tag{20}$$

We are interested in rewriting this in the form

$$E_T = E_A e^{i\Phi}, \tag{21}$$

since that would allow us to use Euler's Identity to retrieve expressions for the magnitude and phase of the total field's amplitude. We recall that Euler's Identity states

$$z = re^{i\theta} = r\cos(\theta) + ri\sin(\theta), \tag{22}$$

where z is a complex function that can be written as a sum of sines and cosines, r is some magnitude, and θ is some phase. The magnitude and phase can be written, respectively, as the following:

$$r = \sqrt{Re[z]^2 + Im[z]^2} \tag{23}$$

$$\theta = \arctan\left(\frac{Im[z]}{Re[z]}\right). \tag{24}$$

By writing out the magnitude and phase using real and imaginary parts, we can write expressions for the magnitude and phase of the total field. The magnitude is

$$E_T = E_f\Big(1 + r^2 + 2r\cos(2\omega t)\Big)^{1/2}, \tag{25}$$

and the phase term is

$$\Phi_T = \arctan\left(\frac{r\sin(2\omega t)}{1 + r cos(2\omega t)}\right), \tag{26}$$

where r is defined as E_b/E_f, the ratio of the backward-propagating to the forward-propagating field magnitudes [1] [11]. It is here we see mathematically what we are trying to understand in relation to experiment. By changing the ratio of the fields, for instance, we seek to better understand the corresponding change in total phase. It has been shown through simulation that, as the ratio of the fields increases past unity, a double-peaking pattern emerges when plotting Φ_T as a function of position z, which can be seen in Figure 5 [1].

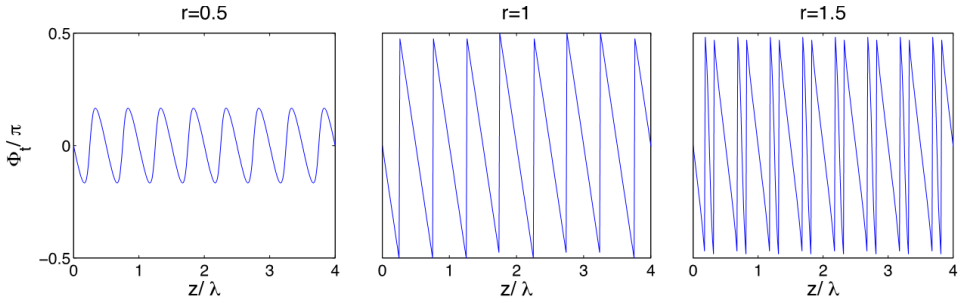

Figure 5: Plots of Φ_T as a function of z for beam ratios of $r = 0.5$, $r = 1$, and $r = 1.5$, respectively. Courtesy of colloquium presentation by A. Lytle.

3 Experiment

3.1 Experimental Setup

The format of our experimental apparatus is shown in Figure 6 [1].

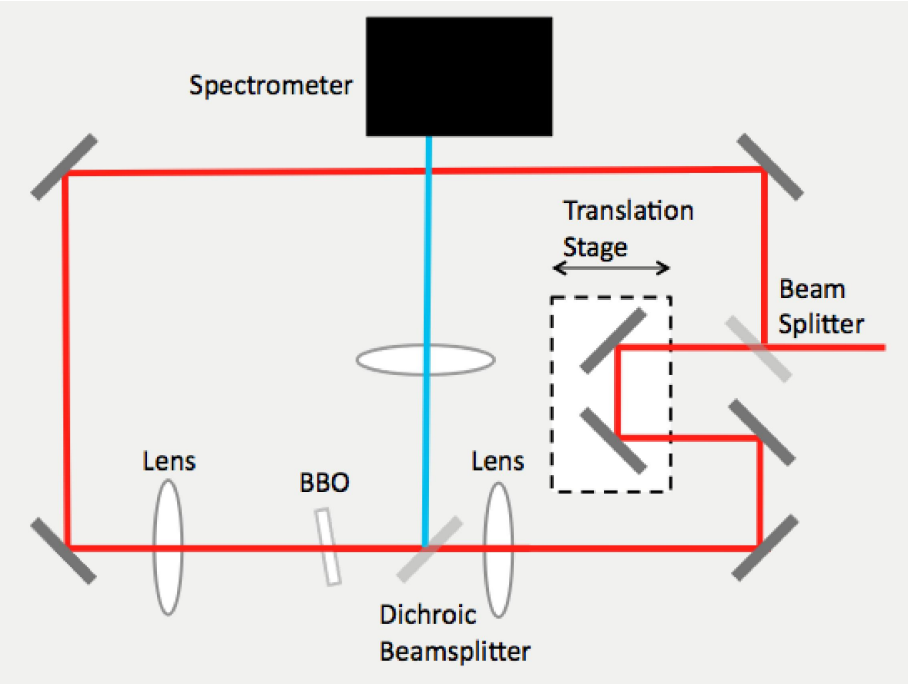

Figure 6: Our setup for the counterpropagating beam experiments.

We use a narrow-band Ti:sapphire laser that oscillates on the order of 10 femtoseconds. The laser we use is spatially- and temporally-localized, making it sufficient to produce intensities needed for SHG. The Ti:sapphire laser hits a beam splitter, where one component continues to move forward, while another component splits off to the right. The beam component that continues forward hits a set of mirrors on a translation stage, which is controlled by a ThorLabs DRV517 Piezo actuator, controlled by a strain-gauge feedback system. The purpose of the translation stage is to change the length of the counter-propagating beam, which allows us to control the phase matching conditions.

At the same time, the forward-propagating beam is sent around the perimeter of the setup

via angled mirrors, through a lens, and into the beta-Barium Borate (BBO) crystal. BBO is a type of crystal that can produce second-harmonic light from its nonlinearity and symmetrical properties. We used a 30-, 100-, and 500-μm BBO in our various experiments. The two beams interfere with each other and produce standing waves within the crystal. The forward SHG beam is then sent into a spectrometer after interacting with a dichroic beamsplitter (which reflects the SHG beam and transmits the forward-propagating fundamental beam). Each adjustment in length of the forward-propagating beam produces a slightly different view of the interference, and each view of the pattern is read by the spectrometer and recorded on LabVIEW software.

3.2 Procedure

The goal of the experiment was to understand the dependence of the SHG signal amplitude on the forward- and counter-propagating beam ratios. By introducing a variable, neutral density filter in either the forward- and counter-propagating beams, we could modulate the power ratio between the beams and observe the change in the SHG signal. Post-experimental analysis involved these data from the amplitude modulation runs. The SHG signal amplitude is directly proportional to the beam power ratio, but the question remains as to how they are proportional to each other. Our hypothesis is that the proportionality is either linear or in accordance with a second-degree polynomial.

4 Results and Analysis

4.1 Primary Results

A plot of the experimental results can be seen in Figure 7. The plots on the left display the second harmonic signal that was generated, the upper plot showing the signal without normalization, and the lower plot showing the signal with normalization. The x-axis is the position (in μm) through the BBO crystal. The y-axis is the wavelength (in nm) as read by the spectrometer. The z-axis is the intensity of the second harmonic signal, at a particular position in the crystal and at a particular wavelength. We normalized the signal by taking the spectrum of the fundamental beam (without the counter-propagating beam, or WOCP), subtracting off an average noise value, and dividing the signal by the noise-subtracted spectrum. The WOCP spectrum is shown in the upper right plot. The bottom right plot shows the second harmonic signal at a particular wavelength. The one specifically shown corresponds to the dashed line in the two plots on the left side of Figure 7.

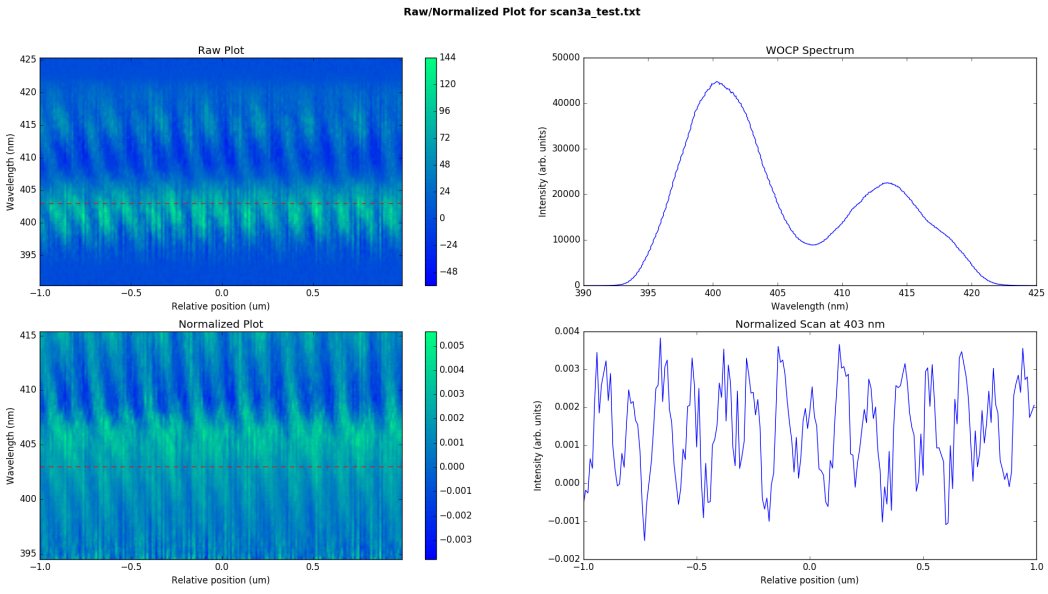

Figure 7: This is an example plot produced after running the Normalization Procedure on scan data. The red dashed line indicates the location where the spectrum is being analyzed, as seen on the bottom right portion of the plot.

The narrow regions of high and low values in the signal plots are what we believe to be the regions of suppression and enhancement due to the counter-propagating beams. We are zoomed-in on one of the regions in Figure 4 where the counter-propagating beam is suppressing the destructive interference zone along each coherence length.

Reducing the noise is a problem that must be addressed in the future if better data is to be acquired. The noise floor seems to be currently irreducible since the source of the noise is primarily mechanical from Hackman and the surrounding environment. If the noise were easier to characterize, it would be beneficial to make a sinusoidal fit to the second harmonic signal, in order to produce residuals for characterizing the noise and removing it.

The code used to produce these plots was written in Python, using the Numpy and Matplotlib packages. It is designed to run with a minimal amount of file dependencies. There are a few modules that need to be included with the main code structure (aside from the data files that are used as input parameters), but otherwise the code is primarily condensed into one file. When running the code from the terminal, the user is immediately prompted to specify two input files. The LabVIEW code tasked with collecting and organizing the experimental data placed the relevant scan information into two files. The "a" file is the first required file, and the "b" file is the second one. The user must specify the spectrum wavelength (in units of nanometers), which is where the bottom right signal plot is produced. In Figure 7, the signal is plotted at a wavelength of 403 nm.

The next input required is a smoothing factor, which is proportional to the how many vertical pixels are averaged together at once during the plotting routine. The basic principle of the smoothing algorithm is to specify a number, which we will denote as the so-called *smoothing factor*, and average that many values in a row. For example, let's say we have three values in a list: [3, 11, 7]. Our smoothing factor will be 3. For simplicity's sake, we will ignore the bordering values and focus on the center value, which is also the philosophy adopted in the code itself. The result of our specifications would be a list that reads: [3, 7, 7]. Notice that

the first and last values in the list are the same as they were before the smoothing took place. The center value is the average of the three values. This is precisely what occurs in the smoothing algorithm for the intensity values read from the input arrays. The example plot in Figure 7 uses a smoothing factor of 11, corresponding to 11 data values average at a time. Next, the code specifies the range of wavelengths that are plotted, in this case the range for the top left plot being from 390 nm to 425 nm. The user is asked again for another smoothing factor for the normalized plot (typically the same), and the wavelength range for the normalized plot (in this case, the plot is from 394 nm to 415 nm).

4.2 Correlation Between Amplitude and Beam Power Ratio

One of the aims of collecting data was to understand the correlation between the amplitude of the second harmonic signal and the power ratio of the forward- and counter-propagating beams. Figure 8 shows data taken between July 6, 2015 and July 9, 2015 of the peak-to-peak amplitude of the second harmonic signal as a function of beam power ratio. The errors are the standard deviations of each measurement. The errors were computed on Microsoft Excel, and the measurements of the peak-to-peak amplitudes were done using Mathematica. The data in Figure 8 is at a wavelength of 403 nm.

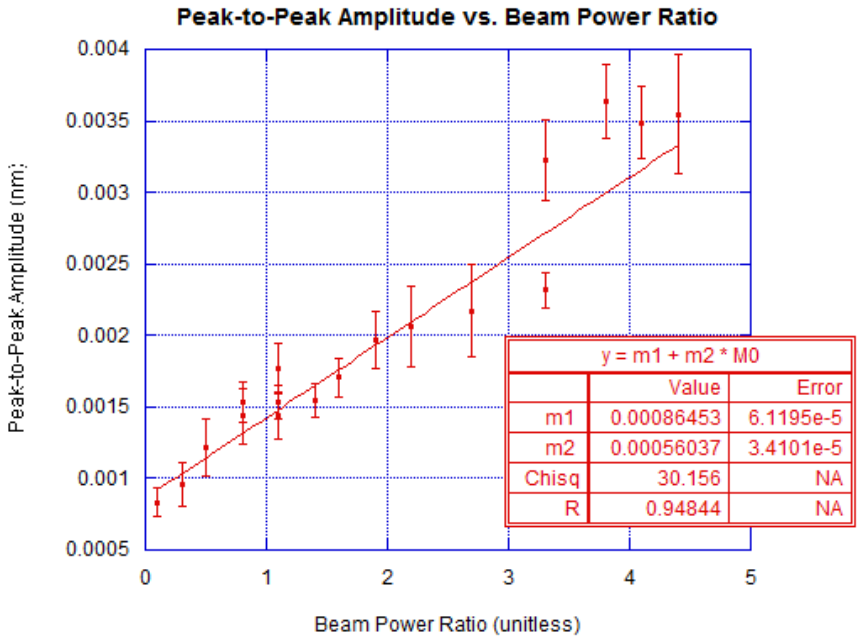

Figure 8: A plot of the peak-to-peak amplitude of the second harmonic signal as a function of beam power ratio, using data acquired between 7/6/2015 and 7/9/2015, at the wavelength of 403 nm. A weighted linear fit has been added to the data. The errors are the standard deviations of each measurement, computed on Microsoft Excel. Plot and linear fit were both done using KaleidaGraph.

Both Python and Mathematica proved incapable of applying a sinusoidal fit to the data. Therefore, these measurements were conducted by hand. Naturally, producing more than a handful of plots like this is tedious, so the next major step in this part of the project involves the creation of an automated fitting procedure. There is reason to suspect that the noise prevalent in the data is responsible for preventing an effective fit. Noise removal, either post-experiment or reducing it in the experimental setup, is imperative if better correlation plots are to be produced.

4.3 Fourier Analysis of Interference Signal

Using simulated data of the second harmonic signal, we conducted Fourier analysis in order to better understand the frequencies of the signal. The simulation is shown in Figure 9. It

simulated the second harmonic signal resulting from counter-propagating electric fields in a BBO crystal.

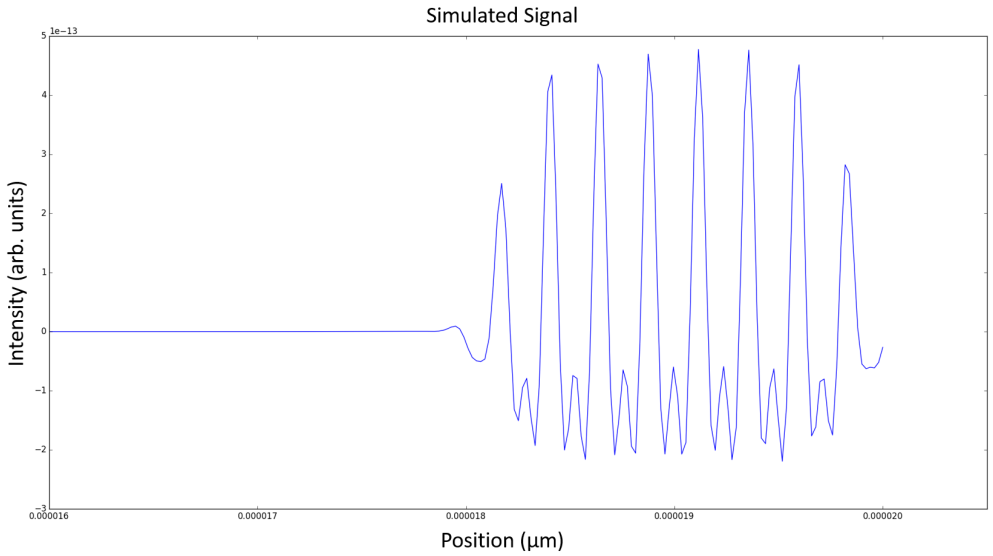

Figure 9: This is the simulated signal, which is the result of modeling two counterpropagating electric fields within the nonlinear medium of beta-Barium Borate. The plot was produced using Python.

The Fourier analysis was accomplished using Python's Fast-Fourier Transform (FFT) function. The result of running an FFT on the simulation is shown in Figure 10. There are two frequencies in the signal, and those are observed in the FFT plot, to the left and the right of the origin. The small bump in the center of the plot is the DC peak, which has been minimized.

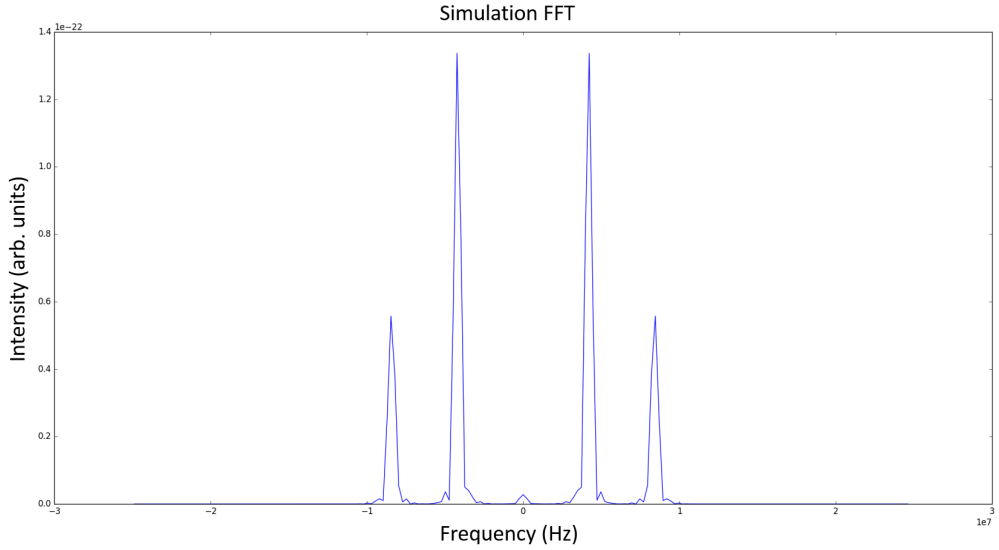

Figure 10: This is the modified Fast-Fourier Transform of the simulated data. Note the reduced DC bump at 0 Hz, a result of the offset correction in the signal, and the flat baseline which indicates that the data are simulated. There are two contributing frequencies, which is consistent with what the simulation input parameters specified.

The next phase of this part of the project involves applying the FFT routine to experimental data. It would be possible to understand the contributing frequencies in the second harmonic signal, and it would be potentially useful to understand the noise that is preventing a cleaner signal from being produced.

5 Conclusion

5.1 Future Work

5.1.1 Continuation of Data Analysis

The main accomplishments of this project fall into two primary categories: acquiring experimental data and constructing the basic software framework for analysis. The experimental data showed many promising features that are similar to, but do not exactly match, the simulated data of two counter-propagating fields in the production of second harmonic light. We noticed many interested features that we would like to further characterize, and the first step was to begin investigating the correlation between the beam power ratio of the counter-propagating fields and the resulting amplitude. It is not yet understood whether the proportionality is linear or another dependence (like a second-order polynomial). Cleaner data will help constrain the correlation between the two variables. Continuing the FFT procedures will also be a next immediate step, as that will help us understand the relationship between the experimental data and the simulation.

5.1.2 Noise Reduction

Many limitations imposed on our understanding of the experimental data have to do with the noise inherent in the signal. The experimental setup itself will have to be further stabilized. The building in which the lab exists is not ideal for experimental nonlinear optics, and the countless sources of mechanical noise make it nearly impossible to characterize and remove the noise on the analysis side. Therefore, the primary area of noise reduction will be on the side of the experimental setup. For instance, we determined after taking a significant amount of data, that the Piezo actuator we used was faulty. We have recently taken measures to replace the actuator with a properly-functioning one, which will help reduce errors in the

signal data.

5.1.3 Development of Analysis Software

The analysis software that has been written so far can take data files produced directly from the experimental setup, and produce a series of plots based on a few input parameters. The code structure is straightforward, but not nearly as streamlined and concise as it can be. There are many dependencies in the file structure in the current way it has been designed. Future investigators would find a useful task in making the code more user-friendly, compact, and efficient. It is reasonable to suspect that additional functionality will be added to the code, and some current functionality will be removed or filed away into a tertiary function. Of course, there are many bugs in the code, and these will be worked out over time as well. The current limitations and peculiarities of the code have been well-documented, and future developers will hopefully improve that for which there was simply not enough time.

6 Appendix

6.1 Derivation of the Square of the Total Field

We wanted to see if the simulation results would be different if the full counter-propagating field was written out. We begin with the total electric field equation:

$$E_{tot}(z,t) = E_f e^{i(kz-\omega t)} + E_f e^{-i(kz-\omega t)} + E_b e^{i(kz+\omega t)} + E_b e^{-i(kz-\omega t)}. \tag{27}$$

The polarization of the second harmonic signal is proportional to the square of the total field, so next we square the previous expression and get

$$\begin{aligned} P_{NL}^{(2)} \propto E_T^2 = &\left[E_f e^{i(kz-\omega t)} + E_f e^{-i(kz-\omega t)} + E_b e^{i(kz+\omega t)} + E_b e^{-i(kz+\omega t)}\right] \\ &\times \left[E_f e^{i(kz-\omega t)} + E_f e^{-i(kz-\omega t)} + E_b e^{i(kz+\omega t)} + E_b e^{-i(kz+\omega t)}\right]. \end{aligned} \tag{28}$$

Next we use the FOIL method to write out the expression more completely:

$$\begin{aligned} E_T^2 = &E_f^2 e^{2i(kz-\omega t)} + E_f^2 e^{i(kz-\omega t)-i(kz-\omega t)} + E_f E_b e^{i(kz-\omega t)+i(kz+\omega t)} \\ &+ E_f E_b e^{i(kz-\omega t)-i(kz+\omega t)} + E_f^2 e^{i(kz-\omega t)-i(kz-\omega t)} + E_f^2 e^{-i(kz-\omega t)-i(kz-\omega t)} \\ &+ E_f E_b e^{i(kz+\omega t)-i(kz-\omega t)} + E_f E_b e^{-i(kz-\omega t)-i(kz+\omega t)} + E_f E_b e^{i(kz-\omega t)+i(kz+\omega t)} \\ &+ E_f E_b e^{i(kz+\omega t)-i(kz-\omega t)} + E_b^2 e^{i(kz+\omega t)+i(kz+\omega t)} + E_b^2 e^{i(kz+\omega t)-i(kz+\omega t)} \\ &+ E_f E_b e^{i(kz-\omega t)-i(kz+\omega t)} + E_f E_b e^{-i(kz-\omega t)-i(kz+\omega t)} + E_b^2 e^{i(kz+\omega t)-i(kz+\omega t)} \\ &+ E_b^2 e^{-i(kz+\omega t)-i(kz+\omega t)}. \end{aligned} \tag{29}$$

Some of the exponential terms cancel (the ones associated with four of the squared field terms). Also, we group the squared terms in the beginning and the cross terms at the end. The result of the cancellations and grouping looks like the following:

$$
\begin{aligned}
E_T^2 = {} & E_f^2 + E_f^2 + E_f^2 e^{i(kz-\omega t)+i(kz-\omega t)} + E_f^2 e^{-i(kz-\omega t)-i(kz-\omega t)} + E_b^2 + E_b^2 \\
& + E_b^2 e^{i(kz+\omega t)+i(kz+\omega t)} + E_b^2 e^{-i(kz+\omega t)-i(kz+\omega t)} + E_f E_b e^{i(kz-\omega t)+i(kz+\omega t)} \\
& + E_f E_b e^{i(kz-\omega t)-i(kz+\omega t)} + E_f E_b e^{i(kz+\omega t)-i(kz-\omega t)} + E_f E_b e^{-i(kz-\omega t)-i(kz+\omega t)} \\
& + E_f E_b e^{i(kz-\omega t)+i(kz+\omega t)} + E_f E_b e^{i(kz+\omega t)-i(kz-\omega t)} + E_f E_b e^{i(kz-\omega t)-i(kz+\omega t)} \\
& + E_f E_b e^{-i(kz-\omega t)-i(kz+\omega t)}.
\end{aligned} \tag{30}
$$

The next step involves factoring out the field magnitudes and grouping together the associated terms:

$$
\begin{aligned}
E_T^2 = {} & E_f^2 \Big[1 + 1 + e^{i(kz-\omega t)+i(kz-\omega t)} + e^{-i(kz-\omega t)-i(kz-\omega t)} \Big] \\
& + E_b^2 \Big[1 + 1 + e^{i(kz+\omega t)+i(kz+\omega t)} + e^{-i(kz+\omega t)-i(kz+\omega t)} \Big] \\
& + E_f E_b \Big[e^{i(kz-\omega t)+i(kz+\omega t)} + e^{i(kz-\omega t)-i(kz+\omega t)} + e^{i(kz+\omega t)-i(kz-\omega t)} \\
& + e^{-i(kz-\omega t)-i(kz+\omega t)} + e^{i(kz-\omega t)+i(kz+\omega t)} + e^{i(kz+\omega t)-i(kz-\omega t)} \\
& + e^{i(kz-\omega t)-i(kz+\omega t)} + e^{-i(kz-\omega t)-i(kz+\omega t)} \Big].
\end{aligned} \tag{31}
$$

Next, we separate the kz and ωt terms in the exponentials in order to cancel and simplify the expression further:

$$
\begin{aligned}
E_T^2 = {} & E_f^2 \Big[2 + e^{ikz} e^{-i\omega t} e^{ikz} e^{-i\omega t} + e^{-ikz} e^{i\omega t} e^{-ikz} e^{i\omega t} \Big] \\
& + E_b^2 \Big[2 + e^{ikz} e^{i\omega t} e^{ikz} e^{i\omega t} + e^{-ikz} e^{-i\omega t} e^{-ikz} e^{-i\omega t} \Big] \\
& + E_f E_b \Big[e^{ikz} e^{-i\omega t} e^{ikz} e^{i\omega t} + e^{ikz} e^{-i\omega t} e^{-ikz} e^{-i\omega t} \\
& + e^{ikz} e^{i\omega t} e^{-ikz} e^{i\omega t} + e^{-ikz} e^{i\omega t} e^{-ikz} e^{-i\omega t} \\
& + e^{ikz} e^{-i\omega t} e^{ikz} e^{i\omega t} + e^{ikz} e^{i\omega t} e^{-ikz} e^{i\omega t} \\
& + e^{ikz} e^{-i\omega t} e^{-ikz} e^{-i\omega t} + e^{-ikz} e^{i\omega t} e^{-ikz} e^{-i\omega t} \Big].
\end{aligned} \tag{32}
$$

Many of the exponential terms cancel out which, once the remaining terms are consolidated, results in

$$\begin{aligned} E_T^2 = E_f^2\Big[2 + e^{2i(kz-\omega t)} + e^{-2i(kz-\omega t)}\Big] + E_b^2\Big[2 + e^{2i(kz+\omega t)} + e^{-2i(kz-\omega t)}\Big] \\ + E_f E_b\Big[e^{2ikz} + e^{-2i\omega t} + e^{2i\omega t} + e^{-2ikz} + e^{2ikz} + e^{2i\omega t} + e^{-2i\omega t} + e^{-2ikz}\Big], \end{aligned} \tag{33}$$

which can further be simplified by combining the exponentials in the third term of the equation:

$$\begin{aligned} E_T^2 = E_f^2\Big[2 + e^{2i(kz-\omega t)} + e^{-2i(kz-\omega t)}\Big] + E_b^2\Big[2 + e^{2i(kz+\omega t)} + e^{-2i(kz-\omega t)}\Big] \\ + E_f E_b\Big[2e^{2ikz} + 2e^{-2ikz} + 2e^{2i\omega t} + 2e^{-2i\omega t}\Big]. \end{aligned} \tag{34}$$

Equation 34 is the last step in the derivation. It is the most simplified version of the full field, resulting from the combination of a forward propagating and counterpropagating electric field. The first term in the equation is entirely due to the forward-propagating field. The second term is from the backward-propagating field. The third term arises from the two fields interacting.

6.2 Analysis Code

A majority of the data analysis required for our experiments was accomplished using the Python programming language. The code was designed for a specific task: to break apart the data arrays produced by the LabVIEW code, and create contour plots demonstrating the intensity values over the length of each scan, as well as over all wavelengths of light observed by the spectrometer. Provided this challenge, *analysis.py* was born.

6.2.1 Structure and Functionality

The structure of *analysis.py* is broken up into five separate functions. These functions are listed in Table 2:

Table 2: Analysis Code Functions

Function Number	Function Name
I	Full/ROI Plot
II	Smooth/ROI Plot
III	WOCP Spectrum Analysis
IV	WCP-WOCP Spectrum Analysis
V	Normalization

We will go on to define the functionality of the five parts of *analysis.py*. Since the primary goal of writing this code was to create a normalized plot to be used for all subsequent analysis, the Normalization Function listed in Table 2 has become a separate module. We will proceed by describing the first four functions that are still within *analysis.py*. These are all precursor functions that were utilized in the the module named *normalize.py*. Then we will discuss the functionality of *normalize.py*, including its module and data file dependencies.

The program initialization procedure is straightforward. It clears the terminal and provides text for the user to choose which function to run for data analysis. The program is designed to run through once and then terminate, which can be tedious if the user wishes to run through multiple iterations of data. However, the code does not implement a continuous loop to keep the it running, as that would detract from focusing on building up the code's functionality. If this code winds up being used more extensively in future parts of this project, then it is recommended that someone add a continuous loop to the code. It will make the code run without the user having to tediously restart each time a plot needs to be produced.

Full/ROI Plot Function

The Full/ROI Plot function takes the first output file produced from the LabVIEW code and produces a contour plot of the intensity values for a particular counter-propagating scan,

across the wavelengths observed by the spectrometer. The user can specify a cropped "region of interest" (ROI) to zoom in on.

The user is prompted to specify which file to load. The LabVIEW code produced three file types from a single scan, so the file should be an "a" file (e.g. "scan3a.txt"). The code immediately converts the specified file into an array. The array dimensions [m, n] are measured, and from this we define a scan length. The array is then broken up into three sub-arrays: a 2-dimensional intensity array, a 1-dimensional wavelength array, and a 1-dimensional position array. Then the code transposes the position array so it can slice off the useless data from the end. Slicing is defined as taken the entire data array and cutting off the useless data values that correspond to background noise. The intensity array is transposed, and then the code runs the plt.contourf() function to make a plot of the three arrays. The user is asked to specify the range of wavelengths that should be plotted, and everything outside of this range is removed. The spectrometer we used for the scan has a manufacturer's calibration on how to convert wavelength into a pixel number. This is useful since we need to slice the array at a specific pixel in order to plot over a smaller scale (ROI). The equations used for this are

$$B_0 = A_0 - \frac{A_1^2}{4A_0} \tag{35}$$

$$B_1 = \frac{A_1}{2A_2} \tag{36}$$

where $A_0 = 348.4316445$, $A_1 = 0.069763835$, and $A_2 = 2.98682 \times 10^{-6}$. The code defines these equations and coefficients, and calculates the minimum and maximum pixel numbers to be used as the boundaries for the slice operation.

Next, the intensity and wavelength arrays are transposed in order to slice them at the calculated minimum and maximum pixel numbers. The wavelength array is transposed back

to its original orientation, and then another call is made to the plt.contourf() function.

Smooth/ROI Plot Function

Similarly to the previous function, the Smooth/ROI Plot function produces a nearly-identical output file to the previous function, the only difference being the incorporation of a smoothing function, which produces a smoothed and cropped contour plot.

The part of the function up to the smoothing algorithm is the exact same plotting procedure that is detailed in the previous function. The smoothing algorithm begins with the creation of an empty list to handle some temporary values. The user is asked to specify the smoothing factor. There is a check in place to make sure the user specifies a number that is lower than the length of the array. The program then calculates the edge pixels to ignore. At first glance, this could seem to be a waste since we are essentially ignoring the pixels on the front and rear edges of the plot and not averaging them. However, this number is low enough for our typical specifications to avoid interfering with a majority of the data. First, the program simply places the front edge values into the list, since they are not being averaged. Next, the program runs through the array, slicing it around the values (the length of the slice being equal to the smoothing factor). The pixels are then averaged and placed into the list for temporary storage. Finally, the rear edge values are placed into the list, for the same reason that the front edge values are. Once the values are all averaged and placed into the temporary list, the list is then converted into an array using the very useful Numpy function called *np.asarray()*. The three arrays are plotted with user-specified properties. From this point on, until the end of the function, is an exact copy of the slicing algorithm. The final output is two plots: both smoothed, one of which is a full plot and the other is a sliced plot.

WOCP Spectrum Analysis Function

The WOCP Spectrum Analysis function produces a plot of the intensity versus wavelength spectrum at a specified region of the contour plot, when there is no counterpropagating

beam interfering with the forward propagating beam. The file used for this function is a "b" file.

After the data file is loaded, the data is converted into a Numpy array and transposed in preparation for creating the spectrum. The user is then asked to specify a location on the intensity plot, which is essentially the length of the array (minus the last two parts, which are the wavelengths and positions). There is a check in place to make sure these boundary conditions are met. The x-axis and y-axis is specified based on a part of the array. The spectrum is then plotted with the properties specified. At this point, the function runs through the smoothing algorithm. The algorithm is modified slightly to include the y (intensity) values only. Since x and y are both 1-dimensional, there is no longer a need for nested-for loops. A smoothed plot is then rendered and produced.

WCP-WOCP Spectrum Analysis Function

The WCP-WOCP Spectrum Analysis function produces a nearly-identical output file to the previous function, but with the introduction of a counter-propagating beam to the forward-propagating beam. The file to be used for this function is an "a" file. The main difference between this function and the previous one is that the initial array is not transposed (since "a" and "b" files have different formats). Therefore, there is no need to describe the procedure of this function, since it is virtually identical to the previous one, save the difference in the initial transpose command.

Normalization Function

The Normalization Function, originally a part of the *analysis.py* module, has now grown to become a separate module, *normalize.py*. The function begins by importing several Python modules that are used for calculations and plotting, as well as some of the routines for smoothing and slicing. These routines are separate modules to make the bulk of the code simpler. The only necessity is to run this primary module in a file that also includes these independent, but vital modules.

The next routine asks for the user to specify which "a" and "b" files to input. The code writes the data files to arrays, measures them, slices them into separate variable arrays for handling, and then trims and transposes them to be used for processing and, eventually, plotting. The user is left with three arrays: a position, wavelength, and intensity array. Copies of these arrays are also used. This is to ensure that nothing is mixed up when those files are plotted, and then used again for normalizing and further plotting. I thought it simpler to make a separate batch of arrays for the non-normalized plot, and another batch of identical arrays for the normalized plot.

Next, the code asks the user to specify at which wavelength the code should make an intensity plot slice. The wavelength is then converted to a pixel value. The temporary arrays are passed into the smooth routine module, which smooths the pixels vertically based on a specified smoothing factor. A typical value used for the factor is 11, but this might be specified differently depending on how this code is used in the future. The smoothed arrays are then passed into the slice routine, and sliced at a minimum and maximum wavelength that is specified by the user. This range is typically 390 nm to 425 nm. Finally, the code plots the three arrays using Python's *contourf()* function.

The normalization routine follows the first plotting routine. Basically, the code takes the spectrum without the counter-propagating field, and smooths it by passing it into the spectrum smoothing routine function. Then, it calculates the mean of the noise in the spectrum, and normalizes the intensity array by dividing each row of values by a row of averaged noise values. This effectively normalizes the intensity array. The following occurs, which is identical to the previous handling of the intensity array: it is smoothed, sliced, and then the normalized array is plotted using Python's *contourf()*.

Also included in the code is a plotting feature of both the WOCP spectrum, and the normalized intensity at a particular wavelength. For this last feature, the intensity of the second harmonic signal is plotted as a function of relative position, at a particular wavelength in

the spectrum. The analysis code finishes running by opening two text files and writing the normalized intensity array values in one and the position array values in another. These files can be read into other programs for further analysis.

6.2.2 Simulation FFT Code

One of the last parts of the code is a basic procedure that uses Numpy's Fast Fourier Transform (FFT) function to take a signal and conduct a Fourier transform on it. The FFT function was used in two situations. The first situation was tacked on to the end of the *normalize.py* script, which takes a Fourier transform of the second harmonic signal from experiment, and then plots and writes the data to file. The second situation does this in an independent module called *cprop_simulation.py*, which takes the Fourier transform of a simulated signal, and both plots it and writes the data to file.

7 Acknowledgments

A thesis is not an easy task to complete, therefore many acknowledgments are in order. First and foremost, I want to thank my advisor, Dr. Amy Lytle, for taking me under her wing and teaching an aspiring astronomer the ways of an experimental physicist. I have learned more than any lecture or exam could have possibly taught me, and I am forever grateful for the opportunity she has provided me, and her patient guidance during the entirety of the project.

I want to thank Dr. Etienne Gagnon for developing the code that produced the extremely useful simulations for this project, for teaching me what it is like to be in a collaboration, and for additional support and encouragement along the way.

Thanks to Steve Spadafore for inadvertently teaching me how sensitive the optical equipment is to mechanical noise from the surrounding environment. Also, thanks to Froney for introducing me to KaleidaGraph. It was useful for the weighted fits I did.

I also acknowledge support from the Hackman Summer Scholars Program and the Research Corporation for Science Advancement (CCSA #21084) for funding this research. Undergraduate students of science benefit from learning the methods of scientific research firsthand. I have learned more about science in this project than in any lecture hall or examination.

Finally, thank you to my family: my parents, sister, girlfriend, and friends. Thank you for your encouragement to strive hard in school and learn as much as I can. I am forever grateful for your support and love.

References

[1] A. Lytle, "Understanding and controlling second harmonic generation," *F&M Physics & Astronomy Department Colloquium*, 2014.

[2] R. W. Boyd, *Nonlinear Optics*. San Diego, CA: Academic Press, 2nd ed., 2003.

[3] R. D. Guenther, *Modern Optics*. New York, NY: Wiley, 1990.

[4] G. New, *Introduction to Nonlinear Optics*. Cambridge: Cambridge University Press, 2011.

[5] E. Hecht, *Optics*. Reading, MA: Addison-Wesley, 2002.

[6] M. M. Fejer, "Nonlinear optical frequency conversion," *Physics Today*, 1994.

[7] R. Paschotta, *Frequency Doubling*. RP Photonics Consulting GmbH, http://www.rp-photonics.com/frequency_doubling.html.

[8] P. A. Franken, A. E. Hill, C. W. Peters, and G. Weinreich, "Generation of optical harmonics," *Physical Review Letters*, 1961.

[9] X. Zhang, A. L. Lytle, T. Popmintchev, X. Zhou, H. C. Kapteyn, M. M. Murnane, and O. Cohen

[10] A. Bahabad, O. Cohen, M. M. Murnane, and H. C. Kapteyn, "Quasi-phase-matching and dispersion characterization of harmonic generation in the perturbative regime using counterpropagating beams,"

[11] A. Lytle, R. Camuccio, R. Myer, A. Penfield, and E. Gagnon, "Control of second harmonic generation using counterpropagating light," *Optics Letters*, Manuscript submitted for publication.