Simple computer program to calculate arbitrary tightly focused (propagating and evanescent) vector light fields

Isael Herrera¹ and Pedro A. Quinto-Su²

¹Instituto de Ciencias Nucleares, Universidad Nacional Autónoma de México, Apartado Postal 70-543, 04510, Cd. Mx., México.

In this work we present a simple code to calculate tightly focused vectorial light fields (propagating and evanescent) generated by input fields that have arbitrary amplitude, phase and polarization. The program considers results from previous studies, like integration via fast Fourier transforms to speed up the integration. The calculations are done in a Cartesian coordinate system that is convenient to compare with experimental results for beams that are shaped with programmable optical elements like spatial light modulators or digital micromirror arrays. We also discuss how to avoid diverging terms at the origin by shifting the angular mesh by half a point and correcting the output by cancelling the phase term that arises from the shifted Fourier transform.

INTRODUCTION

In general, tightly focused light has significant polarization components in the 3 spatial directions and can be described by the vectorial model that Richards and Wolf developed in 1956 [1]. Their model has become even more relevant, as some of the most important optical applications rely on tightly focused laser light.

It is now possible to experimentally generate beams with arbitrary amplitude, phase and polarization, which upon tight focusing can result in very complex structures with rapid changes in amplitude and phase at subwavelength spatial scales in each polarization component. So, the ability to compare the measurements with the calculations while performing the experiment can help identifying the focal plane and possible errors in the input field (like the size at the aperture). For example, for beams that have radial or azimuthal symmetry, simply overfilling the back aperture of the microscope objective might be sufficient to get good agreement with the calculation. However, when the input beams do not possess those symmetries, it is crucial to check the precise spatial dimensions and the equivalence between spatial and angular coordinates over the lens aperture.

In general, the numerical integration in the Richards-Wolf model requires several minutes in a laptop computer to calculate the field at a single transverse spatial plane in the vicinity of the focus. A crucial step to speed up the calculation (by a factor \(\sim 30-100\)) has been the use of the fast Fourier transform algorithm to perform the numerical integration [2].

Recently, a couple of open source codes that calculate tightly focused beams have been published: Infocus [3] and Pyfocus [4], which have helped increase the availability of these tools to experimental groups. Infocus considers propagating fields and the calculations were compared with intensity measurements of beams focused with lenses with numerical apertures in the range between 0.4 – 0.7. Pyfocus is more oriented toward microscopy and intensity distributions, it also considers straight boundaries and evanescent fields. However, both codes describe the fields in cylindrical coordinates and do not consider completely arbitrary vector fields at the input. Also, there are still some details that can be improved in order to simplify the way arbitrary fields are described, matching the geometry of the experimental devices that are used to generate the beams and enable direct point by point comparisons with the experimental measurements.

In this work we provide a simple MATLAB code based on previous results [1,2,5] to calculate arbitrary tightly focused vectorial fields. In the Code (Appendix) we replaced the traditional spherical coordinate system by Cartesian coordinates (spatial and angular) as in [3,6]. This approach helps matching the description of arbitrary fields to those in the experiments, where the structured light is generated with computer controlled rectangular arrays like spatial light modulators (SLM) or digital micromirror arrays (DMD). Furthermore, Cartesian coordinates are the most suitable to express the beams as a superposition of plane waves (in the angular spectrum representation) and for the Fast Fourier transform.

The article is presented in the following way: we start with a summary of the Richards-Wolf model in section 2, then we present a simplified optical system to generate these beams and how it relates to the calculation. In section 4 we describe the numerical implementation which is based on [2] and discuss in detail the construction of the integration mesh, the correction of artifacts and the spatial resolution associated with the physical and computational parameters. This discussion is also useful for simulating diffraction and focusing of paraxial light fields. Section 5 contains several examples with different input polarization states including the small difference between a corrected and uncorrected field which is a mistake which is commonly made. Finally, in section 6 we present the case of a planar interface with an example. The computer programs are in the Appendices.
the light, the refractive index is lens (Figure 1). Inside the spherical surface that focuses beam propagating in a medium with refractive index $n$. FIG. 1. Representation of the focusing model. A paraxial gates from left to right in air ($E$ rays in geometrical optics) corresponds to the far field the field at the lens surface (represented by conjugate energy conservation, the sine condition, and the fact that $E$ is the unitary vector in the $\phi$ direction.

The relations between the spherical and Cartesian unitary vectors are:

$$n_\phi = -\sin(\phi)n_x + \cos(\phi)n_y$$

$$n_\rho = \cos(\phi)n_x + \sin(\phi)n_y$$

$$n_\theta = \cos(\theta)\cos(\phi)n_x + \cos(\theta)\sin(\phi)n_y - \sin(\theta)n_z$$

If the incident field is linearly polarized in the $x$ direction, it can be expressed as $E_{inc}^x = E_{0inc}e^{i\phi_0}n_x$, where $E_{inc}$ and $\phi_0$ are the amplitude and the phase and $n_x$ is the unitary vector in the $x$ direction. Upon focusing, the resulting field has polarization components along the 3 spatial directions. The far field representation of the $x$ component of the focused field as a function of the incident field is $E_{\infty}^x$, which has the 3 polarization components (in a Cartesian basis).

$$E_{\infty}^x = \sqrt{\frac{n_1}{n_2}} \cos(\theta)E_{inc}^x \begin{pmatrix} -\sin(\phi) & -\sin(\phi) & 0 \\ \cos(\phi) & \cos(\phi) & 0 \\ \sin(\phi) & \sin(\phi) & -\sin(\phi) \end{pmatrix}$$

$$E_{\infty}^y = \sqrt{\frac{n_1}{n_2}} \cos(\theta)E_{inc}^y \begin{pmatrix} \cos(\phi) & \cos(\phi) & 0 \\ -\sin(\phi) & \sin(\phi) & 0 \\ \cos(\phi) & \cos(\phi) & -\sin(\phi) \end{pmatrix}$$

While for the case of an incident field $E_{inc} = E_{0inc}n_y$ with $E_{inc}^y = E_{0inc}e^{i\phi_0}n_y$. The $E_{\infty}^y$ representation of the focused field is:

$$E_{\infty}^y = \sqrt{\frac{n_1}{n_2}} \cos(\theta)E_{inc}^y \begin{pmatrix} \cos(\phi) & \cos(\phi) & 0 \\ \sin(\phi) & \sin(\phi) & \cos(\phi) \end{pmatrix}$$

Equations 2 and 3 can be rewritten in Cartesian coordinates using:

$$\frac{x_{\infty}}{f} = \frac{k_x}{k} = \sin(\theta)\cos(\phi)$$

$$\frac{y_{\infty}}{f} = \frac{k_y}{k} = \sin(\theta)\sin(\phi)$$

$$\frac{z_{\infty}}{f} = \frac{k_z}{k} = \cos(\phi)$$

where $k = nk_0$ ($k = k_1$ in sections 4-6) and $k_0 = 2\pi/\lambda$, where $n$ is the refractive index and $\lambda$ the wavelength in vacuum.

The expressions for $E_{\infty}^x$ and $E_{\infty}^y$ are:

$$E_{\infty}^x = \sqrt{\frac{n_1}{n_2}} \frac{k_x}{k} E_{inc}^x \begin{pmatrix} k_x^2 + k_y^2/2k_z/k -k_xk_y + k_xk_y/k \\ -k_xk_y - k_xk_y/k \\ -k_xk_y - k_xk_y/k \end{pmatrix} \left( \frac{1}{k_x^2 + k_y^2} \right)$$

Finally the focused field at $(x, y, z)$ can be written as:

$$E(x,y,z) = -\frac{ikf\epsilon e^{-ikf}}{2\pi} \int_{k_z^2 + k_y^2 \leq k_{max}^2} [c_x E_{\infty}^x + c_y E_{\infty}^y] \frac{1}{k_z} \times \int_{k_z^2 + k_y^2 \leq k_{max}^2} e^{i[k_xz + k_yy + k_z]} dk_x dk_y$$

$$\frac{1}{k_z^2 + k_y^2} \sum_{k_z^2 + k_y^2 \leq k_{max}^2} [c_x E_{\infty}^x + c_y E_{\infty}^y]$$

$$E(x,y,z) = -\frac{ikf\epsilon e^{-ikf}}{2\pi} \int_{k_z^2 + k_y^2 \leq k_{max}^2} [c_x E_{\infty}^x + c_y E_{\infty}^y] \frac{1}{k_z} \times \int_{k_z^2 + k_y^2 \leq k_{max}^2} e^{i[k_xz + k_yy + k_z]} dk_x dk_y$$
where \( z = 0 \) represents the focal plane. In general \( c_x \) and \( c_y \) are complex 2d functions that depend on \((k_x, k_y)\) and define the polarization state of the light at the back aperture of the lens. In the cases of linear or elliptical polarization states, \( c_x \) and \( c_y \) are constants over the aperture domain. The integration domain is restricted to spatial frequencies that satisfy \( k_x^2 + k_y^2 \leq k_{\text{max}}^2 \), where in terms of the numerical aperture \((NA = n \sin \theta_{\text{max}})\) as \( k_{\text{max}} = N A k_0 \). The same limit of \( k_{\text{max}} \) can also be expressed in terms of the aperture radius \( R \) using equation 4: \( R = k_{\text{max}} f / k = N A f / n \) (Fig. 1).

In order to extend the domain of the integrals in eq. (7) to infinity, an aperture function is needed that vanishes for frequencies larger than the magnitude of \( k \):

\[
\Theta(k_x, k_y) = \begin{cases} 
1 & \text{if } k_x^2 + k_y^2 \leq k_{\text{max}}^2 \\
0 & \text{if } k_x^2 + k_y^2 > k_{\text{max}}^2 
\end{cases}
\]  

\[
E(x, y, z) = -\frac{i e^{-ikf}}{2\pi} \int\int \Theta(k_x, k_y) [c_x E_{\text{inc}}^x + c_y E_{\text{inc}}^y] \frac{1}{k_z} \times e^{i[k_x x + k_y y + k_z z]} dk_x dk_y 
\]

The previous expression can be rewritten as:

\[
E(x, y, z) = \frac{i e^{-ikf}}{2\pi} \text{IFT} \left[ \Theta(k_x, k_y) [c_x E_{\text{inc}}^x + c_y E_{\text{inc}}^y] \frac{1}{k_z} e^{i[k_x x + k_y y]} \right]
\]

which can be calculated with the Fast Fourier Transform algorithm (fft) as shown in [2].

**OPTICAL SYSTEM**

The simplified system that we consider is very similar to most beam shaping experimental setups (i.e. holographic optical tweezers) that use beam shaping elements like SLMs or DMDs that can modulate phase and/or amplitude. A vectorial beam input field with an arbitrary polarization state in the transverse direction \((xy)\) can be prepared splitting the components, modulating them independently and then recombining using an interferometer setup where both components have the same optical path length (this can be done with one or two beam shaping elements). Then, a complex polarization mask can be added to set the polarization, this can be done at the beam shaping element or with polarization elements like a quarter wave plate or a q plate. In general, the vector beam is focused by a lens that is placed at a distance of the focal length from the beam shaping element (located in the Fourier plane of the lens), then a second lens collimates the beam, projecting the surface of the beam shaping element to the back aperture of the microscope objective where the field \( E_{\text{inc}} \) is described. This sets the spatial dimension of the incident field, so it also can be written as a function of \((x_\infty, y_\infty)\) (which is a resized projection of the beam shaping element), in addition to the descriptions in terms of the dimensionless \((k_x/k, k_y/k)\) or \((x_\infty/f, y_\infty/f)\). This means that at each point \((x_\infty, y_\infty)\), we know the amplitude, phase and polarization.

In the computer code, we follow the steps of the experiment: The input beam is decomposed into the two complex transverse components \( E_{\text{inc},x} e^{i\phi_x} \) and \( E_{\text{inc},y} e^{i\phi_y} \) and we define them as a function of the spatial coordinates. Then we add a 2d complex polarization mask \( c_x, c_y \) to each component to project the beam into arbitrary polarization states. Notice that \( c_x, c_y \) can also be included in the definition of \( E_{\text{inc}} \) but we decided to separate them to mimic the experiment. Finally there is the aperture \( \Theta \) that can also be controlled by the beam shaping elements and limited by the lens. Those 7 transverse 2d masks \( (E_{\text{inc},x}, E_{\text{inc},y}, \phi_{\text{inc},x}, \phi_{\text{inc},y}, c_x, c_y, \Theta) \) are drawn in Fig. 2 and represent \( E_{\text{inc}} \) at the back aperture of the microscope objective.

In most experiments, one polarization component is modulated by a spatial light modulator than can control phase and amplitude, that modulated beam then propagates through a polarization element that controls \( c_x \) and \( c_y \) like a half wave plate (HWP) to control the a linear polarization state, or a quarter wave plate (QWP) for elliptical polarization. Radial and azimuthal polarization can be obtained with a vortex phase plate. Arbitrary beams can be implemented by modulating independently both transverse polarizations. Fig. 2 shows the case for \( E_{x0} = E_{y0} \) and \( \phi_{x0} = \phi_{y0} \) and a spatially dependent polarization.
NUMERICAL IMPLEMENTATION

To calculate the tightly focused field we consider 3 coordinate systems: the spatial Cartesian coordinates before the lens \((x_\infty, y_\infty)\) that describe the size of the aperture and the beam, the angular system at the same plane \((k_x, k_y)\) and the Cartesian system at the focal plane \((x, y)\). All coordinate systems are implemented in a grid of \(L \times L\) points, where \(L\) is an even number (for the FFT algorithm, usually \(L = 2^{11} - 2^{12}\)) to reduce errors with the numerical implementation of the Fourier Transform FFT.

In the following subsections we define the most important parameters of the focusing system like the numerical aperture and the effective focal length of the lens. Then grid domain, spatial coordinates at the back aperture \(x_\infty, y_\infty\), followed by the aperture function, the angular spectrum and finally the integration through a fft and the spatial grid at the focus.

**Focusing lens**

The focusing lens is usually a high numerical aperture microscope objective (oil immersion) with an effective focal length of \(f\) and a numerical aperture \(NA\), where \(NA = n_2 R/f\) with \(R\) the radius of the back aperture and \(n_2\) the refractive index of the media in front of the lens \((n_2 = 1.518\) for immersion oil, \(n_1 = 1\) for air). The effective numerical aperture \(NA\) (can be adjusted with the beam shaping elements) or with a mechanical iris. We consider that the back aperture is centered at the simulation domain and that it has a diameter of \(N\) points (with \(N\) an even number). Also, the area surrounding the aperture (with diameter \(D\)) has to be padded with zeros. A complete discussion of the expected errors as a function of the number of points for \(L\) and \(N\) respectively is in [2].

**Mesh**

As mentioned in the previous subsection, the grid has a size of \(L \times L\) points or pixels. The 2 dimensional Fourier Transform is defined in that grid. In Matlab, the zero frequency is at the pixel coordinate \((L/2 + 1, L/2 + 1)\). When defining the spatial and angular coordinate system at the back aperture (same \(L \times L\) grid), we set the origin of the coordinates at \((L/2 + m_0, L/2 + m_0)\) with \(m_0 = 0.5\) to eliminate the zero frequency contributions that diverge in equations 5 and 6 (terms \(1/(k_x^2 + k_y^2)\)). Hence, the spatial and angular domains are \((- (L - 1)/2, (L + 1)/2) \Delta x\) and \((- (L - 1)/2, (L + 1)/2) \Delta k\) with \(\Delta x\) and \(\Delta k\) the spatial and angular step sizes respectively and are defined in the following subsections. As a result, the Fourier transform is equal to the centered Fourier transform times a phase term \(\phi_{\text{shift}}(X, Y) = -i2\pi \frac{m_0}{L} X + i2\pi \frac{m_0}{L} Y\). Where \(X\) and \(Y\) are the dimensionless Cartesian coordinates centered at \((L/2 + 1, L/2 + 1)\). In order to correctly compute the complex electric field we have to multiply the resulting field components by the term \(-\phi_{\text{shift}}(X, Y)\).

Figure 3 shows a comparison between the uncorrected (top row) and the corrected phase (bottom row) in the dominant polarization component of a highly focused Gaussian beam that is linearly polarized before focusing. That phase is well known and it consists of concentric constant sections that have a phase difference of \(\pi\). We observe that the sections of the uncorrected phase are not constant, but have a small gradient which is more noticeable in the cross section plot (right column).

**Spatial domain \((x_\infty, x_\infty)\)**

At the back aperture of the microscope objective the spatial domain \(x_\infty\) and \(x_\infty\) is defined multiplying the discrete point grid by the constant \(\Delta x_\infty\) which converts it into a spatial grid (dimensions of micrometers in the code). \(\Delta x_\infty\) is defined by the size of the aperture which is defined with a radius of \(N\) \((N = 2^7)\) points which is equivalent to the physical size of the aperture radius \(R\) (in microns). In this way \(\Delta x_\infty = R/N\). The spatial grids are \((- (L - 1)/2, (L - 1)/2) \Delta x_\infty\) \((m_0 = 0.5)\).
Angular spectrum \((k_x, k_y)\)

The angular spectrum coordinates are defined as \(k_x = k_1 x_\infty / f\) and \(k_y = k_1 y_\infty / f\), where \(k_1 = n_3 k_0\) and \(k_0 = 2\pi/\lambda\). The size of each grid point in the angular spectrum mesh is \(\Delta k = k_0 N.A / N\). Notice that the interval does not include zero, this is done to avoid diverging terms \(1/(k_x^2 + k_y^2)\).

After applying the `fft2`, the size of the spatial output mesh is defined by \(\Delta x_f = 2\pi / (L \Delta k) = NA / (L(N.A))\) \((\Delta x_f = \Delta y_f)\).

**Input Field** \(E_{inc}\) and polarization mask

The input field with the polarization state is described by the seven 2-dimensional masks in Fig. 2: amplitudes \(E_{x0}, E_{y0}\), phases \(\phi_{x0}, \phi_{y0}\), aperture \(\Theta\) and polarization \(c_x, c_y\); all defined on the \(L \times L\) domain. The polarization mask at a given point can be represented by \(ae^{i\phi}\), with \(abs(a) = 1\). As we mentioned before, in the experiments sometimes it is more convenient to use the spatial coordinates \((x_\infty, y_\infty)\) to define the input field, because the size of the field is defined at the beam shaping which is projected onto the back aperture of the microscope objective. Also the spatial dimensions of the input field can be measured with a CCD camera.

Hence, in the examples of section 5, we choose to describe the input as a function of the spatial coordinates.

**Aperture (circular, square, annular)**

The step function in eq. 8 depends on the angular spectrum. However, we can also express it in terms of its spatial size \(R\) which is convenient because is the way it is measured. When implementing an FFT it is very important to consider that a sudden discontinuity will result in the appearance of oscillations or speckle (aliasing). Those effects can be minimized by changing the definition of the aperture function. Here we present the result of [2] it in terms of \((x_\infty, y_\infty)\), in the case of a circular aperture:

\[
\Theta'(x_\infty, y_\infty) = \frac{1}{2} \left(1 + \text{tanh} \left(\frac{3}{4 \Delta X_\infty} \left(R - (x_\infty^2 + y_\infty^2)^{1/2}\right)\right)\right)
\]  
\((11)\)

where \(R = N \Delta X_\infty = f(N.A)/n_2\).

Other common geometries for the aperture are a square or a ring which we can describe in the other angular system to show the equivalence. In the case of a square aperture, the continuous version of the step function \(\Theta(k_x, k_y)\)

**FIG. 4.** Focusing a linearly polarized Gaussian beam. The parameters are: \(L = 2^{12}, N = 2^7, n_2 = 1.518, NA = 1.3, f = 2\) mm.

is:

\[
\Theta(k_x, k_y) = \frac{1}{4} \left(1 + \text{tanh} \left(\frac{3}{4 \Delta k} \left(k_{max} - k_x\right)\right)\right) \times \left(1 + \text{tanh} \left(\frac{3}{4 \Delta k} \left(k_{max} - k_y\right)\right)\right)
\]
\((12)\)

where the \(\sqrt{2}\) factor dividing \(k_{max}\) appears because the maximum diameter for a square aperture is the diagonal.

The modified \(\Theta(k_x, k_y)\) for an annular aperture is:

\[
\Theta(k_x, k_y) = \frac{1}{2} \text{tanh} \left(\frac{3}{4 \Delta k} \left(k_{max1} - (k_x^2 + k_y^2)^{1/2}\right)\right) - \frac{1}{2} \text{tanh} \left(\frac{3}{4 \Delta k} \left(k_{max2} - (k_x^2 + k_y^2)^{1/2}\right)\right)
\]
\((13)\)

where \(k_{max} \geq k_{max1} > k_{max2}\) and \(\Delta k_l = k_{max1} - k_{max2}\) is the width of the annular aperture.

**Integration**

Once the mesh, coordinates and input field set, then the argument of eq. (10) is written and the Fast Fourier Transform is calculated.

**EXAMPLES**

In the following subsections we show specific examples with different polarization states and input fields. The incident fields are described in the spatial or the angular coordinates. The magnitude of the errors and the dependency on the domain size are discussed in [2].
$\sigma = 1$, and vortex phase of order $m = -1$. The parameters are: $L = 2^{12}$, $N = 2^{7}$, $n_2 = 1.518$,NA = 1.3, $f = 2$ mm.

**Linearly polarized Gaussian**

We consider a Gaussian beam defined as:

$$E_{inc}(x_\infty, y_\infty) = E_0 e^{\frac{-(x_\infty^2+y_\infty^2)}{w_0^2}} e^{i\phi(x_\infty, y_\infty)}$$  \hspace{1cm} (14)$$

where $E_0$ (set to 1) is the field amplitude before the lens, $w_0$ is the waist of the Gaussian, $R = D/2$ is the aperture radius and $f_0$ the filling factor for a Gaussian beam. The filling factor is defined as $f_0 = \omega_0/R$, with $\omega_0$ the Gaussian waist. The same field in angular coordinates:

$$E_{inc}(k_x, k_y) = E_0 e^{\frac{-(k_x^2+k_y^2)}{w_0^2}} e^{i\phi(k_x, k_y)}$$  \hspace{1cm} (15)$$

where the new waist is $w'_0 = w_0 k_1/f$.

Figure 4 shows the results for a Gaussian beam initially polarized in the direction $n_x$, focused by a lens with $NA = 1.3$, $n = 1.518$, $f = 2$ mm, $f_0 = 1.0$ and $R = 1.71$ mm.

In the case of a collimated Gaussian beam, it is a good approximation to consider the phase as constant, so it is set to $\phi(x_\infty, y_\infty) = \phi(k_x, k_y)$.

Then, a linear polarization state is selected with the $c_x$ and $c_y$ coefficients. In this case we select only the $x$ component defining $c_x = 1$ and $c_y = 0$ in the $L \times L$ domain.

The top row in Fig. 4 has the 2 dimensional amplitude, phase, aperture, x polarization and y polarization masks. The 6 frames in the bottom depict the result at the focal plane with the 3 intensities and phases.

**Circular polarization**

Same Gaussian with an optical vortex phase with $\phi_\infty = \phi_\infty m$ $m = -1$ through a quarter wave plate, where $\phi_\infty$ is the azimuthal angle in the $x_\infty, y_\infty$ plane. The polarization state, which is defined as $c_x = 1$, $c_y = i$ (left circular polarization with $\sigma = 1$) in the $L \times L$ domain. The input planes in the domain $x_\infty, y_\infty$ (same plane as $k_x, k, y$) are at the top of Fig. 5, while the resulting field at the focus with the 3 intensities and phases is in the lower part. We observe that the in the $z$ component there is coupling between the orbital $m$ and the polarization $\sigma$ resulting in a null charge ($m = 0$).

**Radial Polarization**

Figure 6 shows the polarization masks for the transverse polarization components with $c_x = \cos \phi = k_x/\sqrt{k_x^2 + k_y^2}$ and $c_y = \sin \phi = k_y/\sqrt{k_x^2 + k_y^2}$. We introduced auxiliary polar coordinates $\phi$ and $\rho$ in the $x_\infty, y_\infty$ domain. The angle $\phi$ can also be defined as the inverse tangent of $k_y/k_x$. The calculation considers an annular mask and the same Gaussian input.

**Azimuthal Polarization**

Figure 7 shows the polarization masks for the transverse polarization components $c_x = \cos \phi = -k_y/\sqrt{k_x^2 + k_y^2}$ and $c_y = \sin \phi = k_x/\sqrt{k_x^2 + k_y^2}$. This case is interesting because there is no axial component as expected for this case, the ratio between the amplitude of the $E_z$ and the $E_x$ has an order of $10^{-19}$. 

![Image](Image 69x548 to 285x740)

**FIG. 5.** Focusing a Gaussian beam right circular polarization, 

![Image](Image 338x559 to 541x740)

**FIG. 6.** Focusing a radial polarized annular aperture (TM 0th). The parameters are: $L = 2^{12}$, $N = 2^{7}$, $n_2 = 1.518$, NA = 1.3, $f = 2$ mm.

**FIG. 7.** Focusing a TM aperture with linear (left circular) and azimuthal polarization.
Arbitrary Polarization

We also consider a Gaussian beam with flower (spider web) polarization state as described in \cite{7} defined by the parameter $s = 8$ ($-8$): The polarization is $c_x = \cos((s/2)\phi))$ and $c_y = \sin((s/2)\phi))$ with $s = 8$, (flower). This is shown in Fig. 8. Inverting the sign in $s$ yields a similar pattern (spiderweb).

Arbitrary beam

All the previous examples had considered $E_{0x} = E_{0y}$ and $\phi_{0x} = \phi_{0y}$. Here we consider a beam that has different amplitudes, phases, and polarization profiles in the transverse components. In Fig. 9 we show a beam with parameters in the angular coordinates:

$$E_{incx}(k_x, k_y) = E_{0x}\sqrt{(k_x - k_{max}/2)^2 + k_y^2 e^{-\frac{(k_x^2 + k_y^2)}{k_{max}^2}}}$$

$$E_{incy}(k_x, k_y) = E_{0y}\sqrt{(k_x + k_{max}/2)^2 + k_y^2 e^{-\frac{(k_x^2 + k_y^2)}{k_{max}^2}}}$$

$$\phi_{incx} = \tan^{-1}\frac{k_y}{(k_x - k_{max}/2)}$$

$$\phi_{incy} = \tan^{-1}\frac{k_y}{(k_x + k_{max}/2)}$$

$$c_{incx} = k_x/\sqrt{k_x^2 + k_y^2}$$

$$c_{incy} = k_y/\sqrt{k_x^2 + k_y^2}$$

(16)

Where $E_{0x}$ ($E_{0y}$) is such as the maximum value of $E_{incx}$ ($E_{incy}$) is one.

FIG. 7. Focusing a azimuthal polarized annular aperture (TE 0th). The parameters are: $L = 2^{12}$, $N = 2^7$, $n_2 = 1.518$, NA= 1.3, $f = 2$ mm.

FIG. 8. Focusing a Gaussian beam with flower (order 8) polarization: The parameters are $L = 2^{12}$, $N = 2^7$, $n_2 = 1.518$, NA= 1.3, $f = 2$ mm.

FIG. 9. Focusing of a beam with different amplitudes, phases, and polarization profiles in the transverse components. The first group of images shows the amplitude, phase, polarization maps and the circular aperture. The second group shows the focal fields. The parameters are $L = 2^{12}$, $N = 2^7$, $n_2 = 1.518$, NA= 1.3, $f = 2$ mm.

PLANAR INTERFACE (PROPAGATING AND EVANESCENT)

Many applications consider these beams propagating through a planar interface like optical micromanipulation where the beam propagates through a glass-liquid interface. Due to the refractive index mismatch across the planar boundary there is the possibility of total internal reflection and having evanescent waves. In this section
we reproduce the results contained in Novotny and Hetch book [9]. They consider that the field focuses at \( z = 0 \) and the boundary (xy plane) is at a height of \( z_0 \). The refractive index at \( z < z_0 \) is \( n_1 \) and \( k_{x1} = \sqrt{k_1^2 - (k_x^2 + k_y^2)} \) \( (k_1 = nk_0) \), above the boundary for \( z > z_0 \), the refractive index is \( n_2 \) and \( k_{x2} = \sqrt{k_2^2 - (k_x^2 + k_y^2)} \) \( (k_2 = n_2k_0) \).

\textbf{Fresnel Coefficients}

The reflection and transmission Fresnel coefficients for a planar boundary (s and p components) are:

\[
\begin{align*}
 r^s(k_x, k_y) &= \frac{\mu_2 k_{z1} - \mu_1 k_{z2}}{\mu_2 k_{z1} + \mu_1 k_{z2}} \\
 r^p(k_x, k_y) &= \frac{\mu_2 k_{z1} - \epsilon_1 k_{z2}}{\epsilon_2 k_{z1} + \epsilon_1 k_{z2}} \\
 t^s(k_x, k_y) &= \frac{2\mu_2 k_{z1}}{\mu_2 k_{z1} + \mu_1 k_{z2}} \\
 t^p(k_x, k_y) &= \frac{2\epsilon_2 k_{z1}}{\epsilon_2 k_{z1} + \epsilon_1 k_{z2}} \sqrt{\frac{\mu_2 
_1}{\mu_2 
_2}} \quad (17)
\end{align*}
\]

Normally, for a dielectric we can set \( \mu_1 = \mu_2 = 1 \).

The field between the lens and the end of the boundary (same material) is a superposition of the focused and reflected fields \( E = E_f(x, y, z) + E_r(x, y, z) \) for \( z < z_0 \):

\[
E_f(x, y, z) = -\frac{ie^{-ikf}}{2\pi} \int_{-\infty}^{\infty} \left[c_x E_{\infty,f}^x + c_y E_{\infty,f}^y \right] \times \frac{1}{k_z} e^{ik_x x + ik_y y + ik_{z1} z} dk_x dk_y \quad (18)
\]

\[
E_r(x, y, z) = -\frac{ie^{-ikf}}{2\pi} \int_{-\infty}^{\infty} \left[c_x E_{\infty,r}^x + c_y E_{\infty,r}^y \right] \times \frac{1}{k_z} e^{ik_x x + ik_y y - ik_{z1} z} dk_x dk_y \quad (19)
\]

In the case of \( z > z_0 \) then \( E = E_t \) where the transmitted field:

\[
E_t(x, y, z) = -\frac{ie^{-ikf}}{2\pi} \int_{-\infty}^{\infty} \left[c_x E_{\infty,t}^x + c_y E_{\infty,t}^y \right] \times \frac{1}{k_z} e^{ik_x x + ik_y y + ik_{z2} z} dk_x dk_y \quad (20)
\]

The reflected \( E_{\infty,r} \) and transmitted \( E_{\infty,t} \) in terms of the \( xy \) components of \( E_{\infty} \):

\[
E_{\infty,r}^x = \frac{k_{z1}}{k_1} E_{\infty}^x(k_x/k, k_y/k) \left( e^{ik_x x_{z0}} \right) \times \\
\left( \begin{array}{c}
 r^s k_{x1}^2 - r^p k_x k_{z1}/k_1 \\
 r^s k_y k_{x1}/k_1 \\
 r^p (k_x^2 + k_y^2) k_{x1}/k_1
\end{array} \right) \quad (21)
\]

For the other transverse polarization component (direction \( n_y \)):

\[
E_{\infty,r}^y = \frac{k_{z1}}{k_1} E_{\infty}^y(k_x/k, k_y/k) \left( e^{ik_x x_{z0}} \right) \times \\
\left( \begin{array}{c}
 r^s k_{y1}^2 - r^p k_y k_{z1}/k_1 \\
 r^s k_x k_{y1}/k_1 \\
 -r^p (k_x^2 + k_y^2) k_{y1}/k_1
\end{array} \right) \quad (22)
\]

The transmitted component:

\[
E_{\infty,t}^x = \frac{k_{z1}}{k_1} E_{\infty}^x(k_x/k, k_y/k) \left( e^{ik_x x_{z0}} \right) \times \\
\left( \begin{array}{c}
 -r^s k_{x1}^2 + t^s k_x k_{x2}/k_2 \\
 -r^s k_{x1} k_{y1}/k_2 \\
 t^s (k_x^2 + k_y^2) k_{x2}/k_2 -r^p (k_x^2 + k_y^2) k_{x2}/k_2
\end{array} \right) \quad (23)
\]

For the other transverse polarization component (direction \( n_y \)):

\[
E_{\infty,t}^y = \frac{k_{z1}}{k_1} E_{\infty}^y(k_x/k, k_y/k) \left( e^{ik_x x_{z0}} \right) \times \\
\left( \begin{array}{c}
 -r^s k_{y1}^2 + t^s k_y k_{y2}/k_2 \\
 -r^s k_{y1} k_{x1}/k_2 \\
 t^s (k_x^2 + k_y^2) k_{y2}/k_2 -r^p (k_x^2 + k_y^2) k_{y2}/k_2
\end{array} \right) \quad (24)
\]

Figure 9 shows the case of the linearly polarized Gaussian with \( w_0 = R \) \( (f_0 = 1) \) focused by a lens with \( NA = 1.4 \) at \( z = 0 \), the position of the interface is at \( z_0 = -\lambda, 0, \lambda \) (columns 1, 2 and 3 respectively). The first row has cross sections of the plane \( x, w \) with \( y = 0 \) and the second row plane \( y, z \) with \( x = 0 \). The color maps represent the fourth root of total intensity in order to show more clearly the details. The case with \( z_0 = 0 \) appears in [9].

\textbf{CONCLUSION}

We presented two simple programs to calculate tightly focused vectorial field: one for a propagating field and the other considers a planar interface where evanescent fields can emerge. These programs are based on previous results described in [2] [9]. The main contribution is the discussion about the correction that has to be made for a shifted Fourier transform, which is a subtle detail that is easy to miss. Also, the use of Cartesian coordinates helps when making comparisons with experiments where the light is shaped by rectangular arrays which are described in [9].

\textbf{Acknowledgements}

Work partially funded by DGAPA UNAM PAPIIT grants IN107719 and IN107222; CITIC-LANMAC and
FIG. 10. Intensity $E \cdot E^*$ of a focused linearly polarized Gaussian beam with parameters $f_0 = 1$, $NA = 1.4$, close to an interface glass-air. The spanned area $5\lambda \times 8\lambda$ is the same for all the images, the $(0, 0)$ is the focus of the lens. In the first row is shown the plane $(x, z)$ and in the second row is shown the plane $(y, z)$, where $z$. (a) The interface is located at $z_0 = -\lambda$. (b) The interface is located at the focal plane $z_0 = 0$. (c) The interface is located at $z_0 = \lambda$ and. We computed the field at 200 planes along the $z$ direction.

In the three cases $L = 2048$ and $N = 150$.

CONACYT LN-299057.

* Current address: Fresnel Institute, isael.herrera@fresnel.fr
† pedro.quinto@nucleares.unam.mx

[1] B. Richards and E. Wolf, Electromagnetic diffraction in optical systems. II. Structure of the image field in an aplanatic system. Proc. R. Soc. Lond.- A. 253, 358–379 (1959).
[2] M. Leutenegger, R. Rao, R. A. Leitgeb, and T. Lasser. Fast focus field calculations. Opt. Express 14, 11277-11291 (2006).
[3] Q. Li, M. Chambonneau, M. Blothe, H. Gross, and S. Nolte, Flexible, fast, and benchmarked vectorial model for focused laser beams. Appl. Opt. 60, 3954-3963 (2021).
[4] F. Capriile, L. A. Masullo, F. D. Stefani, PyFocus - A Python package for vectorial calculations of focused optical fields under realistic conditions. Application to toroidal foci. Comput. Phys. Commun. 275, 108315 (2022).
[5] Boruah, B. R. and Neil, M. A. A. Focal field computation of an arbitrarily polarized beam using fast Fourier transforms. Opt. Commun. 282, 4660-4667 (2009).
[6] L. Novotny, and Hecht. Principles of Nano-Optics. (Cambridge Univ. Press, 2012).
[7] E. Otte, K. Tekce, and C. Denz. Tailored intensity landscapes by tight focusing of singular vector beams, Opt. Express 25, 20194-20201 (2017).
clear all; %close all; clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% run time sim 10s with L=2^12, and N=2^7
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%initial parameters
L=2^12; % main domain size
N=2^7; % number of points for the aperture radius
z=0; %observation plane (0 at the focal plane)
m0=1/2; %coordinate shift
xmax=0.9; %half the size of the cropped output in wavelengths
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% Spatial coordinate system before lens (x_inf, y_inf)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d_inf=linspace(-L/2+m0,L/2−m0,L)*dx_inf; %spatial axis shifted by m0.
[x_inf,y_inf]=meshgrid(d_inf,−d_inf); %mesh, y inverted in Matlab images
[theta_inf,rho_inf] = cart2pol(x_inf,y_inf); %auxiliary polar coordinates
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%% angular
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k0=2*pi/lambda;
k1=n2*k0;
dk=k0*NA/N; %dk=(1/2) *(k1/(f))*dx0;
kx=x_inf*k1/f;
ky=y_inf*k1/f;
dxf=2*pi/(L*dk); %equivalent to dxf=N*lambda/(L*NA);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%correction for m0 shift
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x=linspace(-L/2,L/2,L); %%%auxiliary coordinate for correction
y=x; [X,Y]=meshgrid(x,y);
PhaseShift=m0*2*pi*X/L+m0*2*pi*Y/L; %correction phase
%Center of Fourier transform displaced (vert and hor) by m0.
%shift of \(\Delta k_x, \Delta k_y\) = m0*(2*pi/L).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%input field initial field transverse
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
w0=f*NA/n2; % The beam waist is equal to the aperture radius.
E_incx=exp(−(x_inf.ˆ2+y_inf.ˆ2)/(w0ˆ2)); %normalized amplitude E_x0 A_0
phi_incx=double(zeros(L,L)); %%%constant
E_incx=double(zeros(L,L)); %%%constant
E_incy=E_incx;
phi_incy=phi_incx;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%round soft aperture (spatial coordinates)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Theta=0.5*(1+tanh((1.5/1)*(N−sqrt((x_inf/dx_inf).ˆ2+(y_inf/dx_inf).ˆ2))));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%polarization state
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cx=ones(L,L); cy=zeros(L,L); % lin horizontal
% %%polared polar coordinates
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% lin vertical
%cx=zeros(L,L); cy=ones(L,L);
% lin diag
%cx=1*ones(L,L); cy=1*ones(L,L);
% right circ
%cx=ones(L,L); cy=1i*ones(L,L);
% left circ
%cx=1i*ones(L,L); cy=ones(L,L);
% radial
%phi_aux=mod(atan2(ky,kx),2*pi); s=8; cx=cos((s/2)*phi_aux); cy=sin((s/2)*phi_aux);
% azimuthal
%phi

%%input x
E_{inxx}=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_{incx}.*exp(1i*phi_{incx}).*(ky.*2+kx.*kz1./kM1)./(kx.*2+ky.*2);

%%y component
E_{inxy}=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_{incy}.*exp(1i*phi_{incy}).*(-ky.*kx+kx.*ky.*kz1./kM1)./(kx.*2+ky.*2);

%%z component
E_{infxz}=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_{incz}.*exp(1i*phi_{incz}).*(-(kx.*2+ky.*2).*ky./kM1)./(kx.*2+ky.*2);

%%y input
E_{infyx}=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_{incy}.*exp(1i*phi_{incy}).*(-ky.*kx+kx.*ky.*kz1./kM1)./(kx.*2+ky.*2);

%%z component
E_{infyx}=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_{incy}.*exp(1i*phi_{incy}).*(-ky.*kx+kx.*ky.*kz1./kM1)./(kx.*2+ky.*2);

%%integration
Ex0=ifftshift(ifft2((fftshift(Ex)))).*exp(1i*PhaseShift);
Ex=Ex0((L/2+1−xmax2):(L/2+1+xmax2),(L/2+1−xmax2):(L/2+1+xmax2)); %cropped
Ey0=ifftshift(ifft2((fftshift(Ey)))).*exp(1i*PhaseShift);
Ey=Ey0((L/2+1−xmax2):(L/2+1+xmax2),(L/2+1−xmax2):(L/2+1+xmax2)); %cropped
Ez0=ifftshift(ifft2((fftshift(Ez)))).*exp(1i*PhaseShift);
Ez=Ez0((L/2+1−xmax2):(L/2+1+xmax2),(L/2+1−xmax2):(L/2+1+xmax2)); %cropped

% intensity ratios
C0y=max(max(abs(Ey).ˆ2))/max(max(abs(Ex).ˆ2));
C0z=max(max(abs(Ez).ˆ2))/max(max(abs(Ex).ˆ2));

% plots
a='jet';
b='gray';
figure(1) %amplitude Ex
Exn=abs(Ex)./max(max(abs(Ex)));
% APPENDIX B: PROGRAM FOR INTERFACE

clear all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
initial parameters
L=2^10; % main domain size
N=2^6; % number of points for the aperture radius
z0=0; %interface position (focus at z=0)
z=linspace(-2,2,41); %axial domain
m0=1/2; %coordinate shift
xmax=4; %half the size of the cropped output in wavelengths
%(final size of images is 2xmax)

n1=1.518; %refractive index for incident field (immersion oil)
n2=1.518; % refractive index for reflected field (immersion oil)
n3=1; % refractive index for transmitted field (after interface)
e1=n2^2;
e2=n3^2;
m1=1; %permeability
m2=1; %permeability
f=1800; % effective focal length of the lens in um (10^-6 m)
NA=1.3; %effective numerical aperture
lambda=1.064; %wavelength in um
l=1.064*10^(-6); %wavelength um
R=f*NA/n2; %aperture radius in um;
%conversion factor at output plane in um/px

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Spatial coordinate system before lens (x_inf, y_inf)
d_inf=linspace(-L/2+m0,L/2-m0,L)*dx_inf; %spatial axis shifted by m0.
[kx_inf,y_inf]=meshgrid(d_inf,−d_inf); %mesh, y inverted in Matlab images
[theta_inf,rho_inf] = cart2pol(x_inf,y_inf); %auxiliary polar coordinates
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
angular
k0=2*pi/lambda;
k1=n2*k0;
k2=n3*k0;
dk=k0*NA/n2; %dk=(1/2) *(k1/(f))*dx0;
kx=x_inf*k1/f;
yk=y_inf*k1/f;
dxinf=2*pi/(L*dk); %equivalent to dxinf=N*lambda/(L*NA);
%conversion factor at output plane in um/px
km1=k1*ones(L,L); %%%array with magnitude of k1
kxi=2*pi/(L*dk);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
correction for m0 shift
x=linspace(-L/2,L/2,L); %%%auxiliary coordinate for correction
y=x; [X,Y]=meshgrid(x,y);
PhaseShift=m0*2*pi*X/L+m0*2*pi*Y/L; %correction phase
%Center of Fourier transform displaced (vert and hor) by m0.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
input field initial field transverse
w0=f*NA/n2; % The beam waist is equal to the aperture radius.
E_incx=exp(−(x_inf.^2+y_inf.^2)/(w0^2)); %normalized amplitude E_x0 A_0
phi_incx=double(zeros(L,L)); %%%phase phi_x0 fase init; phi_0
E_incx=E_incx;

phase_inc = phi_incx;
round soft aperture (spatial coordinates)
\[ \Theta = 0.5 \left( 1 + \tanh \left( \frac{1.5}{1} \left( N - \sqrt{(x_{\text{inf}}/dx_{\text{inf}})^2 + (y_{\text{inf}}/dy_{\text{inf}})^2} \right) \right) \right) \]

%angular space
\[ k_{\text{max}} = R \cdot k_1 / f \]
\[ \Theta = 0.5 \left( 1 + \tanh \left( \frac{1.5}{dk} \left( k_{\text{max}} - \sqrt{(k_x)^2 + (k_y)^2} \right) \right) \right) \]

%%%polarization
\( c_x = \text{ones}(L,L); \ c_y = \text{zeros}(L,L); \) % lin horizontal
\( c_x = \text{zeros}(L,L); \ c_y = \text{ones}(L,L); \) % lin vertical
\( c_x = \text{ones}(L,L); \ c_y = \text{ones}(L,L); \) % linear diagonal
\( c_x = \text{ones}(L,L); \ c_y = \text{i} \times \text{ones}(L,L); \) % right circular
\( c_x = \text{i} \times \text{ones}(L,L); \ c_y = \text{ones}(L,L); \) % left circular
\( c_x = \text{cos}(\phi) / \sqrt{(k_x)^2 + (k_y)^2}; \ c_y = \text{sin}(\phi) / \sqrt{(k_x)^2 + (k_y)^2}; \) % radial
\( c_x = \text{cos}(\phi) / \sqrt{(k_x)^2 + (k_y)^2}; \ c_y = \text{sin}(\phi) / \sqrt{(k_x)^2 + (k_y)^2}; \) % azimuthal
\( c_x = \text{cos}(\phi) / \sqrt{(k_x)^2 + (k_y)^2}; \ c_y = \text{sin}(\phi) / \sqrt{(k_x)^2 + (k_y)^2}; \) % flower
\( c_x = \text{cos}(\phi) / \sqrt{(k_x)^2 + (k_y)^2}; \ c_y = \text{sin}(\phi) / \sqrt{(k_x)^2 + (k_y)^2}; \) % spider

Fresnel coefficients
\( r_s = \frac{(m_2 \cdot k_z_1 - m_1 \cdot k_z_2)}{(m_2 \cdot k_z_1 + m_1 \cdot k_z_2)} \)
\( r_p = \frac{(e_2 \cdot k_z_1 - e_1 \cdot k_z_2)}{(e_2 \cdot k_z_1 + e_1 \cdot k_z_2)} \)
\( t_s = \frac{2 \cdot m_2 \cdot k_z_1}{m_2 \cdot k_z_1 + m_1 \cdot k_z_2} \)
\( t_p = \sqrt{\frac{m_2 \cdot e_1}{m_1 \cdot e_2}} \cdot \frac{2 \cdot e_2 \cdot k_z_1}{e_2 \cdot k_z_1 + e_1 \cdot k_z_2} \)

%Factors for E\_inf\_r
\( C_{\text{F1}}(r_s \cdot k_x^2 - r_p \cdot k_y^2)/(r_s \cdot k_x^2 + r_p \cdot k_y^2) \)
\( C_{\text{F2}}(r_s \cdot k_x^2 - r_p \cdot k_y^2)/(r_s \cdot k_x^2 + r_p \cdot k_y^2) \)

%Focused propagating before interface
\( C_{\text{F1}}(r_s \cdot k_x^2 - r_p \cdot k_y^2)/(r_s \cdot k_x^2 + r_p \cdot k_y^2) \)
\( C_{\text{F2}}(r_s \cdot k_x^2 - r_p \cdot k_y^2)/(r_s \cdot k_x^2 + r_p \cdot k_y^2) \)

%Transmitted E\_inf\_t
\( C_{\text{F1}}(r_s \cdot k_x^2 - r_p \cdot k_y^2)/(r_s \cdot k_x^2 + r_p \cdot k_y^2) \)
\( C_{\text{F2}}(r_s \cdot k_x^2 - r_p \cdot k_y^2)/(r_s \cdot k_x^2 + r_p \cdot k_y^2) \)
MTz2t = -tp.*(kx.ˆ2+ky.ˆ2).*ky./kM2).*CF2; % y input

for j=1:length(z);
    Hzf=exp(1i*kz1.*z(j)).*Theta; %propagating
    Hzr=exp(-1i*kz1.*z(j)+1i*2*kz1*z0).*Theta; %reflected
    Hzt=exp(1i*kz2.*z(j)+1i*(kz1-kz2)*z0).*Theta; %transmitted

    if z(j) ≤ z0; % propagating and reflected up to the boundary at z<z0
        % focal
        Fieldxxf=(cx.*MTx1.*Eincx.*exp(1i*phiincx)+cy.*MTx2.*Eincy.*exp(1i*phiincy)).*Hzf; %integration x component
        Exinff=ifftshift(ifft2(fftshift((Fieldxxf)))).*exp(1i*PhaseShift);
        Exf=Exinff((L/2+1−xmax2):(L/2+1+xmax2),(L/2+1−xmax2):(L/2+1+xmax2)); %cropped

        % y component Einf
        Fieldxyf=(cx.*MTy1.*Eincx.*exp(1i*phiincx)+cy.*MTy2.*Eincy.*exp(1i*phiincy)).*Hzf; %integration y component
        Eyinff=ifftshift(ifft2((fftshift(Fieldxyf)))).*exp(1i*PhaseShift);
        Eyf=Eyinff((L/2+1−xmax2):(L/2+1+xmax2),(L/2+1−xmax2):(L/2+1+xmax2));

        % z component Einf
        Fieldxzf=(cx.*MTz1.*Eincx.*exp(1i*phiincx)+cy.*MTz2.*Eincy.*exp(1i*phiincy)).*Hzf; %integration z component
        Ez0f=ifftshift(ifft2((fftshift(Fieldxzf)))).*exp(1i*PhaseShift);
        Ezf=Ez0f((L/2+1−xmax2):(L/2+1+xmax2),(L/2+1−xmax2):(L/2+1+xmax2));

        % total field at the material n2 incident+reflected
        Ex(:,:,j)=Exf+Exr;
        Ey(:,:,j)=Eyf+Eyr;
        Ez(:,:,j)=Ezf+Ezr;

    else
        % x component Einf_r
        Fieldxxr=(cx.*MTx1r.*Eincx.*exp(1i*phiincx)+cy.*MTx2r.*Eincy.*exp(1i*phiincy)).*Hzr; %integration
        Exinfr=ifftshift(ifft2((fftshift(Fieldxxr)))).*exp(1i*PhaseShift);
        Ext=Exinfr((L/2+1−xmax2):(L/2+1+xmax2),(L/2+1−xmax2):(L/2+1+xmax2)); %cropped

        % y component Einf_r
        Fieldxyr=(cx.*MTy1r.*Eincx.*exp(1i*phiincx)+cy.*MTy2r.*Eincy.*exp(1i*phiincy)).*Hzr; %integration
        Eyinfr=ifftshift(ifft2((fftshift(Fieldxyr)))).*exp(1i*PhaseShift);
        Eyr=Eyinfr((L/2+1−xmax2):(L/2+1+xmax2),(L/2+1−xmax2):(L/2+1+xmax2));

        % z component Einf_r
        Fieldxzx=(cx.*MTz1r.*Eincx.*exp(1i*phiincx)+cy.*MTz2r.*Eincy.*exp(1i*phiincy)).*Hzr; %integration
        Ez0r=ifftshift(ifft2((fftshift(Fieldxzx)))).*exp(1i*PhaseShift);
        Ezr=Ez0r((L/2+1−xmax2):(L/2+1+xmax2),(L/2+1−xmax2):(L/2+1+xmax2));

        % total field at the material n2 incident+reflected
        Ex(:,:,j)=Exf+Exr;
        Ey(:,:,j)=Eyf+Eyr;
        Ez(:,:,j)=Ezf+Ezr;

    end

end
4

% y component E_{inf t}
Fieldxyt=(cx.*MTyt.*E_{incx}.*exp(li*phi_{incx})+cy.*MTyt.*E_{incy}.*exp(li*phi_{incy})).*Hzt;

% integration
Ey_{inf t}=ifftshift(ifft2((fftshift(Fieldxyt)))).*exp(li*PhaseShift);
Ey=Ey_{inf t}((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2)); % cropped

% z component E_{inf t}
Fieldxzt=(cx.*MTzt.*E_{incx}.*exp(li*phi_{incx})+cy.*MTzt.*E_{incy}.*exp(li*phi_{incy})).*Hzt;

% integration
Ez0t=ifftshift(ifft2((fftshift(Fieldxzt)))).*exp(li*PhaseShift);
Ez=Ez0t((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2)); % cropped

Ex(:,:,j)=Ext; % field beyond the boundary, transmitted
Ey(:,:,j)=Eyt;
Ez(:,:,j)=Ezt;
end

%% plot yz cross section through the center at xmax2+1
figure(1)
EEx=permute(Ex(xmax2+1,:,:),[3 2 1]);
EEy=permute(Ey(xmax2+1,:,:),[3 2 1]);
EEz=permute(Ez(xmax2+1,:,:),[3 2 1]);
I=abs(EEx).^2+abs(EEy).^2+abs(EEz).^2;
Imax=max(max(I));
In=I/Imax;
imagesc((In).^(1/4)); colormap(jet); colorbar