NUMERICAL MODEL OF NON-COLLINEAR PARAMETRIC DOWN-CONVERSION

by

Dustin Shipp

A senior thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Bachelor of Science

Department of Physics and Astronomy

Brigham Young University

April 2008

BRIGHAM YOUNG UNIVERSITY

DEPARTMENT APPROVAL

of a senior thesis submitted by

Dustin Shipp

This thesis has been reviewed by the research advisor, research coordinator, and department chair and has been found to be satisfactory.

Date	Michael Ware, Advisor
Date	Eric Hintz, Research Coordinator
Date	Ross Spencer, Chair

ABSTRACT

NUMERICAL MODEL OF NON-COLLINEAR PARAMETRIC DOWN-CONVERSION

Dustin Shipp

Department of Physics and Astronomy

Bachelor of Science

We used a numerical model to study spontaneous parametric down-conversion, a process in which a single photon splits into two "daughter" photons. The model predicts the location of daughter photons based on phase matching conditions. The user can vary parameters such as crystal type, pump and signal photon wavelength, and geometries of the system. Varying these parameters allows properties of down-conversion to be tested and optimized. We experimentally confirmed several features included in the model for Type I down-conversion in a BBO crystal. We found that the diameter of the down-conversion ring is well modeled by our numerical approach. We experimentally measured the total down-conversion output over all angles at a single wavelength to remain roughly constant as the size of the ring varied. Our model does not predict this well due to its incompleteness.

ACKNOWLEDGMENTS

I acknowledge funding support from the BYU College of Physical and Mathematical Sciences. I also acknowledge the assistance of Michael Ware, Jean-Francois Van Huele, Patrick van Langen, and Laura Hurst.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction to Down-conversion

Many nonlinear crystals exhibit a phenomenon known as spontaneous parametric down-conversion (SPDC). In this process, an incoming photon is absorbed by atoms of the crystal. The energy from this photon is re-emitted in a pair of photons [1]. These "daughter" photons have lower frequencies than the original, or "pump" photon. Conservation of energy and Plank's formula $E = \hbar\omega$ requires that the frequencies of the daughter photons must add up to the frequency of the pump. In addition, the emission directions of the daughter photons are governed by conservation of momentum [2]. From these laws, the frequency and location of down-converted photons can be predicted mathematically in a process known as phase matching.

By convention, the daughter photons are given the names "signal" and "idler." By placing detectors at appropriate angles, one can detect the signal and idler photons. Because the photons are always created in pairs, when one of these photons is detected, it signals the existence of the other with absolute certainty. The detection of both signal and idler photons within a specified time window is referred to as a

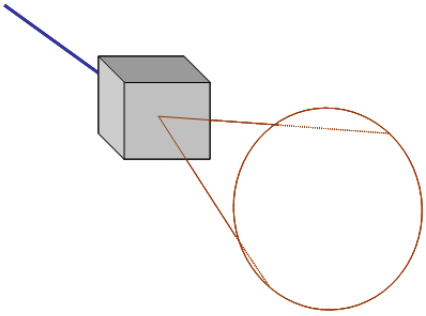

Figure 1.1 Type I parametric down-conversion

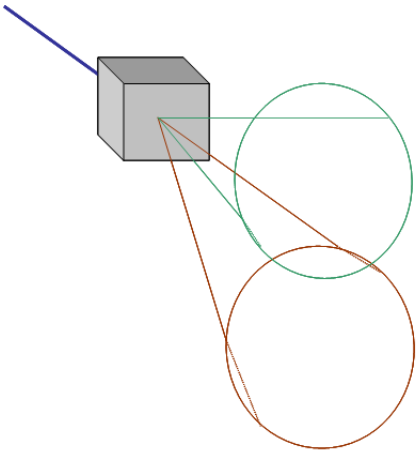

Figure 1.2 Type II parametric down-conversion

"coincidence." [1]

There are two types of SPDC. In Type I down-conversion, both the signal and idler photon are polarized such that they experience the extraordinary index of refraction. If the energy is shared equally between signal and idler photons (i.e. they have the same wavelength), then by $E = cp$, the signal and idler photons must also have the same magnitude of momentum. This means that the photons will propagate along a single cone, 180° apart in azimuthal angle. If the energy is not equally split between the two photons, then the more energetic photon also has more momentum. In order to preserve conservation of momentum, the less energetic photon will propagate at a greater angle away from the pump beam. Thus, its lateral contribution to the momentum of the system can match that of the more energetic photon.

In Type II down-conversion, one photon is polarized along the ordinary axis of the crystal and the other along the extraordinary. This causes daughter photons with degenerate wavelengths to propagate in separate cones. These cones are offset equal distances from the propagation axis of the pump beam [3]. The cones for degenerate SPDC are shown in Figs. 1.1 and 1.2.

Because of the phase matching conditions, the daughter photons are highly correlated in many parameters such as frequency, geometry, polarization, and emission time [4, 5]. Because of this correlation, information about the pump and one of the daughter photons can be used to deduce the properties of the other daughter [6]. Thus, when daughter photons are detected, the timing and location of the detector reveals the existence and properties of the idler photon. This provides a great deal of information about the idler photon without having to observe it.

The correlation properties of daughter photons from SPDC lead to a number of useful applications. The daughter photons can be entangled in a number of properties such as emission time and polarization. This allows the daughter photons to be used to test Bell's inequalities [3]. SPDC also has important applications as a single photon source in quantum cryptography [1]. The signal photon can be detected to herald the arrival of the idler photon, which can then be used for encrypting data in such a way that any attempts to eavesdrop will change the message itself. Those sending and receiving the message can detect this before any information is compromised. Applications in quantum teleportation have also been considered [1]. The entanglement of the daughter photons and the simple process of generating SPDC provide many possibly convenient experiments to test these and other basic laws of quantum information and quantum mechanics [2].

1.2 Computational Modeling

To study the process of SPDC and its daughter photons, researchers must usually know the location, frequency, and polarization of daughter photons. Due to the high correlation between daughter photons and the pump, it is possible to predict the location and properties of daughter photons. Because SPDC is a random process,

only the probabilities can be predicted. Still, with enough photons in the pump beam, these probabilities correspond to observable quantities of photons. This enables researchers to confirm theoretical models with experiments.

The conditions for phase matching are very sensitive to a number of parameters. Changing the type, length, or cut angle of the crystal can change the trajectories of the daughter photons. Altering the frequency, beam waist, or incident angle of the pump beam will also change the properties of down-converted photons. We have developed a computational model to predict the outcome of an experiment with various crystal and pump beam configurations.

Our model uses the phase matching conditions to predict the location of signal and idler photons at a detector. Predictions can be made for various pump wavelengths, crystals, and geometries. Predictions of idler photon detection can be plotted for either specific signal photon wavelengths or over a range of wavelengths, simulating the effect of a bandpass filter. Plots made over a range of wavelengths give an accurate model of what can be seen in SPDC. The relationship between these photons can also be seen in these plots. These results are confirmed by experiment in the Type I case.

Chapter 2

Methods

2.1 Theoretical Model

The core of the computational model is the phase matching condition. After the down-conversion process, the energy and momentum of the pump photon must be conserved in the daughter photons. We approximate the pump beam to be monochromatic. This means that every photon has a clearly defined energy. In this case, the energy conserving phase matching equation is

$$E_{pump} = E_{signal} + E_{idler}. \tag{2.1}$$

By Plank's formula, $E = \hbar\omega$, it follows that

$$\hbar\omega_p = \hbar\omega_s + \hbar\omega_i \tag{2.2}$$

$$\omega_p = \omega_s + \omega_i. \tag{2.3}$$

So the idler frequency is

$$\omega_i = \omega_p - \omega_s \tag{2.4}$$

Conservation of momentum for a simple case is satisfied by

$$\boldsymbol{p}_{pump} = \boldsymbol{p}_{signal} + \boldsymbol{p}_{idler}. \tag{2.5}$$

. Since $\boldsymbol{p} = \hbar\boldsymbol{k}$, where $\mathbf{k}$ is the wave-vector specifying the direction of travel for the wave,

$$\boldsymbol{k}_{pump} = \boldsymbol{k}_{signal} + \boldsymbol{k}_{idler}. \tag{2.6}$$

. Although it is a fairly good approximation that our pump beam is monochromatic, the same approximation cannot be made with its direction. Because of finite size, our laser beam consists of a superposition of plane waves traveling in different directions.

To incorporate the superposition of k vectors into our model, we used the result published by Hong and Mandel [7]. They derive their result by starting with an interaction Hamiltonian for the pump and daughter photons. This result is written for a superposition of plane-wave propagation modes which are then used to find the equations of motion. These are then integrated over the volume of the crystal and over the time photons spend inside it. The result is a detection probability which is dependent on the orientation and wavelengths of the pump, signal, and idler photons, the width of the pump beam, and the properties of the crystal. This result is

$$\Phi = e^{-\frac{1}{2}w_0^2(\Delta k_x^2+\Delta k_y^2)\mathrm{sinc}^2(\frac{1}{2}\Delta \mathrm{k_z L})}, \tag{2.7}$$

where

$$\Delta k_x = 2\pi\left(\frac{n_s\sin(\theta_s)\cos(\phi_s)}{\lambda_s} + \frac{n_i\sin(\theta_i)\cos(\phi_i)}{\lambda_i}\right), \tag{2.8}$$

$$\Delta k_y = 2\pi\left(\frac{n_s\sin(\theta_s)\sin(\phi_s)}{\lambda_s} + \frac{n_i\sin(\theta_i)\sin(\phi_i)}{\lambda_i}\right), \tag{2.9}$$

and

$$\Delta k_z = 2\pi\left(\frac{n_s\cos(\theta_s)}{\lambda_s} + \frac{n_i\cos(\theta_i)}{\lambda_i} - \frac{n_p}{\lambda_p}\right). \tag{2.10}$$

A visual interpretation of Δk_x, Δk_y, and Δk_z is shown in Fig. 2.1. In these equations, w_0 is the width of the pump beam waist, L is the length of the crystal, n is the index of refraction, sinc(x) is defined as $\frac{\sin(x)}{x}$, λ is the wavelength, θ is the angle of the trajectory with respect to the pump beam, ϕ is the azimuthal angle, and p, s, and i are subscripts that refer to the pump, signal, and idler photons, respectively.

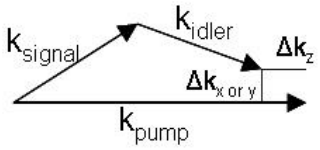

Figure 2.1 Conservation of momentum. Visualization of Δk_x, Δk_y, and Δk_z.

If the pump beam has an infinite radius and the crystal is infinitely long, then the simple approximations hold for conservation of momentum. This phase matching condition reduces to the much simpler Eq. (2.6).

Images are produced by varying the frequency of signal photon and calculating the maximum possible phase match function Φ at each location. This maximum value is related to an integration over all possible idler directions. We then create our images using Φ^2, which is proportional to signal strength corresponding to the experimental observations.

Since SPDC occurs in non-linear crystals, the indexes of refraction can differ for all three photons. The indexes vary depending on the properties of the crystal and the orientation of the pump, signal, and idler beams with respect to the crystal axis. In Type I down-conversion, signal and idler photons both share the same index of refraction. In Type II down-conversion, the signal and idler photons have ordinary and extraordinary indexes of refraction. Since the naming of signal and idler photons is arbitrary, either photon can have either index.

To account for these geometric effects, our model accepts parameters of crystal type, Type I or Type II (if applicable), and angle (both zenith and azimuthal) of the pump beam with respect to the crystal axis. This is generally known from the cut of the crystal. These parameters are used to transform the beam into the crystal frame of reference, where the phase matching conditions are applied. For details on this coordinate transformation, see Ref. [5].

With the user-provided input of crystal type, crystal length, pump beam waist, incident angles of the pump beam, and pump wavelength, our model can create an image of the daughter photons as they would be seen at an image plane. This image can be based on a single signal wavelength or vary over several wavelengths, adding the images on top of each other. In both cases, the idler wavelength is given by Eq. (2.4), rewritten as

$$\lambda_i = \frac{c}{\frac{c}{\lambda_p} - \frac{c}{\lambda_s}} = \frac{1}{\frac{1}{\lambda_p} - \frac{1}{\lambda_s}}, \tag{2.11}$$

where c is the speed of light in a vacuum. The model then tests each point on the grid for the phase matching conditions at each wavelength and plots the probability of photon detection at that point. We ignore pump photons continuing through the crystal in these plots. Some Fortran code of this process is included in Appendix A.1. Plots are included in Section 3.1.

2.2 Experimental Setup

To confirm the model's predictions experimentally, we observed the down-converted photons using a standard CCD camera. The pump laser produced a beam of 351 nm light which we directed through a BBO crystal. The cut of the crystal gave us the capability to observe only the Type I case. The majority of the pump beam passes through the crystal without being down-converted, so we inserted a small beam-stopper to remove it. The down-converted photons propagated at angles allowing them to pass the beam-stopper and continue. The daughter photons were then directed into parallel paths down the table through a lens. We then used band-pass filters to isolate various wavelengths and observe their individual rings with the camera.

Several pictures were taken at various crystal angles. At each angle, half of these

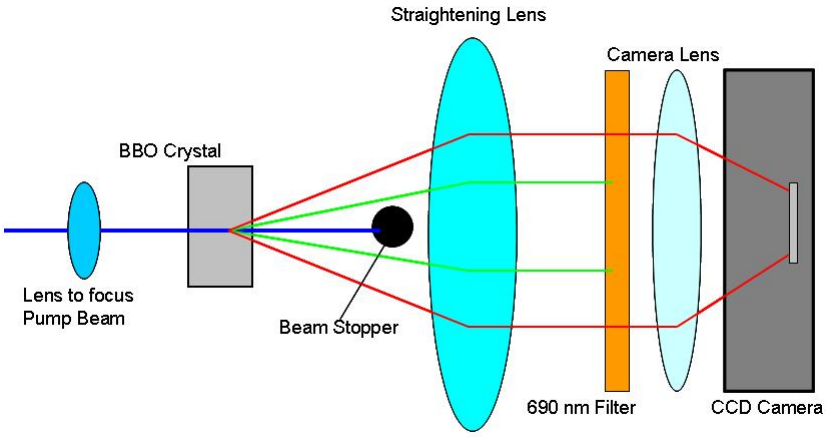

Figure 2.2 Experimental layout.

pictures were taken with a 400 mm focal length lens, placed 466 mm in front of the crystal. This spacing produced a focus at the down-conversion crystal. The beam waist was then smaller. Samples of these pictures are included in Section 3.1.

2.3 Study of Down-conversion Rings

We analyzed the data from our CCD camera in Matlab. We looked at an average over nine horizontal and nine vertical slices from the center of the ring. We then isolated

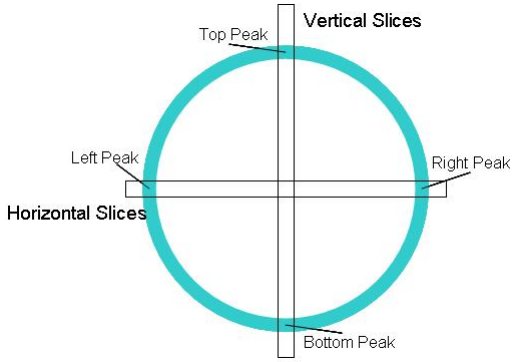

Figure 2.3 Method used for data analysis. The data was analyzed as horizontal and vertical slices. The strength of the down-conversion ring at each peak was fit to a Gaussian curve.

the left, right, top, and bottom peaks and fit them to Gaussian curves. This is shown visually in Fig. 2.3.

The parameters of the curve give the width and center of each peak as well as the level of background. The distance between the centers of the left and right peaks was taken to be the horizontal ring diameter. A vertical ring diameter was found using the top and bottom peaks. The ring diameter was found by averaging the horizontal and vertical ring diameters. The width of the ring was found by averaging the width of all four peaks. After subtracting the background, the total signal strength is found by adding the value of each pixel. The Matlab code for these calculations is included in Appendix A.2.

Chapter 3

Results and Conclusion

3.1 Comparison of Theory to Experiment

Using our model, we created images of down-conversion rings at ten different angles ranging from 33.14° to 34.19° with respect to the crystal axis. Images were of daughter photons with wavelengths in the range 690±5 nm. The pump beam radius had no effect on the images. Exploiting the radial symmetry of the system, we used our model to calculate Φ^2 along the radial axis. Some of these images are included in Fig. 3.1.

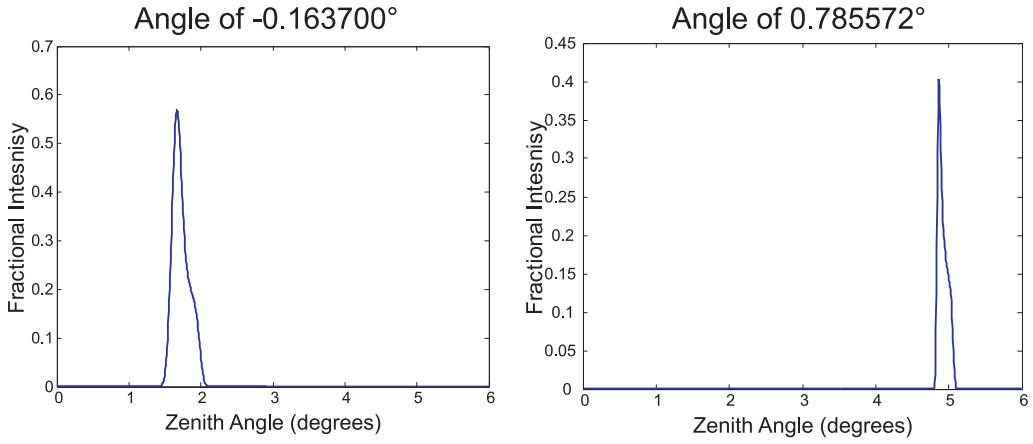

Figure 3.1 Sample figures of Computational Data. Plots are of Φ^2 along a radius taken from the center of the down-conversion ring.

Using a CCD camera, we took pictures of the 690nm down-conversion ring at ten different crystal angles ranging from 0.262° to 0.786° with respect to the crystal face. These correspond to the same angles as the computational data. At each angle, pictures were taken both without a lens and with a 400 mm lens placed 466 mm in front of the crystal. This lens focused the pump beam. Sample pictures from each set of CCD measurements are displayed in Fig. 3.2.

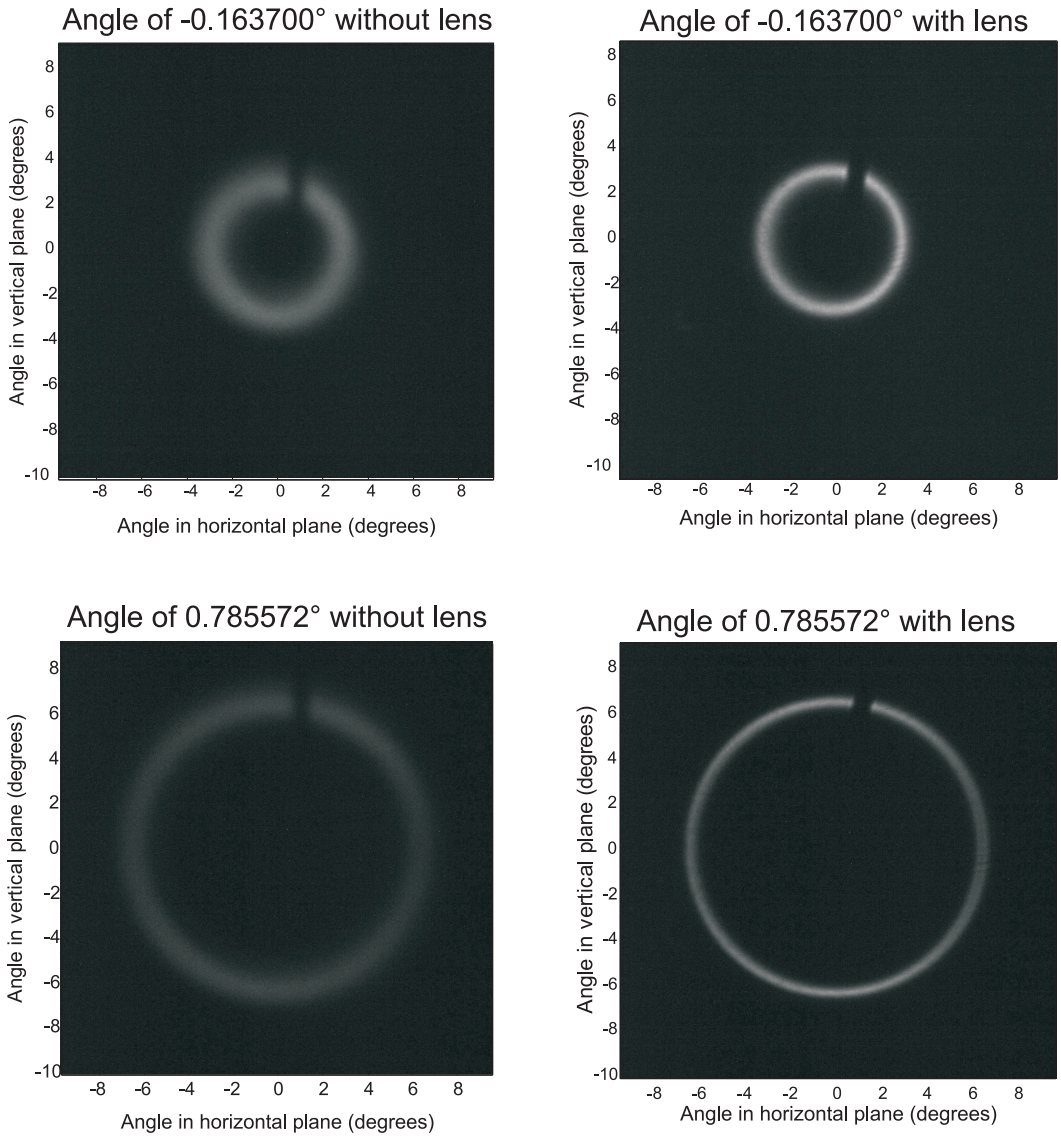

Figure 3.2 Sample of Experimental Pictures taken using a CCD camera.

Using the data analysis discussed in Sec. 2.3, we could fit the down-conversion rings to Gaussian curves to quantify the diameter and width of the down-conversion rings. The trends in these quantities are shown in Fig. 3.3.

Analysis of these data confirms the agreement of our computational model with experiment with regards to ring diameter. The data from the computational model match the observed data assuming a 0.2° error in the measurement of the crystal axis with respect to the crystal face. Such error is not uncommon in such manufactured crystals. This 0.2° error can be seen in the plot of Ring Diameter vs. Crystal Angle in Fig. 3.3. If each point of the experimental data is moved 0.2° to the right, the data line correspond almost exactly with the predictions of our model.

Experimentally, our system with its particular crystal and wavelength showed linearly decreasing trends in the width of the down-conversion ring. The decrease of the pump beam radius from the lens also caused the ring to be sharper and less wide. The computational model does not model this well, as is shown in the plot of Ring Width vs. Crystal Angle in Fig. 3.3. This is due to the incompleteness of the model. The model does not include the effects of pump beam radius and crystal tilt. Currently, changing the angle in the PMATCH program changes the angle of the pump beam with respect to the crystal axis. Experimentally, we change the angle with respect to the crystal face. The non-normal angle of the crystal face creates geometric effects causing the trends we see in the ring width.

The signal strength, the total down-conversion production at a single wavelength summed over all idler angles, was not affected by the crystal angle. This is shown in the plot of Fractional Signal Strength vs. Crystal Angle in Fig. 3.3. The effect of the focusing lens on the intensity was another important trend to study. Visual analysis of the pictures shows that the down-conversion ring is dimmer but more spread out without the lens present. After adjusting for reflection off the lens, the total signal

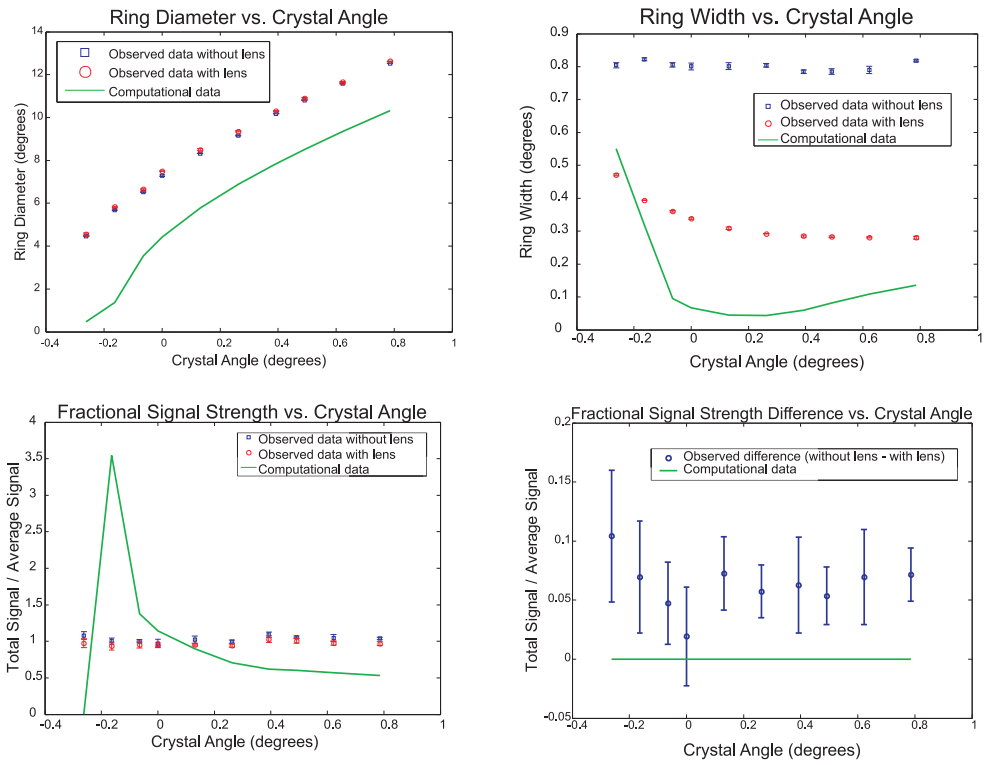

Figure 3.3 Plots of results comparing Computational Model to Observed Data.

was calculated to be stronger without the lens. This is shown in the plot of Fractional Signal Strength Difference vs. Crystal Angle in Fig. 3.3. Our model failed to predict any of these effects on the signal strength.

3.2 Conclusions

The theoretical studies and experiments agree with each other in predicting the location of the down-conversion rings. The diameter of the down-conversion ring increases linearly with the crystal angle. The signal strength of the ring is unaffected by the crystal angle. The lens focusing the pump beam appears to reflect enough of the pump beam away to decrease the production of down-converted photons.

The computational model does not correctly predict observed trends in ring width.

This is a consequence of our model being incomplete. Future work will incorporate the crystal tilt and pump radius effects into the computational model to correct this. To this point, the model is as correct as we would expect it to be.

Bibliography

[1] D. Ljunggren and M. Tengner, "Optimal focusing for maximal collection of entangled narrow-band photon pairs into single-mode fibers," Phys. Rev. A **72,** 062301 (2005).

[2] C. Kurtsiefer, M. Oberparleiter, and H. Weinfurter, "High-efficiency entangled photon pair collection in type-II parametric fluorescence," Phys. Rev. A **64,** 023802 (2001).

[3] C. Kurtsiefer, M. Oberparleiter, and H. Weinfurter, "Generation of correlated photon pairs in type-II parametric down conversion–revisited," Journal of Modern Optics **48,** 1997–2007 (2001).

[4] M. H. Rubin, "Transverse correlation in optical spontaneous parametric down-conversion," Phys. Rev. A **54,** 5349–5360 (1996).

[5] N. Boeuf, D. Branning, I. Chaperot, E. Dauler, S. Guerin, G. Jaeger, A. Muller, and A. Migdall, "Calculating characteristics of noncollinear phase matching in uniaxial and biaxial crystals," Opt. Eng. **39,** 1016–1024 (2000).

[6] A. Migdall, "Correlated-Photon Metrology without Absolute Standards," Physics Today **52,** 41–46 (1999).

[7] C. K. Hong and L. Mandel, "Theory of parametric frequency down conversion of light," Phys. Rev. A **31,** 2409–2418 (1984).

Appendix A

Programming Code

A.1 PMATCH

Below is the PMATCH code in Fortran that was used to create the computational figures. This takes input of Crystal Type, Crystal Length, Pump Beam Waist, Pump Beam Wavelength, Signal Wavelength, and Incident Angles θ and ϕ. The program then produces and image of Φ^2 (see Eq. 2.7) along a radial axis from the center of the down-conversion ring.

```
SUBROUTINE PLOT_PMF_X_Y(XAxis, PlotData)
    USE PM_Data; USE Crystal_Data
    IMPLICIT NONE

!   Arguments
    REAL*8, INTENT(INOUT) :: PlotData(XRes,YRes,ZRes)
    REAL*8, INTENT(INOUT) :: XAxis(XRes)
    !REAL*8, INTENT(INOUT) :: YAxis(YRes)

!   Local variables
    INTEGER Info, XIndex, FrequencyRes, FrequencyIndex
    REAL*8 Guess, PMF_Value, X, X_Step, Y, Frequency_Step, Theta_Sgnl_Ext
    REAL*8 FrequencyMin, FrequencyMax, Frequency_Sgnl,
                Frequency_Pump, Frequency_Idlr
    LOGICAL WAVELENGTHS_VALID, Success

!   Set axis range
    IF (AutoRange) THEN
        XMin = 0
        XMax = 5d0 * Pi / 180
    END IF
    FrequencyRes = 50
    FrequencyMin = (3.0d8/(.690 + .025))
    FrequencyMax = (3.0d8/(.690 - .025))
```

```
    X_Step = (XMax - XMin) / (XRes - 1)
    Frequency_Step = (FrequencyMax - FrequencyMin) / (FrequencyRes - 1)

!   Perform calculations
    Guess = Theta_Idlr_Guess

!   Clear PlotData
    DO XIndex = 1, XRes, 1
        PlotData(XIndex,1,1) = 0
    ENDDO

    DO FrequencyIndex = 1, FrequencyRes, 1
        WRITE(*,"(I3,'% Done')") FLOOR(100.0d0*FrequencyIndex/FrequencyRes)
        Frequency_Sgnl = FrequencyMin + (FrequencyIndex - 1)*Frequency_Step
        Lambda_Sgnl = 3.0d8/Frequency_Sgnl
        Frequency_Pump = 3.0d8 / Lambda_Pump
        Frequency_Idlr = Frequency_Pump - Frequency_Sgnl
        Lambda_Idlr = 3.0d8 / Frequency_Idlr

        DO XIndex = 1, XRes, 1
            X = DTAN(XMin + (XIndex - 1)*X_Step)
            Y = 0

            IF (Angle_External) THEN
                Theta_Sgnl_Ext = DATAN(DSQRT(X**2 + Y**2))
                CALL GetThetaInternal(Theta_Sgnl_Ext,Theta_Sgnl)
            ELSE
                Theta_Sgnl = DATAN(DSQRT(X**2 + Y**2))
            ENDIF

            Phi_Sgnl = DATAN(Y/X)
!         Make Phi_Sgnl between 0 and 2*Pi
            IF (X .LT.0d0) THEN
                Phi_Sgnl = Phi_Sgnl + Pi
            ENDIF

            IF (Phi_Sgnl .LT. 0d0) THEN
                Phi_Sgnl = Phi_Sgnl + 2d0*Pi
            ENDIF

            Phi_Idlr = Phi_Sgnl + Pi

            CALL FIND_OPT_PMF_1D(Guess, PMF_Value, Info)

            PlotData(XIndex,1,1) = PlotData(XIndex,1,1) +
                    (PMF_Value**2/FrequencyRes)
            XAxis(XIndex) = X * 180d0 / Pi
        ENDDO
    ENDDO

    IF ( .NOT. WAVELENGTHS_VALID() ) THEN
```

```
        WRITE(*,"('WAVELENGTH VIOLATION - Pump:',F8.4,'; Signal:',F8.4,';
                Idler:',F8.4)") Lambda_Pump, Lambda_Sgnl, Lambda_Idlr
        Wavelength_Violation = .TRUE.
    ENDIF

END SUBROUTINE PLOT_PMF_X_Y
```

A.2 Data Analysis in Matlab

A.2.1 FindFits.m

This code reads the data from a file, looks at slices from the data (see Fig. 2.3) and fits the data to Gaussian curves. The parameters of the Gaussian curve are used to determine the properties of the down-conversion ring.

```
% FindFits.m
% Extracts data and fits it to a Gaussian curve.

clear; close all;


files = dir('*.mat');
Signal = zeros(ceil(length(files)/2),2);
RingDiameterHorizontal = zeros(ceil(length(files)/2),2);
RingDiameterVertical = zeros(ceil(length(files)/2),2);
RingDiameter = zeros(ceil(length(files)/2),2);
PeakWidthLeft = zeros(ceil(length(files)/2),2);
PeakWidthRight = zeros(ceil(length(files)/2),2);
PeakWidthTop = zeros(ceil(length(files)/2),2);
PeakWidthBottom = zeros(ceil(length(files)/2),2);
PeakWidth = zeros(ceil(length(files)/2),2);

angle = zeros(ceil(length(files)/10),1);
angleinternal = zeros(ceil(length(files)/10),1);
index = [1.58,1.5844,1.5908,1.5908,1.5961,1.6089,1.6239,1.6404, ...
    1.6529,1.6691,1.6868]; % Calculated for these specific angles

for istep = 1:length(files)
    data = load(files(istep).name);

    % Scale is 1 cm per 615 pixels
    scale = 1/615;

    % Distance to focus is 48 mm.
    dist = 4.8;

    % Name the picture
    with = 0;
    if istep<19
        picturename = strcat('0',int2str(ceil(istep/2)));
    else
```

```
    picturename = int2str(ceil(istep/2));
end

if isempty(findstr(files(istep).name,'out')) == 1
    with = 1;
    picturename = strcat(picturename,'a');
end

angle(ceil(istep/10),1)=str2double(files(istep).name(findstr( ...
    files(istep).name,'=')+1:findstr(files(istep).name,'W')-2));
ne = 1.577429;
no = 1.705463;
angleinternal(ceil(istep/10),1) = 180/pi * atan(ne * sin(angle(ceil ...
    (istep/10),1) * pi/180)./(no * (ne^2 - sin(angle(ceil(istep/10),1) ...
    * pi/180)^2)^(1/2)));

xwindow = 301:1300;
ywindow = 1:1000;

cropped = data.data(ywindow,xwindow);

rowout = mean(cropped(496:504,:));
columnout = mean(cropped(:,487:495)');

left = rowout(1:500);
right = rowout(501:1000);
top = columnout(1:500);
bottom = columnout(501:1000);

%**** Left side peak ****

[lmax,lcenter]=max(left);

xl = 1:length(left);
yl = left;
a0l(1) = lcenter;
a0l(2) = 30;
a0l(3) = lmax-40;
a0l(4) = 40;

option=optimset('TolX',1e-2);
al = fminsearch(@leastsq,a0l,option,xl,yl);
if sum((yl-funcfit(al,xl)).^2) > 1e5
    fprintf('l');
    break;
end

%**** Right side peak ****

[rmax,rcenter]=max(right);
```

```
xr = 1:length(right);
yr = right;
a0r(1) = rcenter;
a0r(2) = 30;
a0r(3) = rmax-40;
a0r(4) = 40;

option=optimset('TolX',1e-2);
ar = fminsearch(@leastsq,a0r,option,xr,yr);
if sum((yr-funcfit(ar,xr)).^2) > 1e5
    fprintf('r');
    break;
end

RingDiameterHorizontal(ceil(istep/2),with+1) = length(left) - ...
    al(1) + ar(1);


%**** Top peak ****

[tmax,tcenter]=max(top);

xt = 1:length(top);
yt = top;
a0t(1) = tcenter;
a0t(2) = 30;
a0t(3) = tmax-40;
a0t(4) = 40;

option=optimset('TolX',1e-2);
at = fminsearch(@leastsq,a0t,option,xt,yt);
if sum((yt-funcfit(at,xt)).^2) > 1e5
    fprintf('t');
    break;
end

%**** Bottom peak ****

[bmax,bcenter]=max(bottom);

xb = 1:length(bottom);
yb = bottom;
a0b(1) = bcenter;
a0b(2) = 30;
a0b(3) = bmax-40;
a0b(4) = 40;

option=optimset('TolX',1e-2);
ab = fminsearch(@leastsq,a0b,option,xb,yb);
if sum((yb-funcfit(ab,xb)).^2) > 1e5
    fprintf('b');
```

```
    break;
end

background = (at(4)+ab(4)+al(4)+ar(4))/4;
if with == 0
    reflection = (1-(.4767/2.4767)^2)^2 * (1-((cos(pi/180*angle(ceil ...
        (istep/10))) - index(ceil(istep/10))*cos(pi/180*angleinternal ...
        (ceil(istep/10))))/(cos(pi/180*angle(ceil(istep/10))) + index ...
        (ceil(istep/10))*cos(pi/180*angleinternal(ceil(istep/10)))))^2);
else
    reflection = 1-((cos(pi/180*angle(ceil(istep/10))) ...
        - index(ceil(istep/10))*cos(pi/180*angleinternal(ceil ...
        (istep/10))))/(cos(pi/180*angle(ceil(istep/10))) + index ...
        (ceil(istep/10))*cos(pi/180*angleinternal(ceil(istep/10)))))^2;
end
Signal(ceil(istep/2),with+1) = sum(sum(cropped - background));%*reflection

RingDiameterVertical(ceil(istep/2),with+1) = length(left) - at(1) + ab(1);
RingDiameter(ceil(istep/2),with+1) = 180/pi*atan(scale* ...
    (RingDiameterVertical(ceil(istep/2),with+1) + ...
    RingDiameterHorizontal(ceil(istep/2),with+1))/(2*dist));

PeakWidthLeft(ceil(istep/2),with+1) = al(2);
PeakWidthRight(ceil(istep/2),with+1) = ar(2);
PeakWidthTop(ceil(istep/2),with+1) = at(2);
PeakWidthBottom(ceil(istep/2),with+1) = ab(2);
PeakWidth(ceil(istep/2),with+1) = 180/pi*atan(scale*(PeakWidthLeft ...
    (ceil(istep/2),with+1) + PeakWidthRight(ceil(istep/2),with+1) + ...
    PeakWidthTop(ceil(istep/2),with+1) + PeakWidthBottom(ceil ...
    (istep/2),with+1))/(4*dist));


x = (-(RingDiameterHorizontal(ceil(istep/2),with+1)/2 + al(1)): ...
    length(xwindow) - 1 -(RingDiameterHorizontal(ceil(istep/2), ...
    with+1)/2 + al(1))) * scale;
y = (-(RingDiameterVertical(ceil(istep/2),with+1)/2 + at(1)): ...
    length(ywindow)- 1 - (RingDiameterVertical(ceil(istep/2), ...
    with+1)/2 + at(1))) * scale;

xangle = atan(x/dist)*180/pi;
yangle = atan(y/dist)*180/pi;

% plot vs angles
picture = pcolor(xangle,yangle,cropped);
shading interp;
axis equal;
colormap(lines);
xlabel('Angle in horizontal plane (degrees)');
ylabel('Angle in vertical plane (degrees)');
saveas(picture,picturename,'jpg');
```

```
end

% Average over trials within each angle
AvgRingDiameter = zeros(ceil(length(files)/10),4);
AvgPeakWidth = zeros(ceil(length(files)/10),4);
AvgSignal = zeros(ceil(length(files)/10),4);
for anglestep = 1:ceil(length(files)/10)
    AvgPeakWidth(anglestep,1)=mean(PeakWidth(5 * (anglestep-1) + 1 : 5 * ...
        (anglestep-1) + 5, 1));
    AvgRingDiameter(anglestep,1)=mean(RingDiameter(5 * (anglestep-1) + ...
        1 : 5 * (anglestep-1) + 5, 1));
    AvgSignal(anglestep,1)=mean(Signal(5 * (anglestep-1) + 1 : 5 * ...
        (anglestep-1) + 5, 1));
    AvgPeakWidth(anglestep,2)=mean(PeakWidth(5 * (anglestep-1) + 1 : 5 * ...
        (anglestep-1) + 5, 2));
    AvgRingDiameter(anglestep,2)=mean(RingDiameter(5 * (anglestep-1) + 1 : ...
        5 * (anglestep-1) + 5, 2));
    AvgSignal(anglestep,2)=mean(Signal(5 * (anglestep-1) +1 : 5 * ...
        (anglestep-1) + 5, 2));
    AvgPeakWidth(anglestep,3)=std(PeakWidth(5 * (anglestep-1) + 1 : 5 * ...
        (anglestep-1) + 5, 1));
    AvgRingDiameter(anglestep,3)=std(RingDiameter(5 * (anglestep-1) + 1 : 5 * ...
        (anglestep-1) + 5, 1));
    AvgSignal(anglestep,3)=std(Signal(5 * (anglestep-1) + 1 : 5 * ...
        (anglestep-1) + 5, 1));
    AvgPeakWidth(anglestep,4)=std(PeakWidth(5 * (anglestep-1) + 1 : 5 * ...
        (anglestep-1) + 5, 2));
    AvgRingDiameter(anglestep,4)=std(RingDiameter(5 * (anglestep-1) + 1 : 5 * ...
        (anglestep-1) + 5, 2));
    AvgSignal(anglestep,4)=std(Signal(5 * (anglestep-1) + 1 : 5 * ...
        (anglestep-1) + 5, 2));
end
save 'Angle.mat' angle;
save 'PeakWidth.mat' AvgPeakWidth;
save 'RingDiameter.mat' AvgRingDiameter;
save 'Signal.mat' AvgSignal;
```

A.2.2 funcfit.m

This is the Gaussian curve to which the data is being fit.

```
% funcfit.m

function f=funcfit(a,x)

f=a(3)*exp(-(x-a(1)).^2/(a(2).^2))+a(4);

%a(1) is the center of the peak
%a(2) is the width
%a(3) is a scaling factor
%a(4) is the offset from zero
```

A.2.3 leastsq.m

This part of the code minimizes the difference between the data and the Gaussian fit.

```
% leastsq.m

function s=leastsq(a,x,y)
s=sum((y-funcfit(a,x)).^2);
```

Index