

LUND UNIVERSITY

FYSK02

Applications of Orbital Angular Momentum of Light
in Attosecond Science

Author:

Albin HEDSE

Thesis submitted for B.Sc.
Project duration: 2 months
Department of Physics
Division of Atomic Physics
June 2018

Supervisors

Mathieu GISSELBRECHT
Cord ARNOLD



LUND
UNIVERSITY

July 4, 2018

Acknowledgements

I would like to thank my supervisor, Mathieu Gisselbrecht, for being very helpful and patient with me. You not only taught me a lot about physics, but also a lot about the scientific procedure in general and for putting in the time and effort to help me throughout the project. I would also like to thank my other supervisor, Cord Arnold, for all the helpful advice on how to improve the simulation and assistance regarding the optical aspects of the project. Last but certainly not least, I would like to thank my family and friends for being very supportive of me, I could not ask for better people to have in my life.

Abstract

Within attosecond science, two-color two-photon ionization is widely used to study photoemission dynamics. Traditionally a single-atom model is assumed, meaning that any electron emissions within the volume of interaction are treated as results of entirely identical and isolated systems. This will not be the case when light carrying orbital angular momentum is used. Due to the helical nature of the wavefront of this light, such an assumption can no longer be made. This necessitates a more thorough investigation of the volume of interaction.

The aim of this thesis was to provide a method of mapping this volume of interaction as a means of assisting research in this field by means of a numerical simulation of a stationary light beam. This simulation is tested both for freely propagating beams and beams propagated through a converging lens. The former a prerequisite for the latter, while the second replicates actual experimental conditions, as experiments will be conducted where the beam is in focus. During the course of the project, it was found that optimization was necessary, as the computation time was too long to be useful. This led to computation times being significantly shorter.

It was found that within the relevant environment, the simulation appears to be able to accurately simulate beams of light carrying orbital angular momentum, though further development is necessary before proper experimental implementation can be considered.

Contents

| | |
|--|------------|
| Acknowledgements | iii |
| Abstract | iv |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Theory | 2 |
| 1.2.1 Light Propagation | 2 |
| 1.2.2 Gaussian and Laguerre-Gaussian Beams | 3 |
| 1.2.3 Simulation of Light Propagation: the Huygens-Fresnel Principle | 5 |
| 1.3 This work | 6 |
| 2 Method | 7 |
| 2.1 Principle | 7 |
| 2.1.1 Initialization | 7 |
| 2.1.2 Propagation and Discretization | 8 |
| 2.2 Implementation | 8 |
| 2.2.1 Differences in Early Versions and Challenges | 9 |
| 2.2.2 Current Version | 9 |
| 2.3 Further Considerations | 10 |
| 3 Results | 10 |
| 3.1 Free Propagation | 10 |
| 3.1.1 Gaussian Beam | 11 |
| 3.1.2 Beam with Orbital Angular Momentum | 12 |
| 3.2 Propagation with an Optical Element | 15 |
| 3.2.1 Gaussian Beam Propagating Through an Optical Element | 16 |
| 3.2.2 Beam With Orbital Angular Momentum Propagating Through an Optical Element | 16 |
| 4 Outlook | 19 |
| 5 References | 21 |
| Appendix: Python Simulation Code | 22 |

Abbreviations:

OAM - Orbital Angular Momentum

FWHM - Full Width at Half Maximum

HHG - High Harmonic Generation

RABBITT - Reconstruction of Attosecond Beating By Interference of Two-photon Transitions

IR - Infrared

1 Introduction

1.1 Motivation

Orbital angular momentum (OAM) of light, while theorized before, was not experimentally observed until 1992 [1]. Angular momentum is a physical quantity which is analogous to linear momentum for rotational motions. For light, angular momentum can manifest in two ways: spin angular momentum and OAM, where spin angular momentum manifests only in terms of circular polarization. Orbital angular momentum manifests in terms of the spatial distribution of the light around its axis of propagation. Orbital angular momentum of light is quantized and takes on values given by $l\hbar$ per photon, where l is any integer number. Light carrying OAM has potential applications within several fields, with microscopy [2] and quantum communications [3] being among the most prominent. Recently, it has also been introduced in attosecond science, which is a field of research that deals with the study of ultrafast dynamics of electrons.

Attosecond science is the study of fast processes on the time scales of attoseconds (10^{-18} s). The possibility of applying OAM within attosecond science was proposed a few years ago to create "springs" [4, 5] of either ultra-short light or electron pulses. These springs may be created through the inclusion of OAM in already existing elements of attosecond science. An example of this would be the two-color two-photon ionization interferometric technique called RABBITT [6]. Two-color refers to the fact that the ionization process is carried out through the interaction of an atom with two photons of different wavelength, leading to ionization. Use of this technique has shown that if a driving laser carrying OAM creates attosecond pulses through the process of high harmonic generation (HHG), the emitted photons will carry some amount of OAM [7]. Upon ionization by such a light source, the emitted electrons may also be encoded with information about the light fields involved. The long term perspective of the project is to propose a thought experiment to show whether or not OAM is transferred to these electrons.

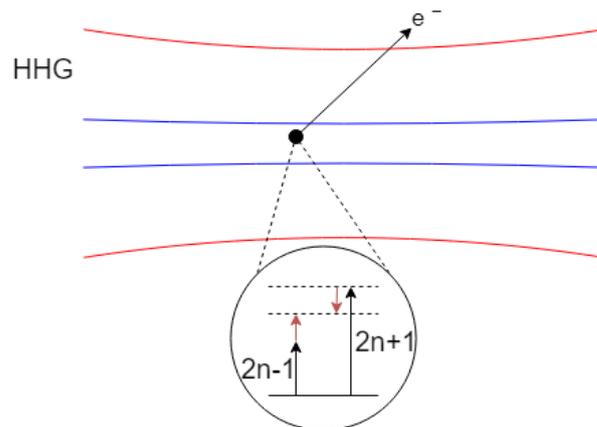


Fig. 1: Schematic illustration of the interaction region of the RABBITT technique. The outer, red borders represent the infrared beam width and the inner, blue borders represent the beam width of the high harmonics. Within these inner borders two-photon ionization occurs, as illustrated in the inset, where two possible "quantum paths" prior to emission can be taken. The large arrows represent absorbed photons of two adjacent odd harmonics $2n-1$ and $2n+1$. The small red arrows represent absorption and emission of a IR photon, respectively.

The principle of this thought experiment is based on the RABBITT interferometric technique, as illustrated in Figure 1. High harmonic generation (HHG) produces attosecond pulse trains, which correspond to a frequency comb of coherent odd-order harmonics of a driving infrared (IR) ultrashort laser pulse [6]. Once the HHG and IR beams are recombined in an interaction region, atoms can be ionized with a two-color two-photon ionization process, illustrated in the inset of Figure 1. Such a process leads to characteristic sideband peaks in the photoelectron spectrum, which can be formed after absorption of an arbitrary harmonic $2n - 1$ followed by absorption of a photon of the driving laser, or by absorption of a higher harmonic $2n + 1$ and emission of a driving laser photon. These quantum paths both lead to a state with identical energy, however, each quantum path takes a different amount of time. These paths can thus interfere and the phase difference between the two paths can be determined by scanning the delay between the HHG and IR.

This type of experiment is currently widely used with regular beams to study the photo-emission time delay upon ionization [6]. There is, however, a key difference for OAM-carrying beams. The main difference is that the spatial coherence of the OAM beams, i.e. the wavefront of the beam, matters. This means that electrons are expected to be emitted with different relative phase based not only on the difference in emission time of their two-photon quantum paths, but also based on their position around the beam axis [7]. A "single atom" response, as it is usually assumed, can thus no longer be used to describe the RABBITT technique and a method of mapping the two-photon process within the volume of interaction is necessary. This is the motivation of this thesis.

1.2 Theory

1.2.1 Light Propagation

In wave optics, a wave of light may be represented by a wave function, which is any function u that can satisfy the wave equation [8]

$$\nabla^2 u - \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} = 0, \quad (1)$$

where c is the speed of light, u is a function dependent on position $\mathbf{r} = (x, y, z)$ and time t and ∇^2 is the Laplace operator. Using the monochromatic approximation, i.e. that all light involved has the same wavelength, the complex function $u(\mathbf{r}, t)$ can be written as

$$u(\mathbf{r}, t) = a(\mathbf{r}) \exp[i\phi(\mathbf{r})] \exp(i2\pi\nu t), \quad (2)$$

where $a(\mathbf{r})$ is the amplitude of the wave, i is the imaginary unit, $\phi(\mathbf{r})$ is the phase and ν is the frequency of the wave. If equation 2 is then substituted into equation 1, one gets a differential equation for the complex amplitude of the wave, $U(\mathbf{r}) = a(\mathbf{r}) \exp[i\phi(\mathbf{r})]$:

$$\nabla^2 U(\mathbf{r}) + k^2 U(\mathbf{r}) = 0, \quad (3)$$

where k is the wavevector $\frac{2\pi\nu}{c}$. The above equation is known as the Helmholtz equation. Solutions of this equation often make use of the paraxial approximation, which uses paraxial waves, i.e. waves with parabolic wavefronts rather than spherical or plane wavefronts. A way to define a paraxial wave is to use a plane wave $A \exp(-ikz)$ and to modulate

its complex envelope A , so that it is a function which varies slowly with its position [8], which gives it the complex amplitude

$$U_{\text{par}}(\mathbf{r}) = A(\mathbf{r}) \exp(-ikz). \quad (4)$$

Important to note, is that $A(\mathbf{r})$ and its derivative with respect to z should both vary slowly within the distance of a single wavelength in the z -direction. This is necessary in order for the wave to be approximately equal to a plane wave in this interval, effectively meaning that $\Delta z = \lambda$ so that $\Delta A \ll A$. As $\Delta A \cong (\partial A/\partial z)\Delta z = (\partial A/\partial z)\lambda$, this gives $\partial A/\partial z \ll A/\lambda = Ak/2\pi$, leading to

$$\frac{\partial A}{\partial z} \ll kA.$$

since $\partial A/\partial z$ also varies slowly within the span of a single wavelength, this means that $\partial^2 A/\partial z^2 \ll k\partial A/\partial z$, giving:

$$\frac{\partial^2 A}{\partial z^2} \ll k^2 A.$$

By substituting equation 4 into equation 3 and using the assumption that $A(\mathbf{r})$ changes slowly with respect to z , one can neglect $\partial^2 A/\partial z^2$ due to its relative size compared to $k\partial A/\partial z$ or $k^2 A$. This can be written as a new partial differential equation for the complex envelope $A(\mathbf{r})$, known as the paraxial Helmholtz equation:

$$\nabla_T^2 A - i2k \frac{\partial A}{\partial z} = 0, \quad (5)$$

where $\nabla_T^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$.

1.2.2 Gaussian and Laguerre-Gaussian Beams

A standard laser beam is often modeled as a Gaussian beam as it resembles actual laser beams, which is why it is useful to use as a point of reference. A Gaussian beam, as the name implies, results in a Gaussian intensity distribution centered at the beam axis. The Gaussian beam equation is written as below [8]:

$$U(\mathbf{r}) = A_0 \frac{W_0}{W(z)} \exp\left(-\frac{\rho^2}{W^2(z)}\right) \exp\left[-ikz - ik\frac{\rho^2}{2R(z)} + i\zeta(z)\right], \quad (6)$$

where A_0 is a constant, $\rho = \sqrt{x^2 + y^2}$ is the radial distance from the axis of propagation, $W(z)$ is a measure of the beam width, $W_0 = \sqrt{\frac{\lambda z_0}{\pi}}$ is known as the waist width, and it represents the width of the beam at its thinnest point which is known as the waist, where z_0 is a constant known as the Rayleigh range, which is determined by the waist width and the wavelength of the beam, $\zeta(z)$ is the Gouy phase term defined as $\zeta(z) = \arctan(\frac{z}{z_0})$ and $R(z)$ is the radius of curvature of the beam wavefront, given by $R(z) = z[1 + (\frac{z_0}{z})^2]$, all of which are illustrated in Figure 2. The waist width in particular is important here, it can be written separately as below:

$$W(z) = W_0 \sqrt{1 + \left(\frac{z}{z_0}\right)^2}, \quad (7)$$

at a distance $z = 0$, the beam width is equal to the waist width. The Rayleigh range itself is also an important beam parameter, which determines the depth of focus of the beam, where the beam is in focus or nearly in focus. A Gaussian beam has a near-flat wavefront near the beam waist, which gets more and more spherical at larger distances from the beam waist. This means that there is a near field where the beam has a near-flat wavefront and a far field where the beam has a nearly spherical wavefront. The Rayleigh range marks precisely where the wavefront of the Gaussian beam has the smallest radius of curvature. Inside of the Rayleigh range the radius rapidly diverges towards infinity as the wavefront gets flatter and it approximately linearly increases for distances further away as the wavefront becomes more spherical. The Rayleigh range is thus an important indicator of where the shift between near and far field occurs for a Gaussian beam.

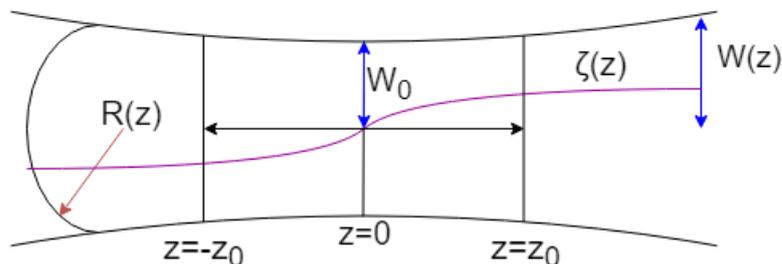


Fig. 2: Illustration of a Gaussian beam near its waist, with some relevant beam parameters. The Gouy phase term $\zeta(z)$, represented by the violet line, describes a phase shift of the beam wavefront relative to the phase of a plane wave which is initially phase coherent with it after having traveled a distance z from the beam waist [8]. The radius of curvature, depicted by the red arrow, gives the radius of curvature of the wavefront at a position z relative to the beam waist. The beam width, characterized by the blue arrows, describes the radius of the beam relative to its axis of propagation. Represented by the black arrow and horizontal bars is the depth of focus, within which the distance to the beam waist is less than z_0 , the distance associated with the Rayleigh range.

For an actual experiment though, it is hard to measure this 'beam width' and one may wish to use the full width at half maximum (FWHM) of the beam intensity instead. The intensity of the Gaussian beam, the definition $I(\mathbf{r}) = |U(\mathbf{r})|^2$ is used, which, using equation 6, gives the function

$$I(\rho, z) = I_0 \left(\frac{W_0}{W(z)} \right)^2 \exp \left[-\frac{2\rho^2}{W^2(z)} \right], \quad (8)$$

With a normalized beam intensity with respect to a given distance, the FWHM is given by [9]:

$$\text{FWHM} = W(z) \sqrt{2 \ln(2)}. \quad (9)$$

The Gaussian beam is one possible solution of the paraxial Helmholtz equation, however, there are many other possible solutions that exist. The complete set of solutions to the paraxial Helmholtz equation that is of interest to this work is the Laguerre-Gaussian set, as its individual solutions represent all possible beams which carry OAM.

The Laguerre-Gaussian beam is found by applying the paraxial Helmholtz equation in terms of cylindrical coordinates (ρ, ϕ, z) on a wave with the complex amplitude of a modulated Gaussian beam. Now the azimuthal symmetry with respect to the beam axis is broken, as it now has a helical wavefront. A separation of variables technique is then used, which gives a set of differential equations. The solutions to these differential equations give a complete set of solutions to the paraxial Helmholtz equation [8], as shown below:

$$U_{l,m}(\rho, \phi, z) = A_{l,m} \left[\frac{W_0}{W(z)} \right] \left(\frac{\rho}{W(z)} \right)^l \mathbb{L}_m^l \left(\frac{2\rho^2}{W^2(z)} \right) \exp \left(-\frac{\rho^2}{W^2(z)} \right) \exp \left[-ikz - ik\frac{\rho^2}{2R(z)} - il\phi + i(l+2m+1)\zeta(z) \right]$$

where l is an integer number which indicates the amount of OAM carried by the beam and m is a positive integer number which represent beams with $m+1$ radial positions with local intensity maxima. $A_{l,m}$ is a constant, $\mathbb{L}_m^l(\cdot)$ is the general Laguerre polynomial function, as defined by:

$$\mathbb{L}_m^l(x) = (x^{-l}e^x/m!)(d^m/dx^m)(x^{l+m}e^{-x}).$$

$U_{l,m}(\rho, \phi, z)$ is the complex amplitude of a Laguerre-Gaussian beam. Here, the specific properties of the $m \neq 0$ modes [10] are not of interest currently, as it adds little to experimental investigations. There are three key points worth noting here. Firstly, if $l = m = 0$ is inserted into the Laguerre-Gaussian beam equation, the regular Gaussian beam remains. Secondly, the intensity of these Laguerre-Gaussian modes as given by $I = |U|^2$, is for $l \neq 0$ modes of the beam zero along the beam axis. This yields the characteristic ring-shaped intensity profile of an OAM-carrying beam. Thirdly, the phase of the Laguerre-Gaussian beam is linearly dependent on the angle ϕ , proportional to the value of l . This means that Laguerre-Gaussian beam modes where $l \neq 0$ have a spiral phase, which is why the beam wavefront is helical and carries OAM.

From the intensity distribution of the Laguerre-Gaussian beam mode, the radius of maximal intensity for Laguerre-Gaussian modes can be derived (done in the supplementary material of [7]) and is for a fixed wavelength given by:

$$\varrho_{\max} = W(z) \sqrt{\frac{l}{2}}. \quad (10)$$

The radius of maximal intensity may be used as a measure of the beam divergence of a Laguerre-Gaussian beam modes. Equation 10 above indicates that Laguerre-Gaussian $m = 0$ beam modes carrying OAM have a beam divergence similar to the $l = 0$ mode, with the difference being the constant of proportionality $\sqrt{\frac{l}{2}}$. This linear relation between Laguerre-Gaussian modes and regular beams will be used as an indication of the general relation between OAM-carrying and regular beams.

1.2.3 Simulation of Light Propagation: the Huygens-Fresnel Principle

The Huygens-Fresnel Principle states that every point on a wavefront generates spherical waves, which in turn generate a new wavefront [8]. Theoretically, one could perfectly simulate a wave with infinitely many of these point emitters. The field at a given position,

(x, y, z) , emitted by each point emitter located in a plane at $z=0$ is calculated using the following equation:

$$E(x, y, z) = \int \int \frac{E_0(x', y', 0)}{r} \exp[i(-kr + \varphi(x', y'))] dx dy, \quad (11)$$

where $r = \sqrt{(x - x')^2 + (y - y')^2 + z^2}$ is the distance from the emitter, φ is the phase of the emitter and E_0 is the field strength of the emitter, given for instance for a Gaussian beam at the waist:

$$E_0(x', y') = A_0 \exp \left[-\frac{x'^2 + y'^2}{W_0} \right] \quad (12)$$

where A_0 is a constant.

For a numerical calculation, a finite amount of points is necessary and the criteria for proper beam propagation will be presented in the method section and discussed in the result section.

To modify the beams properties or a change in the optical path, a phase $\varphi(x', y')$ can be applied. For instance, by applying a spiral phase, φ_{spp} , to a regular $l = 0$ Gaussian beam, one adds a phase term $-il\phi$ like the one in the Laguerre-Gaussian beam. Such a phase can be written as:

$$\varphi_{spp} = l \cdot \arctan\left(\frac{y}{x}\right) \quad (13)$$

where x, y are the coordinates of the point to which the phase is assigned. This modified beam will not be one of the Laguerre-Gaussian beam modes, as Laguerre-Gaussian beams additionally have a larger factor in front of their Gouy phase term and dependence on the Laguerre polynomial. However, such a beam is easy to make experimentally by introducing a spiral phase plate and presents most of the interesting behavior of a Laguerre-Gaussian beam, i.e. the spiral-phase modified beam have the OAM's property due to its helical wavefront [7].

If we now consider the introduction of an optical element such as a thin lens, a phase φ_{lens} will be applied to the beam:

$$\varphi_{lens}(x', y') = k \frac{x'^2 + y'^2}{2f} \quad (14)$$

where f is the focal length of the lens and k is the wavevector associated with the beam [8].

1.3 This work

The goal of this work is to assist in the investigation of the volume of interaction of attosecond light pulse trains carrying OAM. This is done through the development of a Python program which aims at numerically simulating beam propagation carrying OAM under realistic experimental conditions.

The program in its current state simulates a beam of light through use of matrices and spherical point emitters, which is propagated through a converging lens. The simulation can observe the beam in a specific plane, such as the plane where the beam is refocused, which is the primary area of interest for experimental observations. Further description of the simulation is given, along with an explanation of the numerous challenges encountered, in the method section.

In the results section, the simulation is compared with existing analytical derivations and its accuracy is analyzed. Its viability as an experimental aid and potential improvements are discussed in the end of the work.

2 Method

The simulation introduced above makes use of square matrices to observe a propagating wave. Each of the matrix elements in these matrices represents a spherical wave point source, these are used to propagate the beam itself. Each matrix has a certain size in terms of the amount of elements it contains and a spatial size, which determines the physical size of the plane it represents. These two terms in conjunction determine the resolution of the matrix. Having a good enough resolution, i.e. matrix points per distance, is crucial in order for proper propagation. In order to avoid diffraction, the matrix must have a spatial width larger than the beam by a factor of $2\pi \cdot W(z)$ to avoid diffraction, something that changes somewhat for OAM-carrying beams, as will be discussed in coming sections. For a Gaussian beam, however, the mentioned plane width is the one that allows for maximal resolution without diffraction effects.

2.1 Principle

In the simulation, matrices are used to represent spatial planes, where the user may observe the intensity, field distribution, or phase of the simulated beam. The beam itself is propagated through use of equation 11 to a plane, called the "observation plane" from a plane henceforth referred to as the "source plane". The field contribution of each point emitter in the origin matrix is summed at each point in the target matrix as illustrated in Figure 3. Eventually a third matrix was added to simulate an optical element, here a converging lens, illustrated later in the results section in Figure 7. For the purposes of finding the intensity of the beam, a few new matrices are created. These matrices simply take the field values at their corresponding plane and calculate the absolute square of each element, giving the intensity distribution. The values in this matrix are then normalized such that the highest value in it is assigned the value 1. These matrices are then the ones used for most calculations, such as for finding the FWHM of the Gaussian beam.

2.1.1 Initialization

Initial x and y-coordinates are set by using the beam widths described by equation 7 and multiplying them by π . "Step sizes" Δx and Δy are then calculated based on the dimensions of the matrix and the initial x and y-coordinates. The step sizes, i.e. the distances between each matrix point in the x and y-directions, are calculated so that the coordinate values at the top left corner are the same as the ones on the bottom right corner, except multiplied by a factor of (-1) , such that the center of the matrix is positioned at the origin. The x and y-coordinates are saved as vectors in a position matrix. The process is repeated for all matrix indices i, j in the source plane and l, m in the observation plane, until all matrices involved in the propagation are generated.

After this, the initial energies E_0 of the source plane are calculated based on the coordinate values of the appropriate position matrix using equation 12. These values are then distributed in the source plane based on the one coordinates used in the position matrix used to calculate the field there.

In addition to all other matrices, the simulation also makes use of phase matrices, which are used to apply either the initial spiral phase necessary for the OAM-carrying beams, which are simply matrices which carry the phase applied at each point in the

appropriate matrix. The spiral phase matrices are generated by applying a phase between 0 and 2π to positions in a matrix based on their azimuthal angular position around the center relative to a given axis, which is here set to be the positive x-axis of the matrix. The phase given by the optical element is calculated using Eq. 14 and the position matrix for the optical element.

2.1.2 Propagation and Discretization

When using the Huygens-Fresnel principle to simulate beam propagation, there is a matter of discretization of the point emitters to be taken into account. Consequently, care must be taken when selecting enough emitters used to represent the beam, so that one can properly resolve the wave or beam within a useful angle of divergence. The angle supported by the current discretization is given by the position between wave emitters, provided that these are all evenly spaced with respect to each other in the plane they are all located in. The angle within which the beam is properly resolved is found in the same way as one may find the lateral resolution of an object and is given by:

$$\sin(\theta) = \frac{\lambda}{2\Delta x} \quad (15)$$

where θ is the angle resolved by the current resolution and Δx is the distance between wave emitters. This equation assumes that the spacing between adjacent emitters in the grid is constant, which is the case in this simulation.

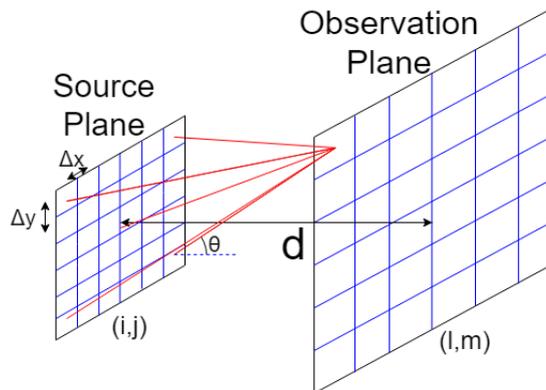


Fig. 3: An illustration of the propagation process used in the simulation, where d represents the distance between the planes and θ represents the angle from equation 15, i, j, l, m are the matrix indices, as used in equation 16 and Δx and Δy represent the spatial distance between matrix points in the x and y -directions. θ is defined so that it is the largest of the angles between the lines between matrix points (seen in red) and the plane normal. The field contribution from each point in the source plane is calculated at a point in the observation plane. The process is then repeated until the resulting field is calculated for every point in the observation plane. The field can then be observed, analyzed and used for further propagation.

2.2 Implementation

The simulation is programmed using Python with its standard libraries in addition to the libraries NumPy and matplotlib.

2.2.1 Differences in Early Versions and Challenges

Initially, the simulation used four nested loops in order to propagate the beam from one plane to another. The two "outer" loops select the point in the observation plane for which the beam calculations are to be made. The two inner loops then calculate the field contribution in that point from each point in the source plane. This was the basis for the early versions of the simulations for studying free propagation. The nested loop structure sums each sum index from 0 to n , where n represents the amount of matrix columns or rows, working as described in equation 16 below.

$$E_{obs\ l,m} = \sum_{i,j} \frac{E_{i,j\ source}}{r_{ij-lm}} \exp[-ikr_{ij-lm}], \quad (16)$$

$E_{obs\ l,m}$ is the field strength in the observation plane associated with indices (l, m) and E_{source} is the field strength of the emitter associated with the indices (i, j) in the source plane. Additionally, r_{ij-lm} represents the distance between matrix point in the source plane represented with the indices (i, j) and the matrix point in the observation plane represented with indices (l, m) . This process is repeated for every index (l, m) in the observation plane as illustrated in Figure 3 above.

Among the several challenges, the by far most time-consuming one was encountered after the addition of an optical element. The addition of a converging lens made the need for a higher matrix resolution for further propagation obvious. Any increase in matrix resolution also increased computation times, as the amount of calculations increased exponentially with increasing matrix resolution. This led to computations for propagation including an optical element with a source plane matrix size of 111×111 taking upwards of two days. This immense computation time was untenable though unavoidable with the way the simulation was programmed at the time. The simulation was then reprogrammed so that instead of using loops to select each point contribution from the source plane and add it for each point in the source plane, the source plane was instead vectorized and operations were applied onto the vector. As a consequence all calculations for each point in the observation plane occurring simultaneously rather than relying on loops and led to the removal of two of the nested loops. Despite the fact that the same amount of computations were still performed, performance was greatly enhanced. This was due to the bulk of the reliance on loop iteration now being gone. The time in which the simulation properly performed all of its calculations was thus cut down to approximately 30 minutes, which was a far more useful computational time.

2.2.2 Current Version

All of the above changes ultimately lead to the current version of the simulation. The main distinguishing feature of the new version is that two of the nested loops have been replaced with vector calculations, as shown below. Calculations are only performed between the vector elements with the same index. Then the result of each individual calculation is summed up for each point, the equations below describe how these calculations are input into Python, which takes entire vectors and performs all calculations elementwise. The actual calculations themselves are carried out in much the same way as previously, however this method allows for parallel calculation processes, which shortens computational time.

$$E_{obs\ l,m} = \frac{\mathbf{E}_{source}}{\mathbf{r}} \exp[-ik\mathbf{r}], \quad \mathbf{E}_{source} = \begin{pmatrix} E_{0,0\ source} \\ E_{0,1\ source} \\ \vdots \\ E_{n,n\ source} \end{pmatrix}, \quad \mathbf{r} = \begin{pmatrix} r_{00-lm} \\ r_{01-lm} \\ \vdots \\ r_{nn-lm} \end{pmatrix},$$

As the calculations are carried out, the term on the right-hand side to the right of the sum becomes, as interpreted by Python:

$$\begin{aligned} \frac{\mathbf{E}_{source}}{\mathbf{r}} \exp[-ik\mathbf{r}] &= \frac{E_{0,0\ source}}{(r_{00-lm})} \exp[-ikr_{00-lm}] + \frac{E_{0,1\ source}}{(r_{01-lm})} \exp[-ikr_{01-lm}] + \dots + \\ &+ \frac{E_{n,n\ source}}{(r_{nn-lm})} \exp[-ikr_{nn-lm}] = \sum_{i,j} \frac{E_{i,j\ source}}{r_{ij-lm}} \exp[-ikr_{ij-lm}]. \end{aligned}$$

This means that the calculation that occurs here is equivalent to the previous one, but is executed much more quickly due to the decreased reliance on loops, here represented by sums. The key change between these two methods lies in how the calculations are processed, rather than the calculations themselves.

2.3 Further Considerations

The simulation is rather flexible and it is possible to use as many matrices as is necessary to simulate various effects. Each matrix used, however, adds linearly to the time it takes for the simulation to compute the full beam propagation. Certain matrices contain more matrix points by necessity. As the physical size of the plane the matrix represents is larger, more matrix points are needed to maintain the same, or nearly the same resolution, as the matrices contain an integer amount of points. This is because they are not always scaled by integers, but rather based on the distance between planes. Additionally to be considered is also the possibility of verifying that flux is conserved between matrices after propagation. This would ensure that there is no loss of flux during propagation.

The source plane matrix used for measurements here is as mentioned above 111×111 , which translates to a resolution of approximately $3.437 \mu\text{m}/\text{step}$ using the selected waist width of $60 \mu\text{m}$.

3 Results

3.1 Free Propagation

The free-propagation section of the simulation refers to the part of the simulation that only concerns the propagation of the beam through space without additional optical elements other than possibly an initial spiral phase plate influencing the propagation of the beam, as illustrated in Figure 3. The results of this type of propagation will be discussed here. This is done in order to ensure that the simulation indeed is numerically stable and consistent with analytical formulae for Gaussian beam propagation. The beam used here has a wavelength $\lambda = 800 \text{ nm}$ and a waist width of $60 \mu\text{m}$.

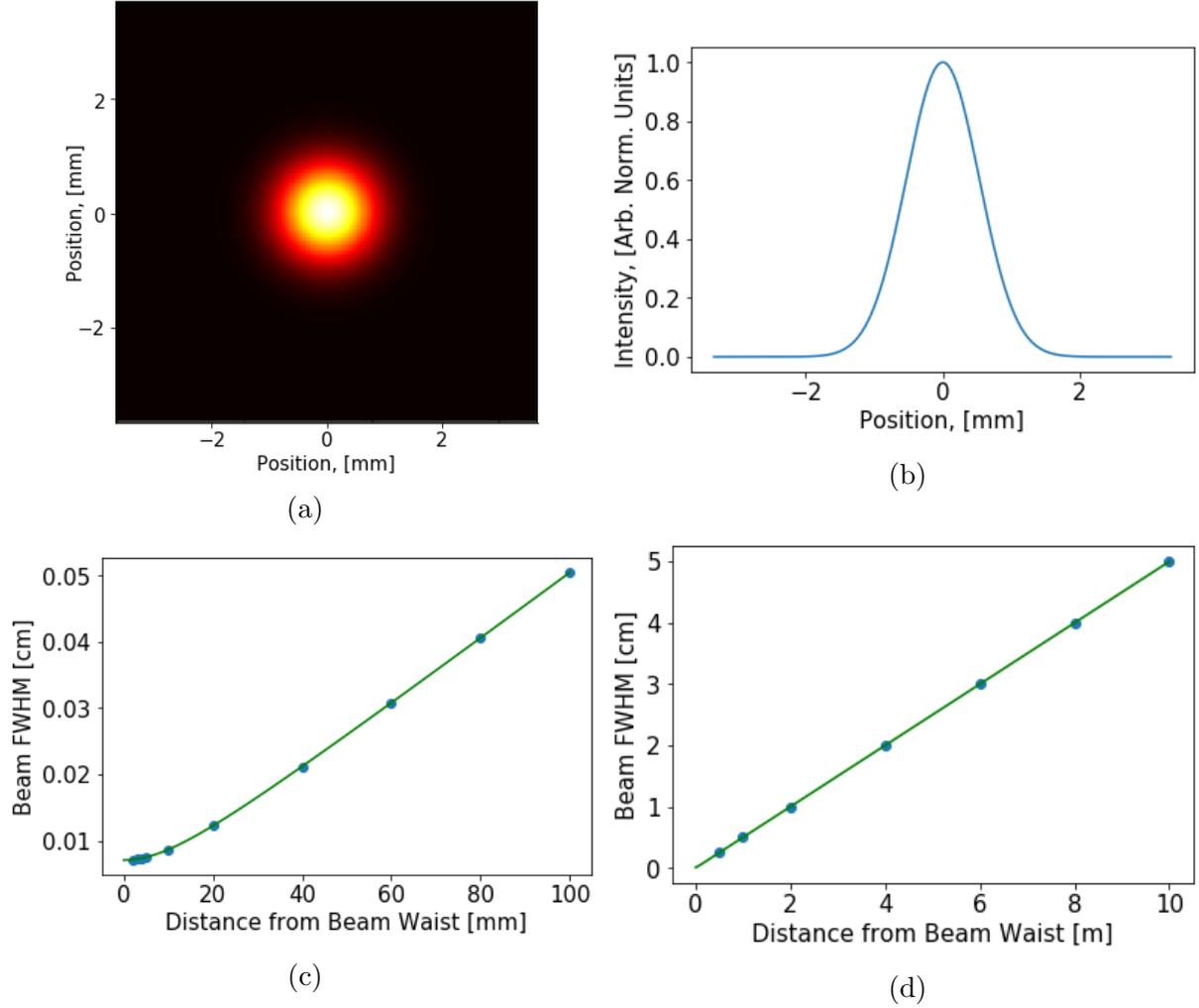


Fig. 4: Intensity distribution of a propagated Gaussian Beam. In (a) one can see the intensity distribution over the whole observation plane, whereas (b) shows a one-dimensional intensity distribution provided by a horizontal cut through the center of the observation plane. In (c) and (d) one can see the measured FWHM of the simulated beam as the blue dots and the theoretical FWHM values can be seen as the green line. For (a) and (b), the distance between planes was set to $d = 25$ cm.

3.1.1 Gaussian Beam

In Figure 4 the intensity distribution of a simple propagating beam can be seen. The width of the observation plane is $2\pi W(z)$. Here, the main purpose is to verify that the beam propagated by the simulation accurately simulates a Gaussian beam and agrees with known analytical formulae for a sufficient matrix width and resolution. This is done through the measurement of the FWHM of the beam at different distances d , the values measured from the simulation were then compared to the theoretical values, as displayed in Figures 4c and 4d.

The closest to the beam waist that the FWHM was measured was with the observation plane at a distance $d = 2$ mm away. Shorter distances were not used, as the resolution of the source plane, i.e. the amount of matrix points per distance, can not support the beam accurately at distances that are too short. The beam used has a Rayleigh range of $z_0 \approx 14$ mm. Using this as an indicator of the location of the near and far fields gives that

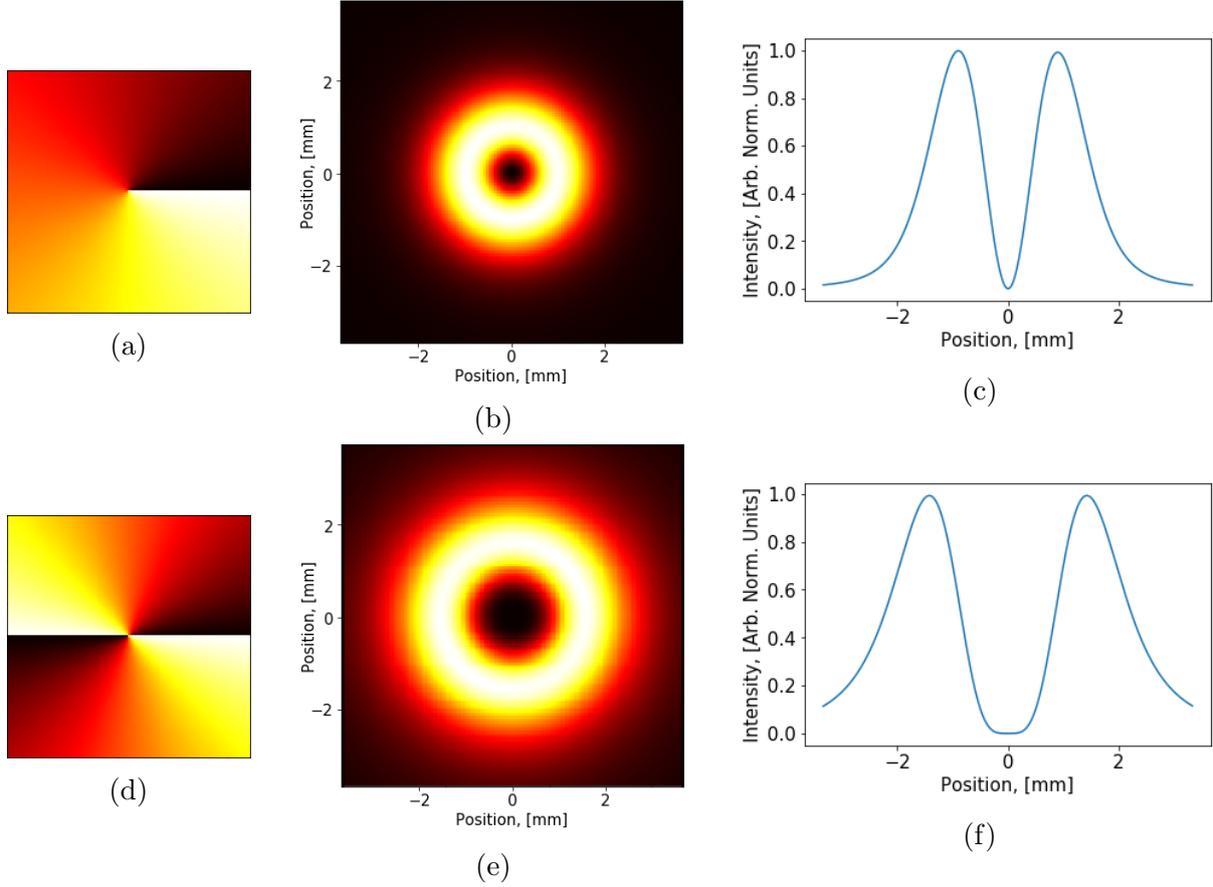


Fig. 5: (a) and (d) illustrate the phase plates applied to give rise to beams carrying $l = 1$ and $l = 2$ units of OAM per photon, respectively. The phase plate illustrated by (a) has values from 0 to 2π increasing with the angular displacement from an arbitrary zero point for one rotation. The same process occurs twice in one rotation in (d), a pattern that is followed for further increasing values of l . (b) and (e) show the intensity distribution across the observation plane of the beams carrying $l = 1$ and $l = 2$ units of OAM, respectively. (c) and (f) show the one-dimensional intensity distributions for the same respective beams, with a horizontal cut through the center of the observation plane, as in Figure 4b. All measurements here were taken at $d = 25$ cm.

there are still values taken within the near field of the source plane. These results indicate that the simulation still works when the beam wavefront is near-flat and not only when it is spherical. The FWHM values given by the simulation, as can be seen in Figures 4c and 4d agree very well with the analytically calculated values, providing evidence that the simulation is accurate and consistent with respect to known theory, allowing us to now investigate beams carrying OAM.

3.1.2 Beam with Orbital Angular Momentum

The spiral phases seen in Figures 5a and 5d, represent the spiral phase plates applied at the beam's source plane, which is how its OAM-characteristics are acquired. There are a few issues with this type of beam, i.e. it has a high dependence on its phase and there is a phase singularity in its central point. This singularity is a point defect, meaning that it propagates as a spherical wave. Consequently, it is not completely accounted for in

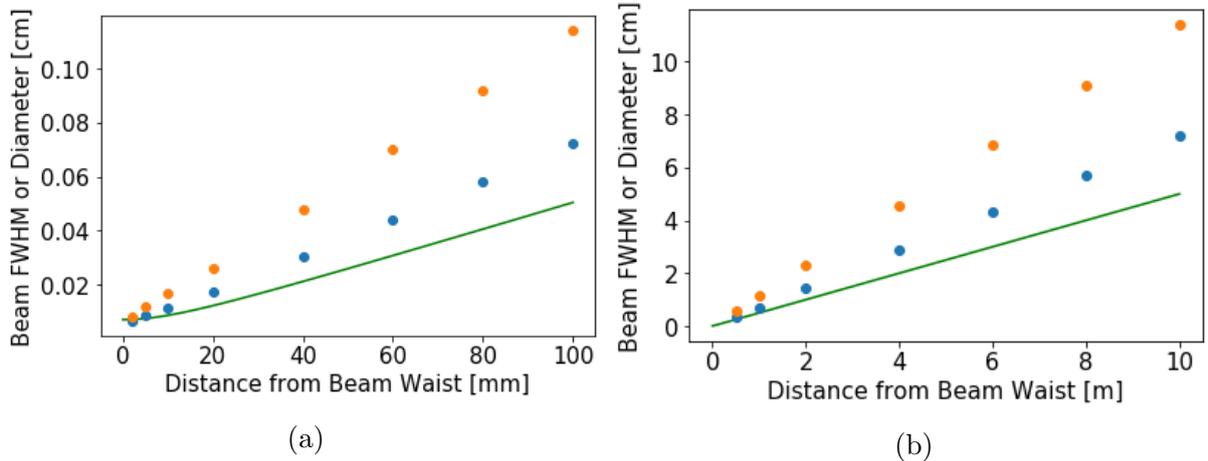


Fig. 6: Pictured are comparisons of the beam divergence of some beams carrying OAM, represented by the blue and orange dots, with the width of the Gaussian beam with the same wavelength and waist width, represented by the green line. Beam width is here represented by the Gaussian beam FWHM, while the blue dots and orange dots represent the ring diameter of maximum intensity of the OAM-carrying beam modes with l -values 1 and 2 respectively.

further propagation which leads to diffraction. This investigation is conducted in order to determine the working conditions for further propagation of OAM-carrying beams and in order to investigate the behavior of OAM-carrying beams in the simulation.

In Figure 5, as in the previous section, the width of the observation plane is given by $2\pi W(z)$ where $W(z)$ is the beam width. As can be observed in Figure 5, the intensity pattern grows wider relative to the plane size (which here is constant) for increasing values of l . This indicates that a beam which carries more OAM diverges more quickly than a beam which carries less OAM. In addition to this, the divergence of the beam is constant regardless of the distance between the source plane and the observation plane some distance away from the beam waist. This means that the intensity patterns seen in Figures 5b and 5e will always appear the same size relative to the observation plane for the given observation plane width. We find that the divergence of a beam carrying OAM is proportional to the Gaussian beam divergence, such that they are related by a constant of proportionality, as expected by Eq. 10.

In Figure 6, a comparison between the beam divergence of the different beams is shown. A very important note to make is that while the FWHM was used to measure the divergence of the Gaussian beam, for the beams carrying OAM the beam diameter given by the radial position of the beam's intensity maximum was used instead initially. This is what was used as it is a well-defined distance for Laguerre-Gaussian beams. This made it easier to compare expected beam divergences, as when comparisons for beam divergences are made between Gaussian and Laguerre-Gaussian beam modes are made, this parameter is used [7]. In later parts of the simulation, beam divergence will be measured using the beam width for OAM-carrying beams as well as it gives a direct comparison between equivalent beam quantities.

As can be seen in Figure 6, the divergence of the OAM-carrying beam is indeed directly proportional to that of the Gaussian beam with a constant of proportionality, with the exception of a short section at a distance from the beam waist corresponding approximately to the Rayleigh range of the Gaussian Beam. The constants given by the ratio of

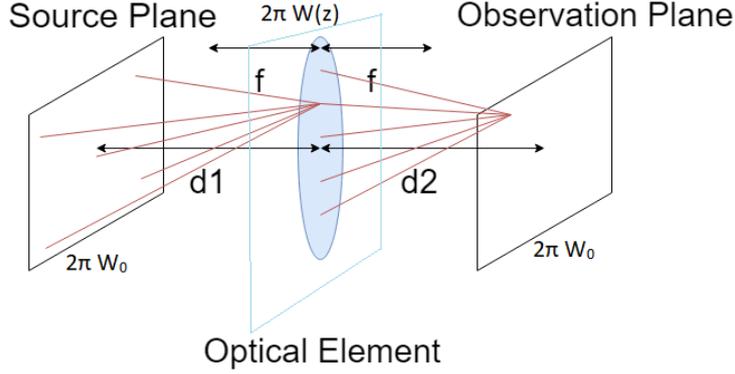


Fig. 7: An illustration of the propagation process used for the second part of the simulation, where an additional plane has been added in the form of an optical element. Here, f represents the focal length of the optical element (here a converging lens), $d1$ and $d2$ represent the distances between the source plane and the optical element plane and the distance between the optical element and the observation plane, respectively. The spatial widths of each matrix is here shown below or above each matrix. In reality, the matrix used is identical to the others in that it has coordinates mapped to each of its matrix points, which store the field data of the incoming beam. The purpose of the optical element matrix is to apply a new phase to the incoming wave, similarly to how the spiral phase is added to the beam at the source plane.

the ring diameter over the FWHM of the Gaussian beam are here used as a measure of relative beam divergence of the OAM-carrying beams. C_{1near} and C_{2near} are the constants measured in the near field of beams carrying $l = 1$ and $l = 2$, respectively, in Figure 6a. Similarly, C_{1far} and C_{2far} represent the constants measured in the far field of the respective beams, which are shown in Figure 6b. $C_{1near} = 1.31 \pm 0.18$, $C_{2near} = 1.95 \pm 0.39$, $C_{1far} = 1.432 \pm 0.0070$ and $C_{2far} = 2.2786 \pm 0.00063$. These constants seem to indicate that the beam divergence has a linear relation to the beam width, something which is given by equation 10 for Laguerre-Gaussian beam modes.

The relatively high standard deviation in the near field is due to the rapid, non-linear decrease in diameter that occurs at the distance from the beam waist approximately corresponding to the Rayleigh range. The beams carrying OAM also seem to display visual artifacts at distances where artifacts are not observed for a Gaussian beam. This is likely caused by diffraction of the point defect at short distances.

On the other hand, the far field constants have a much smaller standard deviation. It appears that the constant is rather consistent in most measurements. The main reason for the low standard deviation is most likely due to discrete nature of the matrix. It is possible that the points where the actual maximum would have been is located between matrix points. If this was the case, then the actual maximum could be approximately found through interpolation. This would probably cause the standard deviation to appear larger than it currently is for the far field calculations. Regardless of the precision, however, the constants in the far field are definitely accurately calculated and they remain consistent throughout the measurements.

In summary, we find that the OAM-carrying beam diverges linearly at long distances which Gaussian beams also do. The constant of proportionality relating the two beams was also found, which can then be applied for further propagation of these beams. However, when passing a Gaussian beam through a spiral phase plate, such as the ones in

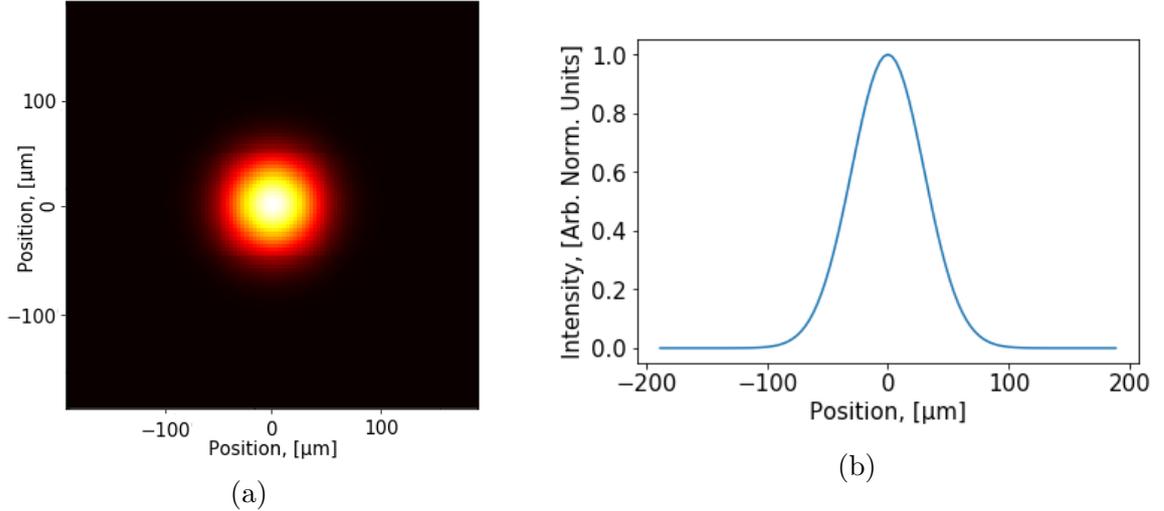


Fig. 8: The intensity distribution of a Gaussian Beam after being refocused through an optical element to a second waist. The distances used were $d_1 = d_2 = 2f = 12.5$ cm, this corresponds to the distances illustrated in Figure 7. (a) shows the intensity distribution in the second waist where the beam is again focused with the same beam waist as the previous beam waist. (b) shows a one-dimensional intensity pattern given by a horizontal cut of the observation plane where $y = 0$.

Figures 5a and 5d, it does not correspond to a Laguerre-Gaussian mode. As a consequence, there is no analytical solution with which to correlate the simulated beam. Despite this, it has been analytically shown that the diameter of maximal intensity of Laguerre-Gaussian beam modes is related to the beam divergence of the Gaussian beam by a constant, as given by equation 10. The fact that the simulated OAM-carrying beams also has a similar constant proportionality to the Gaussian beam seems to indicate that the simulation is to some extent consistent with what one could expect from a theoretical OAM-carrying beam.

At this point the simulation does not replicate an experimental setup though, as the beam is not focused which it would be in an actual experiment. This problem is solved with the addition of an optical element which refocuses the beam. This could be done through simulation of a converging lens, as is done in section 3.2.

3.2 Propagation with an Optical Element

The simulation has now been modified by the addition of an optical element, here in the form of a converging lens, as is illustrated in Figure 7. As previously, the beam light used here has a wavelength of $\lambda = 800$ nm and the beam waist used is $60 \mu\text{m}$. The optical element is scaled such that its width is $2\pi W(z)$, the width of the observation plane is equal to $2\pi W_0$ and the distances involved, referred to by their names from Figure 7, are $d_1 = d_2 = 2f = 25$ cm. These distances were picked so that the observation plane would be located at the secondary beam waist, caused by the optical element within the paraxial approximation.

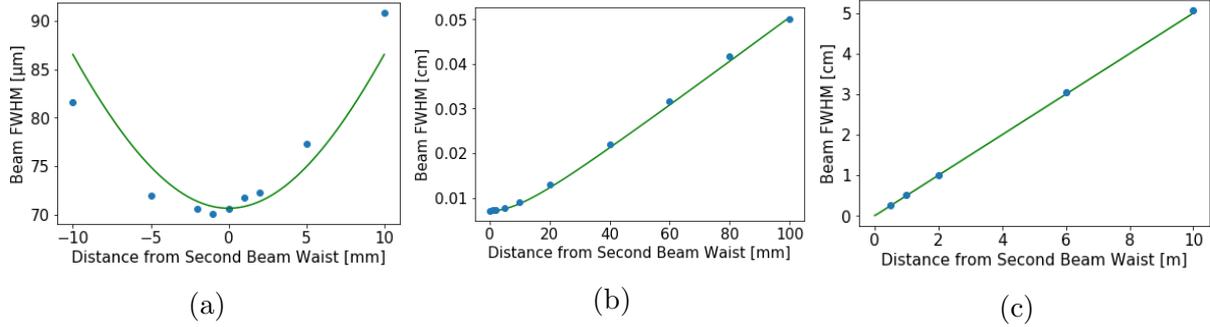


Fig. 9: Measurements of the FWHM of the beam around its second waist. The blue dots represent measurements taken of the simulated Gaussian around its second waist. The green line represents the theoretical values of the original beam FWHM around its waist. Shown in (a) is a small region around where the second beam waist should be. In (b) and (c), distances in the relative near and far field of the second beam waist is shown, respectively.

3.2.1 Gaussian Beam Propagating Through an Optical Element

As seen in Figure 8 the Gaussian beam appears to retain its Gaussian characteristics after propagation through the optical element, with both Figures 8a and 8b appearing to have the same structure as before. In order to verify that the optical element is working as intended, the FWHM of the Gaussian beam is measured at points around the secondary beam waist and compared with the theoretical FWHM values of the original Gaussian beam. Because of the way the distances and focal range of the lens are chosen, more specifically because $d_1 = d_2 = 2f$, the optical element should be equidistant from the source and observation planes, which are of the same width in this case.

The measured FWHM of the refocused beam is plotted alongside the theoretical FWHM in Figure 9. As seen in Figures 9b and 9c, the simulated FWHM agrees well with the theoretical far field, until within the Rayleigh range of the secondary waist. It seems that the simulation rather accurately simulates a Gaussian beam after propagation through a converging lens. However, Figure 9a indicates that there is a slight discrepancy very close to the secondary waist. The simulated waist is slightly thinner than it theoretically should be. The waist also appears displaced around 1 mm in the direction of the optical element, with respect to the theoretical secondary beam waist. This may have to do with the $2f$ -imaging used ($d_1 = d_2 = 2f$) not necessarily being precisely applicable to a Gaussian beam, due to it propagating as a paraxial wave, rather than a spherical or plane wave. With this in mind, the simulation can also be tested for beams carrying OAM, which was the main intent of the simulation initially.

3.2.2 Beam With Orbital Angular Momentum Propagating Through an Optical Element

As before, the spiral phases of Figures 5a and 5d are applied to the Gaussian beam in the source plane. Aside from this, the remainder of the procedure is carried out identically to that of the Gaussian beam propagation through the optical element. Here the beams are no longer as ring-shaped in Figures 10a and 10c as previously. One can also observe a "wing" structure in Figures 10b and 10d outside of the primary peaks, which look like smaller, secondary peaks. This is caused by diffraction of the point

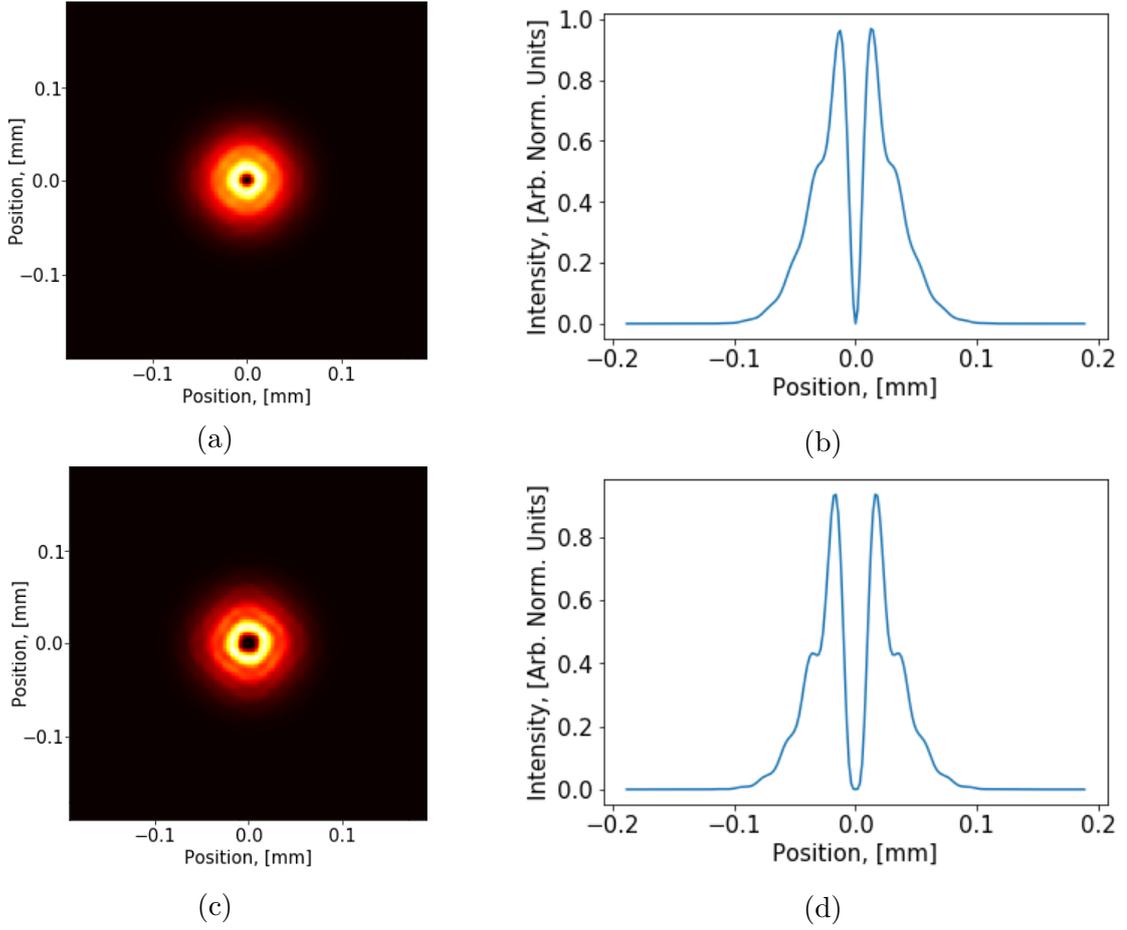


Fig. 10: Shown are the intensity patterns of the refocused OAM-carrying beams. (a) and (b) show the intensity distribution over the whole observation plane and a horizontal one-dimensional cut of the intensity pattern of the $l = 1$ beam. (c) and (d) show the same patterns, but for an $l = 2$ beam.

defect after propagation through the optical element, which is why we see no such thing for the Gaussian beam in the previous section. However, in Figures 10a and 10c some slight rectangular disturbances in the intensity pattern can be seen. The effects of diffraction here have been mostly avoided, due to the adjustments of the size of the plane corresponding to the lens. As the lens is 25 cm away from the original beam waist, for the purposes of rescaling the matrix the far field constants from before were used. Using Equation 9 gives the matrix width $2\pi\sqrt{2\ln(2)}C_{far}W(z)$ for the observation plane. This matrix width is the one that was used to take measurements. It then appears as if the beam's deviation from a regular ring shape is caused by an insufficient resolution. When the simulation was done with a lower resolution than the one used in Figure 10, which was done with the matrix dimensions 211×211 , the beam appears less ring-shaped, which indicates that this is mainly an issue of discretization.

Measurements were then made of the beam divergence, initially using the method of finding the diameter of highest intensity with comparison to a theoretical FWHM of a Gaussian beam as before. However, these quantities are not equivalent and ultimately the definition of beam width as the radius within which 86 % of the beam's power is contained was used for both the Gaussian and OAM-carrying beams to compare their divergence. This is illustrated in figures 11a and 11b. While this width is harder to

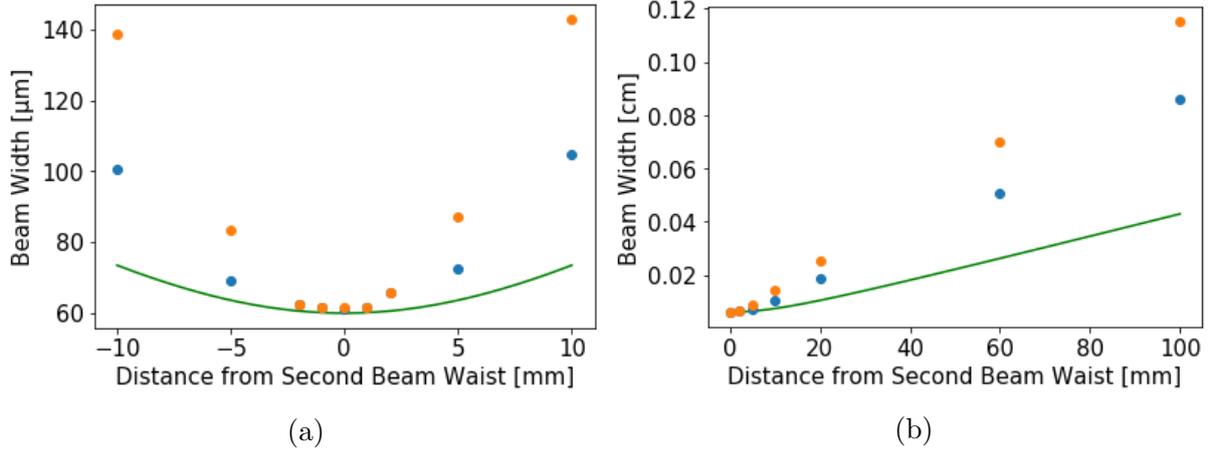


Fig. 11: Measurements of beam divergence of beams carrying OAM compared to that of the beam width of a theoretical Gaussian beam, where the blue and orange dots represent the beam carrying $l = 1$ and $l = 2$ units of OAM per photon respectively.

measure practically than the radius of maximal intensity, it is not useful for distances close to the secondary waist as the radius of maximal intensity decreases rapidly, but the beam width does not. The actual beam width measurements, as seen in Figure 11, show that the beam width of the OAM-carrying beam exhibits a decrease and increase near the waist very similar to that of the theoretical Gaussian beam width, although the OAM-carrying beam width decreases and increases more drastically. Although it is difficult to discern from Figure 11a, the beam waist width is slightly larger for higher orders of OAM-carrying beams, with the Gaussian beam having the thinnest waist. In Figure 11b it is possible to observe the linear relation between the beam widths that was observed before as well, although here the widths used are physically equivalent between the beams. Measurements for the $l = 2$ beam, were not reported at distances further away than 50 cm. This is because the intensity pattern here was heavily distorted for this beam, which was unexpected. Although this behavior is an issue, nothing similar occurs close to the second waist, which is where the simulation is mainly intended to be used.

Using the radius of an OAM-carrying beam works rather well in the far field as the OAM-carrying beam there behaves more similar to a Laguerre-Gaussian beam, which is what made it possible to use before. However, so close to the secondary beam waist it is necessary to use the regular definition of beam width in order for the simulation to be of use. In Figure 12, the radii of the simulated OAM-carrying beams are compared to their equivalent Laguerre-Gaussian beam modes given the same radius at the waist. As can be observed from Figure 12 the radius of the simulated OAM-carrying beam decreases and increases very differently near the beam waist in comparison with the Laguerre-Gaussian beam modes. The radii of the simulated beams also diverge more in the far field than those of the Laguerre-Gaussian beams. This is rather clear evidence that the OAM-carrying beams that are simulated are not equivalent to the individual Laguerre-Gaussian beam modes.

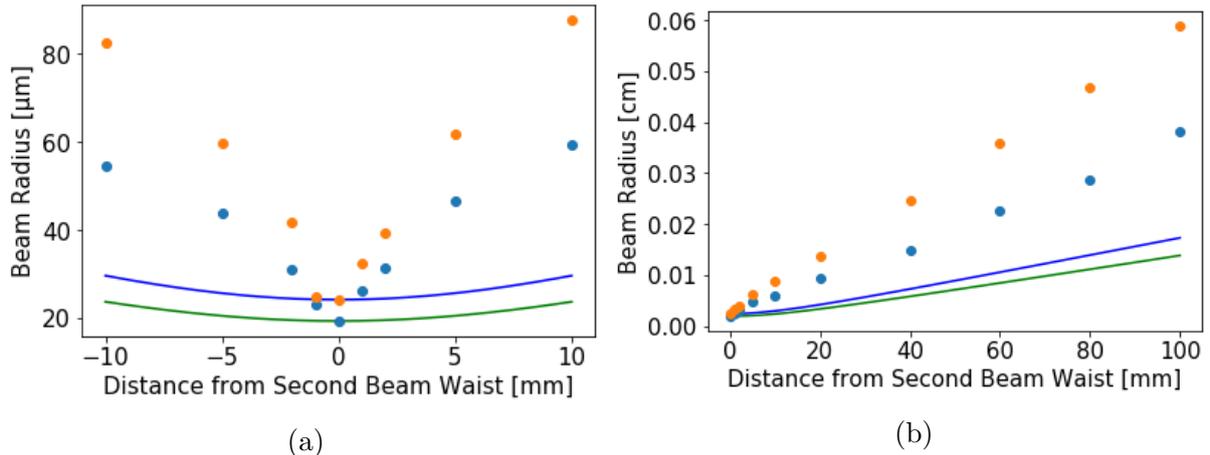


Fig. 12: Measurements of the beam radii of OAM-carrying beams are here compared to the radii of two theoretical Laguerre-Gaussian modes. The blue and orange dots represent the beam carrying $l = 1$ and $l = 2$ units of OAM per photon respectively and the green and blue line represent the Laguerre-Gaussian beam modes with $l = 1$ and $l = 2$ respectively. The waist widths of these theoretical beams were adjusted so that their radii at the waist align with their simulated counterparts. The rapid decrease and increase in beam diameter seen in Figure 6a is also observed here in (a) and to a lesser extent in (b).

4 Outlook

The simulation appears to work well when freely propagating a beam of light, with the simulated FWHM-values very close to the analytically calculated FWHM of a Gaussian beam. The beam divergence of the OAM-carrying beams also appear to be linear in the far field, which is consistent with what was expected of OAM-carrying beams. As one moves to the propagation with an optical element, one can observe that the intensity distribution appears to be incorrectly shaped primarily due to an insufficient resolution of the matrices involved. This can be resolved through an increased discretization of the matrices. The measurements of the beam width seem to behave as expected from the unmodified beam results within the range of the simulation. This indicates that the simulation is rather well-suited to simulate a Gaussian or OAM-carrying beam to a second beam waist, as it was intended. With this considered seems that the simulation is able to achieve its goal to some extent, despite some issues. That is, to accurately be able to simulate a beam, be it Gaussian or OAM-carrying, for use in assisting experiments near its second waist, where different parts of the volume of interaction can be observed, which is the main area of interest to this work.

The simulation provides not only the intensity distribution, but also the field strength at any given position in one of the used planes. This additionally gives access to the possibility of observing the relative phase of different positions in space of the beam. It is possible to run the simulation for different OAM-modes of the beam and observe several phases at different points. This would be done in order to observe the combined field strength of several beams carrying different amounts of OAM and with differing wavelengths, such as those generated by high-harmonic generation.

Continued work on this project would likely be focused around improving the computational time. This could be accomplished by increasing the amount of calculations using vectors, as this allows for parallelization of calculations. When possible, this paral-

lization method is a much more efficient way of doing many simultaneous calculations than looping the process for every point in a matrix, as is done now. An increase in computational capacity for the simulation has several benefits, not only in that it allows current calculations to be done faster, but it also allows for further discretization of matrices. Additional discretization for the purposes of accuracy of the simulation is always a benefit. It would also allow for a greater spatial resolution, which is also an important consideration, considering the scale on which the simulation is intended to work on, on which a clear resolution on the order of nanometers is necessary, which is best achieved with increased discretization.

Another way in which the simulation might be improved for further use is to add the functionality of propagating several beams at a time. As it currently stands, the simulation allows for only one beam to be propagated at a time. This means that all calculations made with one beam must be saved before one propagates an additional beam, the results gained from propagation must then be superimposed with previous results, which should grant the same result as propagating the beams simultaneously. The process would be expedited and simplified by the possibility of propagating several beams at once. This is especially true in cases discussed before, within attosecond science, where each harmonic with its own wavelength and OAM would be represented along with the driving infrared beam. Doing such calculation without the possibility of propagating all beams at once would prove to be highly impractical, as the data of each simulation would need to be saved and the simulation would need to be restarted once per harmonic. Hence support for additional beams should be considered for future work.

While the simulation seems like it has potential, more work is needed to be put into it before it can be brought into actual experimental work as it currently can only be used for visualizing a plane in the volume of interaction. However, it does seem like it can be further developed into a powerful numerical tool for visualizing the volume of interaction between light and matter upon further development.

5 References

- [1] Allen, L., Beijersbergen, M. W., Spreeuw, R. J. C. and Woerdman, J. P. Orbital angular momentum of light and the transformation of Laguerre-Gaussian laser modes. *Phys. Rev. A* 45, 81858189 (1992).
- [2] Fürhapter, S., Jesacher, A., Bernet, S. and Ritsch-Marte, M. Spiral interferometry. *Opt. Lett.* 30, 19531955 (2005).
- [3] Wang, J. et al. Terabit free-space data transmission employing orbital angular momentum multiplexing. *Nat. Photon.* 6, 488496 (2012).
- [4] Pariente, G. and Quéré, F. Spatio-temporal light springs: extended encoding of orbital angular momentum in ultrashort pulses. *Opt. Lett.* 40, 20372040 (2015).
- [5] Hernández-García, C., Picón, A., San Román, J. and Plaja, L. Attosecond extreme ultraviolet vortices from high-order harmonic generation. *Phys. Rev. Lett.* 111, 083602 (2013).
- [6] Kroon, David (2016). Attosecond Interferometry: Techniques and Spectroscopy, *Department of Atomic Physics, Lund University*. (2016) ISBN 978-91-7623-688-8.
- [7] Géneaux, R. et al. Synthesis and characterization of attosecond light vortices in the extreme ultraviolet. *Nat. Commun.* 7:12583 doi: 10.1038/ncomms12583 (2016).
- [8] Bahaa, E. A. Saleh and Malvin, Carl Teich, (2007), Fundamentals of Photonics, Third Printing. *John Wiley and Sons*. ISBN 978-0-471-35832-9.
- [9] Hill, Dan (April 4, 2007). How to Convert FWHM Measurements to 1/e-Squared Halfwidths. *Radiant Zemax Knowledge Base*. Retrieved June 7, 2016. Available from: <http://customers.zemax.com/os/resources/learn/knowledgebase/how-to-convert-fwhm-measurements-to-1-e-squared-ha>
- [10] Plick, W. and Krenn, M. Physical meaning of the radial index of Laguerre-Gauss beams. *Phys. Rev. A* 92, 063841 (2015)

Appendix: Python Simulation Code

```
import math as math
import numpy as np
import matplotlib.pyplot as plt

#Physical constants
wavl=800.*10**(-9) #Wavelength of wave used
k=2.*np.pi/wavl #Wavenumber of light
n=111 #dimensions of matrix
nn=float(n)
z=0.25 #distance between "incoming" and "lens", and "lens" and "outgoing" planes
A1=1. #beam constant
Wwidth=6.*10**(-5) #Waist width of beam
z0=np.pi*(Wwidth**2)/(wavl) #the Rayleigh range of the incoming beam
l=0 #the orbital angular momentum number of the beam
f1=0.125 #the focal length of the first lens
bd=1.0 #distance of the observation plane from the theoretical second beam waist

#Programming constants
I=np.zeros((n,n),dtype=np.complex)#matrix plane with "incoming" waves
Pos=np.zeros((n,n),dtype=np.complex)#matrix with positions of elements in I and O
#the real part represents x-coordinates, imaginary part represents y-coordinates
Phas=np.zeros((n,n),dtype=float)#matrix with phase of different elements in I
O=np.zeros((n,n),dtype=np.complex)#matrix with "outgoing" waves at target plane
OGauss=np.zeros((n,n),dtype=np.complex)#matrix with "outgoing" waves for Gaussian
TPos=np.zeros((6*n-5,6*n-5),dtype=np.complex)#matrix with positions on target plane
PhasO=np.zeros((n,n),dtype=float)#phase at outgoing plate
Osquare=np.zeros((n,n),dtype=np.complex)# matrix with intensity at outgoing plate
Lens1=np.zeros((6*n-5,6*n-5),dtype=float)#matrix with a phase made to simulate a
lens (the first)
LensPlane1=np.zeros((6*n-5,6*n-5),dtype=np.complex)#matrix in the plane
of the lens, used for calculations on the way to the outgoing plate
EPos=np.zeros((n,n),dtype=np.complex)#matrix with positions in the final obser-
vation plane
t=np.array([0.0j]) #empty array for complex number addition

def EnWeight(Inc=np.zeros((n,n),dtype=np.complex),xmat=0,yamat=0,l=0.01,A0=(-
1.0j*A1/z0),w0=Wwidth): #Calculates the energy weighting of the point emitters
return A0*(w0/(BeamWidth(l,w0)))*np.e**(-((Inc[xmat,yamat].real)**2+(Inc[xmat,yamat].imag)**2)/((

def BeamWidth(l=0.01,w0=Wwidth):#calculates the beam width of the gaussian beam
at distance l from waist
return w0*np.sqrt(1+(l/z0)**2)

def BeamRadOfC(l=0.01):#calculates the radius of curvature of the gaussian beam at
distance l from waist
```

```

return l*(1+(1/z0)**2)
def GouyPhas(l=0.01):#calculates the Gouy phase of the beam for a set distance from
the waist
return math.atan(1/z0)

def FWHM(l=0.01,w0=Wwidth): #calculates the full width at half maximum of a
gaussian beam
return BeamWidth(l,w0)*np.sqrt(2*np.log(2))

def PointCalc(xy=0j,l=z):#calculates the distance between a point on I to (0,0) on O
return np.sqrt(l**2 + (xy.real)**2 + (xy.imag)**2)

def DistCalc(xy=0j, xpos=int((n-1)),ypos=int((n-1)),l=z,pos=TPos): # uses position
matrix to calculate distance between points on the two plates
return np.sqrt(l**2 + (xy.real-pos[xpos,ypos].real)**2 + (xy.imag-
pos[xpos,ypos].imag)**2)

def GaussWave(l=10**(-9), xpos=int((n-1)),ypos=int((n-1))):
return ((-1.0j*A1/z0)*((np.sqrt(wavl*z0/np.pi))/((np.sqrt(wavl*z0/np.pi)*np.sqrt(1+(1/z0)**2))))*(np
abs(TPos[xpos,ypos])**2/((np.sqrt(wavl*z0/(np.pi))*np.sqrt(1+(1/z0)**2))**2))*np.e**(-
1.0j*k*l+1.0j*np.arctan(1/z0)-(1.0j*k*((np.abs(TPos[xpos,ypos])**2)/(2.*l*(1+(z0/l)**2))))))

def PhasCalc(xy=0j, l=10**(-9)): #calculation of phase on different parts of the I plane
return (k*l-np.arctan(1/z0)+(k*((xy.real)**2 + (xy.imag)**2)/(2.*l*(1+(z0/l)**2))))

def EnergyWave(r,Eo=2.,phase=0.,kw=k): # wave equation for a single "point spherical
emitter" of optical waves
return ((Eo/r)*np.e**(-1.0j*(kw*r)+1.0j*(phase)))

def LensCalc(xcoord=0.,ycoord=0.,fl=f1,wavn=k):#returns the phase added to the wave
for passing through a lens
return (wavn*(xcoord**2+ycoord**2)/(2.*fl))

#For calculating the sum of all waves at a point in the "O" matrix using huygen's
principle, so no interference, apart from center
def EnergyOutCalc(Inc=np.zeros((n,n),dtype=np.complex),E0=np.zeros((n,n),dtype=np.complex),xcoo
1)/2),ycoord=int((n-1)/2),d=z,PH=np.zeros((n,n),dtype=np.complex),pos=TPos,rang=n):
E=0
for i in range(rang):
for l in range(rang):
r=DistCalc(Inc[i,l],xcoord,ycoord,d,pos)
E=E+EnergyWave(r,E0[i,l],PH[i,l])
return(E)

#Same as EnergyOutCalc, only using vectors instead of for loops
def EnergyOutVecCalc(Invec,Posvec,Phasvec,pos=TPos,x=0,y=0,l=z):
E=Invec
r=DistCalc(Posvec,x,y,l,pos)

```

```

E=EnergyWave(r,Invec,Phasvec)
return(sum(E))

Inx=-np.pi*Wwidth#x- and y-positions for the "I" matrix point (0,0)
Iny=np.pi*Wwidth

Intx=-np.pi*BeamWidth(z,Wwidth) #x- and y-positions for the target plate "O"(0,0)
Inty=np.pi*BeamWidth(z,Wwidth)

Inttx=-np.pi*BeamWidth(bd,Wwidth)#x- and y-positions for the "I" matrix point (0,0)
Intty=np.pi*BeamWidth(bd,Wwidth)

nx=Inx# x- and y-positions for matrix points (used in loop below)
ny=Iny
ntx=Intx
nty=Inty
nttx=Inttx
ntty=Intty

stepdistx=-Inx/(((nn-1.))/2.) #step distance calculations used in the loop below
stepdisty=-Iny/(((nn-1.))/2.)

stepdisttx=-Intx/((6*nn-5.)/2.) #step distance calculations used in the loop below
stepdistty=-Inty/((6*nn-5.)/2.)

stepdistttx=-Inttx/(((nn-1.))/2.) #step distance calculations used in the loop below
stepdisttty=-Intty/(((nn-1.))/2.)

for i in range(n): #this loop assigns a coordinate value to each point in the matrix, also
assigns values to the phase plate matrix and the lens matrix
for j in range(n):
Pos[i,j]= t+nx+1.0j*ny
EPos[i,j]= t+nttx+1.0j*ntty
if l == 0:
Phas=Phas
elif l % 2 != 0:
if math.atan2(Pos[i,j].imag,Pos[i,j].real)>0:
Phas[i,j]=l*(math.atan2(Pos[i,j].imag,Pos[i,j].real)+2*np.pi)
else:
Phas[i,j]=l*(math.atan2(Pos[i,j].imag,Pos[i,j].real))
else:
if math.atan2(Pos[i,j].imag,Pos[i,j].real)>0:
Phas[i,j]=l*(math.atan2(Pos[i,j].imag,Pos[i,j].real)+np.pi)
else:
Phas[i,j]=l*(math.atan2(Pos[i,j].imag,Pos[i,j].real))
nx=nx+stepdistx
nttx=nttx+stepdistttx
ny=ny+stepdisty

```

```

ntty=ntty+stepdisttty
nx=Inx
nttx=Inttx

for i in range(n): #this loop assigns an "E0" value to each point in the matrix
for l in range(n):
I[i,l]=t+EnWeight(Pos,i,l,0)

E0vec=I.flatten()
r0vec=Pos.flatten()
phasvec=Phas.flatten()

for i in range(6*n-5):#this loop assigns coordinates to the "target" matrix
for l in range(6*n-5):
TPos[i,l]=t+ntx+1.0j*nty
Lens1[i,l]=LensCalc(TPos[i,l].real,TPos[i,l].imag,f1,k)
ntx=ntx+stepdisttx
nty=nty+stepdisttty
ntx=Intx

#counters for the plotting functions
Xplot=np.empty(6*n-5)
Yplot=np.empty(6*n-5)
nxcont=0
nycont=0

for i in range(6*n-5):# This loop gives the lens matrix the interference values at each
point in the matrix
for l in range(6*n-5):
LensPlane1[i,l]=EnergyOutVecCalc(E0vec,r0vec,phasvec,TPos,i,l,z)
if nxcont%6*n-5:
Xplot[nxcont]=TPos[i,l].real
nxcont=nxcont+1 if nycont%6*n-5:
Yplot[nycont]=TPos[i,l].imag
nycont=nycont+1

E1vec=LensPlane1.flatten()
r1vec=TPos.flatten()
lensvec=Lens1.flatten()

for i in range(n):# This loop gives the lens matrix the interference values at each point
in the matrix
for l in range(n):
O[i,l]=EnergyOutVecCalc(E1vec,r1vec,lensvec,EPos,i,l,z+bd)
if math.atan2(O[i,l].imag,EPos[i,l].real)>0:
PhasO[i,l]=(math.atan2(O[i,l].imag,O[i,l].real)+2*np.pi)
if math.atan2(O[i,l].imag,EPos[i,l].real)<=0:
PhasO[i,l]=(math.atan2(O[i,l].imag,O[i,l].real))

```

```

if nxcont;n:
Xplot[nxcont]=TPos[i,l].real
nxcont=nxcont+1
if nycont;n:
Yplot[nycont]=TPos[i,l].imag
nycont=nycont+1
Osquare=abs(O)**2
Osquare=Osquare/(np.max(np.max(Osquare,axis=1))) #This normalizes the intensity
values with respect to the peak

#Code for plotting the 2D heat-map and the "slice" of intensity along the center of
the plate
plt.rcParams.update({'font.size': 15})

#2d Heat map

LensSquare1=abs(LensPlane1)**2
LensSquare1=LensSquare1/(np.max(np.max(LensSquare1,axis=1)))

plt.imshow(LensSquare1, cmap='hot', interpolation='nearest')
plt.show()

plt.imshow(Osquare, cmap='hot', interpolation='nearest')
plt.show()

plt.imshow(PhasO, cmap='hot', interpolation='nearest')
plt.show()

plt.imshow(Phas, cmap='hot', interpolation='nearest')
plt.show()

#"Intensity slice" plot

Intslice=LensSquare1[int((6*n-5)/2),0:6*n-5].real
Posslice=TPos[int((6*n-5)/2),0:6*n-5].real
plt.plot(Posslice*1000,Intslice)
plt.ylabel('Intensity, [Arb. Norm. Units]')
plt.xlabel('Position, [mm]')
plt.show()

Intslice=Osquare[int((n-1)/2),0:n].real
Posslice=EPos[int((n-1)/2),0:n].real
plt.plot(Posslice*1000,Intslice)
plt.ylabel('Intensity, [Arb. Norm. Units]')
plt.xlabel('Position, [mm]')
plt.show()

```