

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/365371945>

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

Preprint · November 2022

DOI: 10.48550/arXiv.2211.06725

CITATION

1

READS

50

2 authors, including:



Israel Herrera

Institut Fresnel

7 PUBLICATIONS 4 CITATIONS

SEE PROFILE

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

Isael Herrera^{1,*} and Pedro A. Quinto-Su^{1,†}

¹*Instituto 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 posses 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: Infox [3] and Pyfocus [4], which have helped increase the availability of these tools to experimental groups. Infox 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 [5, 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.

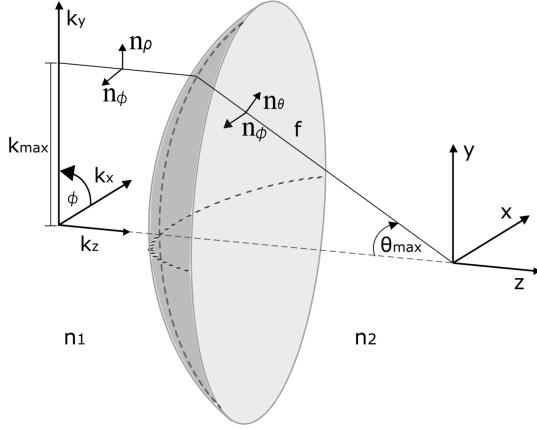


FIG. 1. Representation of the focusing model. A paraxial beam propagating in a medium with refractive index n_1 , incident into lens that focuses at f in a medium with a refractive index n_2 . The lens is represented by a spherical surface of reference centered at the focal point with radius f .

GENERAL THEORY: RICHARDS WOLF MODEL

The model of Richards and Wolf [1] is derived from energy conservation, the sine condition, and the fact that the field at the lens surface (represented by conjugate rays in geometrical optics) corresponds to the far field \mathbf{E}_∞ of the focused field \mathbf{E} .

The incident beam (\mathbf{E}_{inc}) is focused by an aplanatic lens (Figure 1). Inside the spherical surface that focuses the light, the refractive index is n_2 ($n_2 = 1.518$ for glass and immersion oil). We assume the incident light propagates from left to right in air ($n_1 = 1$) with a transverse polarization state before it is focused by the lens. Here we only consider the light that is transmitted by the lens which is focused, so we assume an unitary transmission coefficient. Upon focusing, each transverse polarization component acquires additional components on the other spatial directions. The fields are described using spherical coordinates which naturally match the directions of the parallel and perpendicular components of the electric fields to the surface of the aplanatic lens.

The relations between the spherical and Cartesian unitary vectors are:

$$\begin{aligned}\mathbf{n}_\phi &= -\sin(\phi)\mathbf{n}_x + \cos(\phi)\mathbf{n}_y \\ \mathbf{n}_\rho &= \cos(\phi)\mathbf{n}_x + \sin(\phi)\mathbf{n}_y \\ \mathbf{n}_\theta &= \cos(\theta)\cos(\phi)\mathbf{n}_x + \cos(\theta)\sin(\phi)\mathbf{n}_y - \sin(\theta)\mathbf{n}_z\end{aligned}\quad (1)$$

If the incident field is linearly polarized in the x direction, it can be expressed as $E_{inc}^x = E_{0x}e^{i\phi_{0x}}\mathbf{n}_x$, where E_{inc}^x and ϕ_{0x} are the amplitude and the phase and \mathbf{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_∞^x which has the 3 polarization components (in a Cartesian basis).

$$\mathbf{E}_\infty^x = \sqrt{\frac{n_1}{n_2}} \sqrt{\cos(\theta)} E_{inc}^x \left\{ -\sin(\phi) \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{pmatrix} + \cos(\phi) \begin{pmatrix} \cos(\theta)\cos(\phi) \\ \cos(\theta)\sin(\phi) \\ -\sin(\theta) \end{pmatrix} \right\} \quad (2)$$

While for the case of an incident field $\mathbf{E}_{inc} = E_{inc}^y\mathbf{n}_y$ with $E_{inc}^y = E_{0y}e^{i\phi_{0y}}$. The E_∞^y representation of the focused field is:

$$\mathbf{E}_\infty^y = \sqrt{\frac{n_1}{n_2}} \sqrt{\cos(\theta)} E_{inc}^y \left\{ \cos(\phi) \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{pmatrix} + \sin(\phi) \begin{pmatrix} \cos(\theta)\cos(\phi) \\ \cos(\theta)\sin(\phi) \\ -\sin(\theta) \end{pmatrix} \right\} \quad (3)$$

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

$$\begin{aligned}\frac{x_\infty}{f} &= \frac{k_x}{k} = \sin(\theta)\cos(\phi) \\ \frac{y_\infty}{f} &= \frac{k_y}{k} = \sin(\theta)\sin(\phi) \\ \frac{z_\infty}{f} &= \frac{k_z}{k} = \cos(\theta)\end{aligned}\quad (4)$$

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

The expressions for \mathbf{E}_∞^x and \mathbf{E}_∞^y are:

$$\mathbf{E}_\infty^x = \sqrt{\frac{n_1}{n_2}} \sqrt{\frac{k_z}{k}} E_{inc}^x \left(\frac{k_x}{k}, \frac{k_y}{k} \right) \begin{pmatrix} k_y^2 + k_x^2 k_z/k \\ -k_x k_y + k_x k_y k_z/k \\ -(k_x^2 + k_y^2) k_x/k \end{pmatrix} \left(\frac{1}{k_x^2 + k_y^2} \right) \quad (5)$$

For the other transverse polarization component (direction \mathbf{n}_y).

$$\mathbf{E}_\infty^y = \sqrt{\frac{n_1}{n_2}} \sqrt{\frac{k_z}{k}} E_{inc}^y \left(\frac{k_x}{k}, \frac{k_y}{k} \right) \begin{pmatrix} -k_x k_y + k_x k_y k_z/k \\ k_x^2 + k_y^2 k_z/k \\ -(k_x^2 + k_y^2) k_y/k \end{pmatrix} \left(\frac{1}{k_x^2 + k_y^2} \right) \quad (6)$$

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

$$\mathbf{E}(x, y, z) = -\frac{ikfe^{-ikf}}{2\pi} \iint_{k_x^2 + k_y^2 \leq k_{max}^2} [c_x \mathbf{E}_\infty^x + c_y \mathbf{E}_\infty^y] \frac{1}{k_z} \times e^{i[k_x x + k_y y + k_z z]} dk_x dk_y \quad (7)$$

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 $k_x^2 + k_y^2 \leq k_{max}^2$, where in terms of the numerical aperture ($NA = n \sin \theta_{max}$) as $k_{max} = NAk_0$. The same limit of k_{max} can also be expressed in terms of the aperture radius R using equation 4: $R = k_{max}f/k = NAf/n$ (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 :

$$\Theta(k_x, k_y) = \begin{cases} 1 & \text{if } k_x^2 + k_y^2 \leq k_{max}^2 \\ 0 & \text{if } k_x^2 + k_y^2 > k_{max}^2 \end{cases} \quad (8)$$

$$\mathbf{E}(x, y, z) = -\frac{ife^{-ikf}}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Theta(k_x, k_y) [c_x \mathbf{E}_{\infty}^x + c_y \mathbf{E}_{\infty}^y] \frac{1}{k_z} \times e^{i[k_x x + k_y y + k_z z]} dk_x dk_y \quad (9)$$

The previous expression can be rewritten as:

$$\mathbf{E}(x, y, z) = -\frac{ife^{-ikf}}{2\pi} \text{IFT} \left[\Theta(k_x, k_y) [c_x \mathbf{E}_{\infty}^x + c_y \mathbf{E}_{\infty}^y] \frac{1}{k_z} e^{ik_z z} \right] \quad (10)$$

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_{inc} is described. This sets the

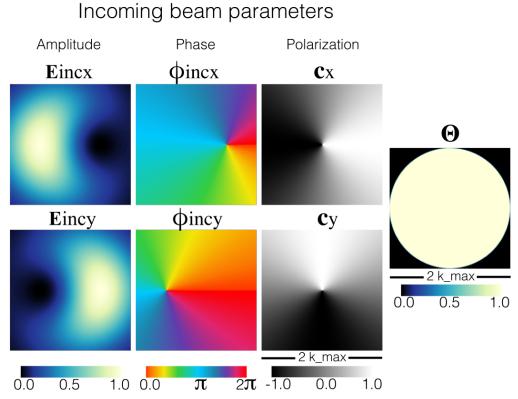


FIG. 2. Input field. The input plane where the incident field is described, amplitude, phase, polarization and aperture can be defined as functions of the angular representation (k_x, k_y) or can also be written in terms of the spatial coordinates (x_{∞}, y_{∞}) . In this way, the points that span the aperture have a radius R or k_{max} .

spatial dimension of the incident field, so it also can be written as a function of (x_{∞}, y_{∞}) (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_{∞}, y_{∞}) , 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 $E_{inc,x} e^{i\phi_x}$ and $E_{inc,y} e^{i\phi_y}$ 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_{inc} but we decided to separate them to mimic the experiment. Finally there is the aperture Θ that can also be controlled by the beam shaping elements and limited by the lens. Those 7 transverse 2d masks ($E_{inc,x}, E_{inc,y}, \phi_{incx}, \phi_{incy}, c_x, c_y, \Theta$) are drawn in Fig. 2 and represent E_{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_∞, y_∞) 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_∞, y_∞ , 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 Δx and Δk 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 $\phi_{shift}(X, Y) = -i2\pi \frac{m_0}{L} X + i2\pi \frac{m_0}{L} Y$. Where X and

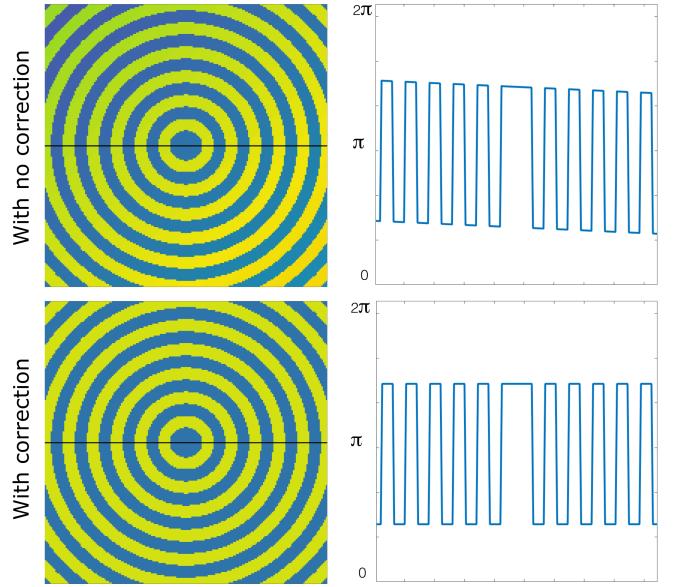


FIG. 3. Effect of the origin translation in the phase of the resulting complex fields and correction. In the first column it is shown the phase of the x component of a focused x-linearly polarized Gaussian beam, in the second column it is shown the phase evaluated on the horizontal axis indicated on the phase map.

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 $-\phi_{shift}(X, Y)$.

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 π . 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_∞, x_∞)

At the back aperture of the microscope objective the spatial domain x_∞ and y_∞ is defined multiplying the discrete point grid by the constant Δx_∞ which converts it into a spatial grid (dimensions of micrometers in the code). Δx_∞ 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 $k_1 = n_2 k_0$ 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 $\Delta x_f = 2\pi/(L\Delta k) = N\lambda/(L(NA))$ ($\Delta x_f = \Delta y_f$).

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 ϕ_{x0}, ϕ_{y0} , aperture Θ 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 $a e^{ib}$, with $\text{abs}(a) = 1$. As we mentioned before, in the experiments sometimes it is more convenient to use the spatial coordinates (x_∞, y_∞) 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_∞, y_∞) , in the case of a circular aperture:

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

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

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 $\Theta(k_x, k_y)$

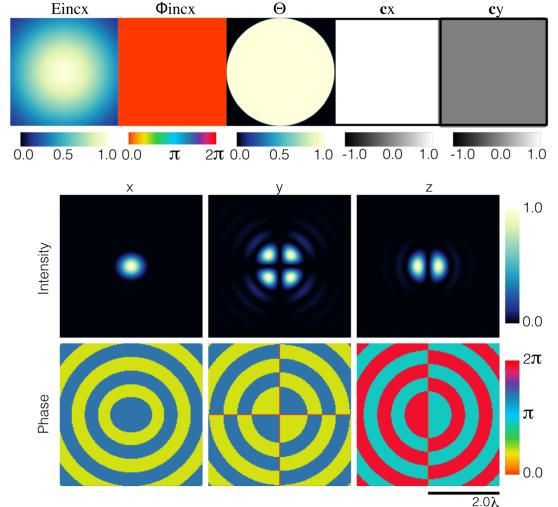


FIG. 4. Focusing a linearly polarized Gaussian beam. The parameters are: $L = 2^{12}$, $N = 2^7$, $n_2 = 1.518$, NA= 1.3, $f = 2$ mm.

is:

$$\Theta(k_x, k_y) = \frac{1}{4} \left(1 + \tanh \left[\frac{3}{4\Delta k} \left(\frac{k_{max}}{\sqrt{2}} - k_x \right) \right] \right) \times \left(1 + \tanh \left[\frac{3}{4\Delta k} \left(\frac{k_{max}}{\sqrt{2}} - k_y \right) \right] \right) \quad (12)$$

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:

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

where $k_{max} \geq k_{max1} > k_{max2}$ and $\Delta k_t = k_{max1} - k_{max2}$ 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].

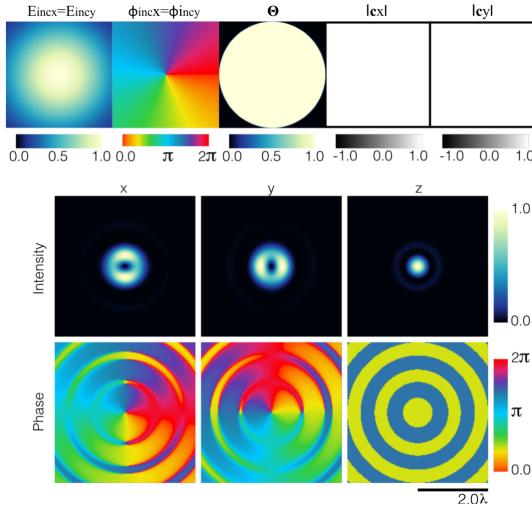


FIG. 5. Focusing a Gaussian beam right circular polarization, $\sigma = 1$, and vortex phase of order $m = -1$. The parameters are: $L = 2^{12}$, $N = 2^7$, $n_2 = 1.518$, NA = 1.3, $f = 2$ mm.

Linearly polarized Gaussian

We consider a Gaussian beam defined as:

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

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 ω_0 the Gaussian waist. The same field in angular coordinates:

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

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 \mathbf{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.

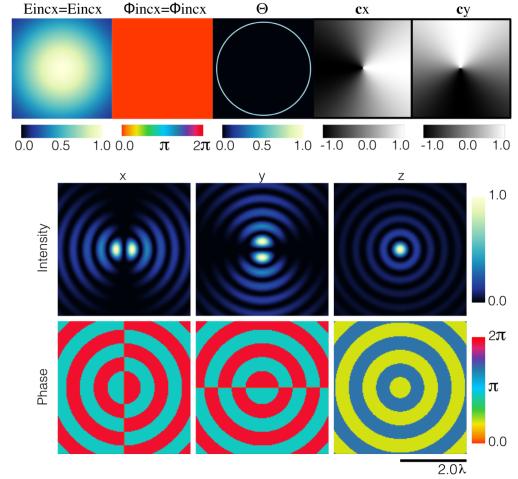


FIG. 6. Focusing a radial polarized annular aperture (TM 0th). The parameters are: $L = 2^{12}$, $N = 2^7$, $n_2 = 1.518$, NA = 1.3, $f = 2$ mm.

Circular polarization

Same Gaussian with an optical vortex phase with $\phi_{0x} = \phi_\infty m$ $m = -1$ through a quarter wave plate, where ϕ_∞ is the azimuthal angle in the x_∞, y_∞ 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_∞, y_∞ (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 σ 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 ϕ and ρ in the x_∞, y_∞ domain. The angle ϕ 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 $c_x = \cos \phi = -k_y / \sqrt{k_x^2 + k_y^2}$ $c_y = \sin \phi = k_x / \sqrt{k_x^2 + k_y^2}$. 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} .

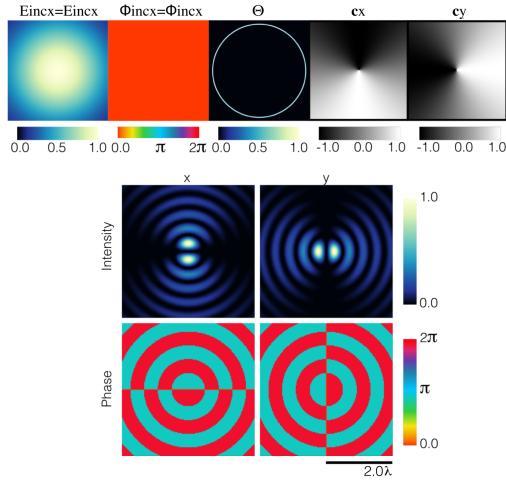


FIG. 7. Focusing a azimuthal polarized annular aperture (TE 0th). The parameters are: $L = 2^{12}$, $N = 2^7$, $n_2 = 1.518$, NA= 1.3, $f = 2$ mm.

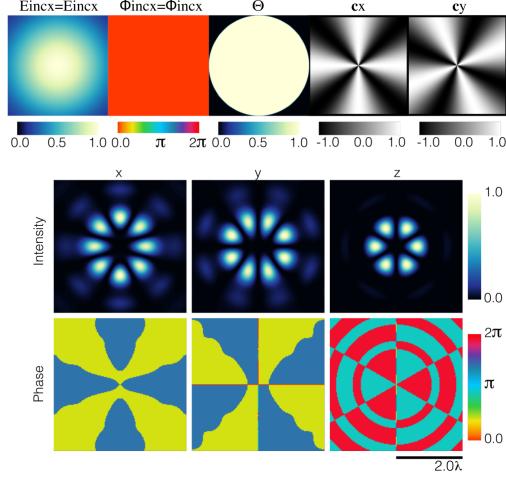


FIG. 8. Focusing a Gaussian beam with flower (order 8) polarization: The parameters are $L = 2^{12}$, $N = 2^7$, $n_2 = 1.518$, NA= 1.3, $f = 2$ mm.

Arbitrary Polarization

We also consider a Gaussian beam with flower (spider web) polarization state as described in [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:

$$\begin{aligned} E_{incx}(k_x, k_y) &= E_{0x} \sqrt{(k_x - k_{max}/2)^2 + k_y^2} e^{-\frac{(k_x^2+k_y^2)}{k_{max}^2}} \\ E_{incx}(k_x, k_y) &= E_{0y} \sqrt{(k_x + k_{max}/2)^2 + k_y^2} e^{-\frac{(k_x^2+k_y^2)}{k_{max}^2}} \\ \phi_{incx} &= \tan^{-1}[k_y/(k_x - k_{max}/2)] \\ \phi_{incy} &= \tan^{-1}[k_y/(k_x + k_{max}/2)] \\ c_{incx} &= k_x / \sqrt{k_x^2 + k_y^2} \\ c_{incy} &= k_y / \sqrt{k_x^2 + k_y^2} \end{aligned} \quad (16)$$

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

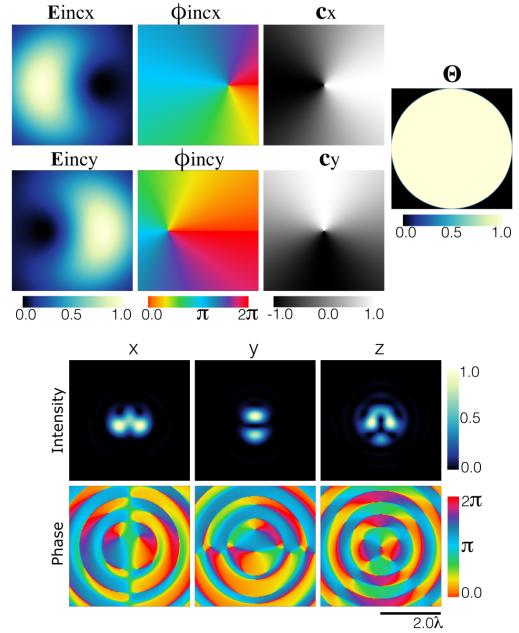


FIG. 9. Focusing of a beam with different amplitudes, phases, and polarization profiles in the transverse components. The first group of images shows the amplitude, phase, polarization maps and the circular aperture. The second group shows the focal fields. The parameters are $L = 2^{12}$, $N = 2^7$, $n_2 = 1.518$, 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 [6]. 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 $k_{z1} = \sqrt{k_1^2 - (k_x^2 + k_y^2)}$ ($k_1 = nk_0$), above the boundary for $z > z_0$, the refractive index is n_2 and $k_{z2} = \sqrt{k_2^2 - (k_x^2 + k_y^2)}$ ($k_2 = n_2k_0$).

Fresnel Coefficients

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

$$\begin{aligned} r^s(k_x, k_y) &= \frac{\mu_2 k_{z1} - \mu_1 k_{z2}}{\mu_2 k_{z1} + \mu_1 k_{z2}} \\ r^p(k_x, k_y) &= \frac{\epsilon_2 k_{z1} - \epsilon_1 k_{z2}}{\epsilon_2 k_{z1} + \epsilon_1 k_{z2}} \\ t^s(k_x, k_y) &= \frac{2\mu_2 k_{z1}}{\mu_2 k_{z1} + \mu_1 k_{z2}} \\ t^p(k_x, k_y) &= \frac{2\epsilon_2 k_{z1}}{\epsilon_2 k_{z1} + \epsilon_1 k_{z2}} \sqrt{\frac{\mu_2 \epsilon_1}{\mu_1 \epsilon_2}} \end{aligned} \quad (17)$$

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 $\mathbf{E} = \mathbf{E}_f(x, y, z) + \mathbf{E}_r(x, y, z)$ for $z < z_0$:

$$\mathbf{E}_f(x, y, z) = -\frac{ife^{-ikf}}{2\pi} \iint_{-\infty}^{\infty} [c_x \mathbf{E}_{\infty, f}^x + c_y \mathbf{E}_{\infty, f}^y] \times \frac{1}{k_z} e^{ik_x x + ik_y y + ik_{z1} z} dk_x dk_y \quad (18)$$

$$\mathbf{E}_r(x, y, z) = -\frac{ife^{-ikf}}{2\pi} \iint_{-\infty}^{\infty} [c_x \mathbf{E}_{\infty, r}^x + c_y \mathbf{E}_{\infty, r}^y] \times \frac{1}{k_z} e^{ik_x x + ik_y y - ik_{z1} z} dk_x dk_y \quad (19)$$

In the case of $z > z_0$ then $\mathbf{E} = \mathbf{E}_t$ where the transmitted field:

$$\mathbf{E}_t(x, y, z) = -\frac{ife^{-ikf}}{2\pi} \iint_{-\infty}^{\infty} [c_x \mathbf{E}_{\infty, t}^x + c_y \mathbf{E}_{\infty, t}^y] \times \frac{1}{k_z} e^{ik_x x + ik_y y + ik_{z2} z} dk_x dk_y \quad (20)$$

The reflected $\mathbf{E}_{\infty, r}$ and transmitted $\mathbf{E}_{\infty, t}$ in terms of the xy components of \mathbf{E}_{inc} :

$$\begin{aligned} \mathbf{E}_{\infty, r}^x &= \sqrt{\frac{k_{z1}}{k_1}} E_{inc}^x \left(\frac{k_x}{k}, \frac{k_y}{k} \right) \left(\frac{e^{2ik_{z1} z_0}}{k_x^2 + k_y^2} \right) \times \\ &\quad \left(\begin{array}{l} r^s k_y^2 - r^p k_x^2 k_{z1} / k_1 \\ -r^s k_x k_y - r^p k_x k_y k_{z1} / k_1 \\ -r^p (k_x^2 + k_y^2) k_x / k_1 \end{array} \right) \end{aligned} \quad (21)$$

For the other transverse polarization component (direction \mathbf{n}_y):

$$\begin{aligned} \mathbf{E}_{\infty, r}^y &= \sqrt{\frac{k_{z1}}{k_1}} E_{inc}^y \left(\frac{k_x}{k}, \frac{k_y}{k} \right) \left(\frac{e^{2ik_{z1} z_0}}{k_x^2 + k_y^2} \right) \times \\ &\quad \left(\begin{array}{l} -r^s k_x k_y - r^p k_x k_y k_z / k_1 \\ r^s k_x^2 - r^p k_y^2 k_{z1} / k_1 \\ -r^p (k_x^2 + k_y^2) k_y / k_1 \end{array} \right) \end{aligned} \quad (22)$$

The transmitted component:

$$\begin{aligned} \mathbf{E}_{\infty, t}^x &= \sqrt{\frac{k_{z1}}{k_1}} E_{inc}^x \left(\frac{k_x}{k}, \frac{k_y}{k} \right) \left(\frac{e^{i(k_{z1} - k_{z2}) z_0}}{k_x^2 + k_y^2} \right) \times \\ &\quad \left(\begin{array}{l} t^s k_y^2 + t^p k_x^2 k_{z2} / k_2 \\ -t^s k_x k_y + t^p k_x k_y k_{z2} / k_2 \\ -t^p (k_x^2 + k_y^2) k_x / k_2 \end{array} \right) \frac{k_{z2}}{k_{z1}} \end{aligned} \quad (23)$$

For the other transverse polarization component (direction \mathbf{n}_y):

$$\begin{aligned} \mathbf{E}_{\infty, t}^y &= \sqrt{\frac{k_{z1}}{k_1}} E_{inc}^y \left(\frac{k_x}{k}, \frac{k_y}{k} \right) \left(\frac{e^{i(k_{z1} - k_{z2}) z_0}}{k_x^2 + k_y^2} \right) \times \\ &\quad \left(\begin{array}{l} -t^s k_x k_y + t^p k_x k_y k_{z2} / k_2 \\ t^s k_x^2 + t^p k_y^2 k_{z2} / k_2 \\ -t^p (k_x^2 + k_y^2) k_y / k_2 \end{array} \right) \frac{k_{z2}}{k_{z1}} \end{aligned} \quad (24)$$

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, z with $y = 0$ and the second row plane y, z with $x = 0$. The colormaps represent the fourth root of total intensity in order to show more clearly the details. The case with $z_0 = 0$ appears in [6].

CONCLUSION

We presented two simple programs to calculate tightly focused vectorial light fields: 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, 6]. 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 [6].

Acknowledgements

Work partially funded by DGAPA UNAM PAPIIT grants IN107719 and IN107222; CTIC-LANMAC and

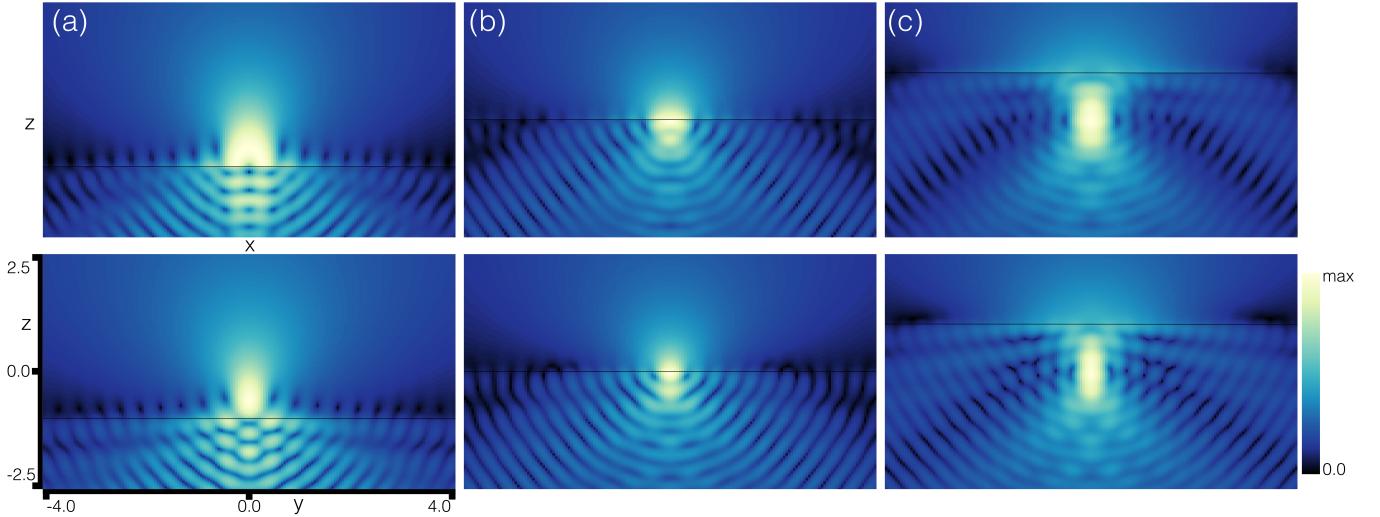


FIG. 10. Intensity $\mathbf{E} \cdot \mathbf{E}^*$ of a focused linearly polarized Gaussian beam with parameters $f_0 = 1$, NA= 1.4, close to an interface glass-air. The spanned area $5\lambda \times 8\lambda$ is the same for all the images, the $(0, 0)$ is the focus of the lens. In the first row is 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

-
- * 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.Caprile, 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).

APPENDIX A: PROGRAM TO CALCULATE PROPAGATING BEAM

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 clear all; %close all; clc
3 %run time sim 10s with L=2^12, and N=2^7
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%initial parameters
6 L=2^12; % main domain size
7 N=2^7; % number of points for the aperture radius
8 z=0; %observation plane (0 at the focal plane)
9 m0=1/2; %coordinate shift
10 xmax=0.9; %half the size of the cropped output in wavelengths
11 %(final size of images is 2xmax)
12 n1=1.0; % refractive index before the lens (air)
13 n2=1.518; % refractive index after the lens (immersion oil)
14 f=1800; % effective focal length of the lens in um (10^-6 m)
15 NA=1.4; %effective numerical aperture
16 lambda=1.064; %wavelength in um
17 R=f*NA/n2; %aperture radius in um;
18 dx_inf=R/N; %spatial resolution at aperture um/px or um/point
19
20 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
21 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Spatial coordinate system before lens (x.inf, y.inf)
22 d_inf=linspace(-L/2+m0,L/2-m0,L)*dx_inf; %spatial axis shifted by m0.
23 [x_inf,y_inf]=meshgrid(d_inf,-d_inf); %mesh, y inverted in Matlab images
24 [theta_inf,rho_inf] = cart2pol(x_inf,y_inf); %auxiliary polar coordinates
25
26 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% angular
28 k0=2*pi/lambda;
29 k1=n2*k0;
30 dk=k0*NA/N; %dk=(1/2)*(k1/(f))*dx0;
31 kx=x_inf*k1/f;
32 ky=y_inf*k1/f;
33 dxf=2*pi/(L*dk); %equivalent to dxf=N*lambda/(L*NA);
34 %conversion factor at output plane in um/px
35 kM1=k1*ones(L,L); %%array with magnitude of k1
36 kz1=sqrt(kM1.^2-kx.^2-ky.^2);
37
38 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39 xmax2=round(xmax*lambda/dxf); %xmax in px at output plane
40
41 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
42 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%correction for m0 shift
43 x=linspace(-L/2,L/2,L); %%auxiliary coordinate for correction
44 y=x; [X,Y]=meshgrid(x,y);
45 PhaseShift=m0*2*pi*X/L+m0*2*pi*Y/L; %correction phase
46 %Center of Fourier transform displaced (vert and hor) by m0.
47 %shift of \Delta kx=\Delta ky= m0*(2*pi/L).
48
49 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
50 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%input field initial field transverse
51 w0=f*NA/n2; % The beam waist is equal to the aperture radius.
52 E_incx=exp(-(x_inf.^2+y_inf.^2)/(w0^2)); %normalized amplitude E_x0 A_0
53 phi_incx=double(zeros(L,L)); %%constant
54 E_incy=E_incx;
55 phi_incy = phi_incx;
56
57 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
58 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%round soft aperture (spatial coordinates)
59 Theta=0.5*(1+tanh((1.5/1)*(N-sqrt((x_inf/dx_inf).^2+(y_inf/dx_inf).^2)))); %angular space
60 %kmax=R*k1/f;
61 %Theta=0.5*(1+tanh((1.5/dk)*(kmax-sqrt((kx).^2+(ky).^2)))); %polarization state
62 %cx=ones(L,L); cy=zeros(L,L); % lin horizontal

```

```

67 cx=zeros(L,L); cy=ones(L,L); % lin vertical
68 %cx=1*ones(L,L); cy=1*ones(L,L); % lin diag
69 %cx=ones(L,L); cy=li*ones(L,L); %%right circ
70 %cx=li*ones(L,L); cy=ones(L,L); %%left circ
71 %cx=kx./sqrt(kx.^2+ky.^2); cy=ky./sqrt(kx.^2+ky.^2); %%radial
72 %cx=-ky./sqrt(kx.^2+ky.^2); cy= kx./sqrt(kx.^2+ky.^2);%%azimuthal
73 %phi_aux=mod(atan2(ky,kx),2*pi); s=8; cx=cos((s/2)*phi_aux); cy=sin((s/2)*phi_aux); %flower
74
75
76
77 %%%%%%%%%%%%%%
78 %%%%%%E.inf %%%%%%
79 %input x
80 %x component
81 E_infxx=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_incx.*exp(li*phi_incx).*...
82 (ky.^2+kx.^2.*kz1./kM1)./(kx.^2+ky.^2);
83 %y component
84 E_infxy=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_incx.*exp(li*phi_incx).*...
85 (-kx.*ky+kx.*ky.*kz1./kM1)./(kx.^2+ky.^2);
86 %z component
87 E_infxz=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_incx.*exp(li*phi_incx).*...
88 (-kx.^2+ky.^2).*kx./kM1)./(kx.^2+ky.^2);
89
90 %y input
91 %x component
92 E_infyx=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_incy.*exp(li*phi_incy).*...
93 (-ky.*kx+kx.*ky.*kz1./kM1)./(kx.^2+ky.^2);
94 %y component
95 E_infyx=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_incy.*exp(li*phi_incy).*...
96 (kx.^2+ky.^2.*kz1./kM1)./(kx.^2+ky.^2);
97 %z component
98 E_infyz=sqrt(n1/n2)*sqrt(kz1./kM1).*Theta.*E_incy.*exp(li*phi_incy).*...
99 (-kx.^2+ky.^2).*ky./kM1)./(kx.^2+ky.^2);
100
101 %factors to assemble E_inf
102 CF2=-li*f*exp(-li*f*2*pi*n2/lambda)*exp(li*kz1.*z)./(2*pi*kz1);
103
104 %%E_inf
105 Fieldx=CF2.* (cx.*E_infxx +cy.*E_infyx);
106 Fieldy=CF2.* (cx.*E_infxy +cy.*E_infyx);
107 Fieldz=CF2.* (cx.*E_infxz +cy.*E_infyz);
108
109 %%%%%%
110 %%Integration
111 Ex0=ifftshift(ifftshift((Fieldx))).*exp(li*PhaseShift);
112 Ex=Ex0((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2)); %cropped
113 %x=y=0 at the center
114
115 Ey0=ifftshift(ifft2((fftshift(Fieldy))).*exp(li*PhaseShift));
116 Ey=Ey0((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2)); %cropped
117
118 Ez0=ifftshift(ifft2((fftshift(Fieldz))).*exp(li*PhaseShift));
119 Ez=Ez0((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2)); %cropped
120
121 %%%%%%
122 %%%%%%
123
124 %intensity ratios
125 c0y=max(max(abs(Ey).^2))/max(max(abs(Ex).^2));
126 c0z=max(max(abs(Ez).^2))/max(max(abs(Ex).^2));
127
128 %%%%%%
129 %plots
130
131
132 a='jet';
133 b='gray';
134
135 figure(1) %amplitude Ex
136 Exn=abs(Ex)./max(max(abs(Ex)));

```

APPENDIX B: PROGRAM FOR INTERFACE

```

1 %%%%%%%%%%%%%%%%
2 clear all; %
3 %%%%%%%%%%%%%%%%
4 %%initial parameters
5 L=2^10; % main domain size
6 N=2^6; % number of points for the aperture radius
7 z0=0; %interface position (focus at z=0)
8 z=linspace(-2,2,41); %axial domain
9 m0=1/2; %coordinate shift
10 xmax=4; %half the size of the cropped output in wavelengths
11 %(final size of images is 2xmax)
12 n1=1.518; %refractive index for incident field (immersion oil)
13 n2=1.518; % refractive index for reflected field (immersion oil)
14 n3=1; % refractive index for transmitted field (after interface)
15 e1=n2^2;
16 e2=n3^2;
17 m1=1; %permeability
18 m2=1; %permeability
19 f=1800; % effective focal length of the lens in um ( $10^{-6}$  m)
20 NA=1.3; %effective numerical aperture
21 lambda=1.064; %wavelength in um
22 l=1.064; %wavelength um
23 R=f*NA/n2; %aperture radius in um;
24 dx_inf=R/N; %spatial dimension at aperture um/px or um/point
25
26 %%%%%%%%%%%%%%%
27 %% Spatial coordinate system before lens (x.inf, y.inf)
28 d_inf=linspace(-L/2+m0,L/2-m0,L)*dx_inf; %spatial axis shifted by m0.
29 [x_inf,y_inf]=meshgrid(d_inf,-d_inf); %mesh, y inverted in Matlab images
30 [theta_inf,rho_inf] = cart2pol(x_inf,y_inf); %auxiliary polar coordinates
31 %%%%%%%%%%%%%%%
32
33 %% %%%%%%% angular
34 %% %%%%%%
35 k0=2*pi/lambda;
36 k1=n2*k0;
37 k2=n3*k0;
38 dk=k0*NA/N; %dk=(1/2)*(k1/(f))*dx0;
39 kx=x_inf*k1/f;
40 ky=y_inf*k1/f;
41 dxf=2*pi/(L*dk); %equivalent to dxf=N*lambda/(L*NA);
42 %conversion factor at output plane in um/px
43 kM1=k1*ones(L,L); %%array with magnitude of k1
44 kz1=sqrt(kM1.^2-kx.^2-ky.^2);
45 kM2=k2*ones(L,L);
46 kz2=sqrt(kM2.^2-kx.^2-ky.^2);
47
48 %%%%%%%%
49 xmax2=round(xmax*lambda/dxf); %xmax in px at output plane
50
51 %%%%%%%%
52 %%correction for m0 shift
53 x=linspace(-L/2,L/2,L); %%auxiliary coordinate for correction
54 y=x; [X,Y]=meshgrid(x,y);
55 PhaseShift=m0*2*pi*X/L+m0*2*pi*Y/L; %correction phase
56 %Center of Fourier transform displaced (vert and hor) by m0.
57 %shift of \Delta kx=\Delta ky= m0*(2*pi/L).
58
59 %%%%%%%%
60 %%input field initial field transverse
61 w0=f*NA/n2; % The beam waist is equal to the aperture radius.
62 E_incx=exp(-(x_inf.^2+y_inf.^2)/(w0^2)); %normalized amplitude E_x0 A_0
63 phi_incx=double(zeros(L,L)); %%phase phi_x0 fase init; phi_0
64 E_incy=E_incx;
65 phi_incy = phi_incx;
66

```

```

67 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
68 %%round soft aperture (spatial coordinates)
69 Theta=0.5*(1+tanh((1.5/1)*(N-sqrt((x_inf/dx_inf).^2+(y_inf/dy_inf).^2)))); 
70 %angular space
71 %kmax=R*k1/f;
72 %Theta=0.5*(1+tanh((1.5/dk)*(kmax-sqrt((kx).^2+(ky).^2)))); 
73
74
75
76 %%%%%%
77 %%polarization
78 CX=ones(L,L); cy=zeros(L,L); % lin horizontal
79 %cx=zeros(L,L); cy=ones(L,L); % lin vertical
80 %cx=1*ones(L,L); cy=1*ones(L,L); % lin diag
81 %cx=ones(L,L); cy=1i*ones(L,L); %%right circ
82 %cx=1i*ones(L,L); cy=ones(L,L); %%left circ
83 %cx=(kx)./sqrt(kx.^2+ky.^2); cy=(ky)./sqrt(kx.^2+ky.^2); %%radial cx=cos(phi) cy=sin(phi)
84 %cx=-ky./sqrt(kx.^2+ky.^2); cy= kx./sqrt(kx.^2+ky.^2);%%azimuthal cx=cos(phi) cy=sin(phi)
85 %phik=atan2(ky,kx); s12=8; cx=cos((s12/2)*phik); cy=sin((s12/2)*phik);%flower
86 %phik=atan2(ky,kx); s12=-8; cx=cos((s12/2)*phik); cy=sin((s12/2)*phik);% spider
87
88 %%%%%%
89 %%Fresnel coeficients
90 rs=(m2*kz1-m1*kz2)./(m2*kz1+m1*kz2);
91 rp=(e2*kz1-e1*kz2)./(e2*kz1+e1*kz2);
92 ts=(2*m2*kz1)./(m2*kz1+m1*kz2);
93 tp=sqrt((m2*e1)/(m1*e2))*(2*e2*kz1)./(e2*kz1+e1*kz2);
94
95 %%%%%%
96 %%Factors for E.inf
97 CF1=-(1./kz1).*sqrt(kz1./kM1).* (1./(kx.^2+ky.^2))*li*f*exp(-li*f*2*pi*n2/lambda); %Factors for ...
E_inf_r and f
98 CF2=-(1./kz2).*sqrt(kz1./kM1).* (1./(kx.^2+ky.^2))*li*f*exp(-li*f*2*pi*n2/lambda).* (kz2./kz1); ...
%Factors for E.inf _t
99
100 %%%%%%
101 %%Focused propagating before interface
102 %for E.inf
103 % x component
104 MTx1=(ky.^2 +kx.^2.*kz1./kM1).*CF1; %lens transmision factor, incident x
105 MTx2=(-ky.*kx +(ky.*kx.*kz1)./kM1).*CF1; %lens transmission factor incident y
106 %y component
107 MTy1=(-kx.*ky+(kx.*ky.*kz1)./kM1).*CF1;%lens transmision factor, incident x
108 MTy2=(kx.^2 +ky.^2.*kz1./kM1).*CF1;%lens transmision factor, incident y
109 %z component
110 MTz1=(-(kx.^2+ky.^2).*kx./kM1).*CF1;%lens transmision factor, incident x
111 MTz2=(-(kx.^2+ky.^2).*ky./kM1).*CF1; %lens transmision factor, incident y
112
113 %%%%%%
114 %% Reflected E.inf.r
115 %x component
116 MTx1r=(rs.*ky.^2 -rp.*kx.^2.*kz1./kM1).*CF1; % incident x
117 MTx2r=(-rs.*ky.*kx -rp.* (ky.*kx.*kz1)./kM1).*CF1;% incident y
118 %y component
119 MTy1r=(-rs.*kx.*ky-rp.* (kx.*ky.*kz1)./kM1).*CF1;% incident x
120 MTy2r=(rs.*kx.^2 -rp.*ky.^2.*kz1./kM1).*CF1;% incident y
121 %z component
122 MTz1r=(-rp.* (kx.^2+ky.^2).*kx./kM1).*CF1;%incident x
123 MTz2r=(-rp.* (kx.^2+ky.^2).*ky./kM1).*CF1; % incident y
124
125 %%%%%%
126 %%Transmited E.inf.t
127 %x component
128 MTx1t=(ts.*ky.^2+tp.*kx.^2.*kz2./kM2).*CF2; % x input
129 MTx2t=(-ts.*ky.*kx+tp.* (ky.*kx.*kz2)./kM2).*CF2;% y input
130 %y component
131 MTy1t=(-ts.*kx.*ky+tp.* (kx.*ky.*kz2).* (1./kM2)).*CF2;% x input
132 MTy2t=(ts.*kx.^2+tp.*ky.^2.*kz2./kM2).*CF2;% y input
133 %z component
134 MTz1t=(-tp.* (kx.^2+ky.^2).*kx./kM2).*CF2;%x input

```

```

135 MTz2t=(-tp.* (kx.^2+ky.^2).*ky./kM2).*CF2; % y input
136
137 %%%%%%%%%%%%%%%%
138 %%%%%%%%start
139
140 for j=1:length(z);
141 Hzf=exp(li*kz1.*z(j)).*Theta; %propagating
142 Hxr=exp(-li*kz1.*z(j)+li*2*kz1*z0).*Theta; %reflected
143 Hzt=exp(li*kz2.*z(j)+li*(kz1-kz2)*z0).*Theta; %transmitted
144
145
146
147 if z(j)<=z0; % propagating and reflected up to the boundary at z<z0
148 %%%%%%%%
149 % x component E_inf
150 Fieldxxf=(cx.*MTx1.*E_incx.*exp(li*phi_incx)+cy.*MTx2.*E_incy.*exp(li*phi_incy)).*Hzf;
151 %integration x component
152 Ex_inff=ifftshift(ifft2(fftshift((Fieldxxf))).*exp(li*PhaseShift));
153 Exf=Ex_inff((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2)); %cropped
154
155 %y component E_inf
156 Fieldxyf=(cx.*MTy1.*E_incx.*exp(li*phi_incx)+cy.*MTy2.*E_incy.*exp(li*phi_incy)).*Hzf;
157 %integration y component
158 Ey_inff=ifftshift(ifft2((fftshift(Fieldxyf))).*exp(li*PhaseShift));
159 Eyf=Ey_inff((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2));
160
161 %z component E_inf
162 Fieldzxf=(cx.*MTz1.*E_incx.*exp(li*phi_incx)+cy.*MTz2.*E_incy.*exp(li*phi_incy)).*Hzf;
163 %integration z component
164 Ez0f=ifftshift(ifft2((fftshift(Fieldzxf))).*exp(li*PhaseShift));
165 Ezf=Ez0f((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2));
166 %%%%%%%%
167
168 %%%%%%%%
169 %%%%%%%%
170 %x component E_inf_r
171 Fieldxxr=(cx.*MTx1r.*E_incx.*exp(li*phi_incx)+cy.*MTx2r.*E_incy.*exp(li*phi_incy)).*Hxr;
172 %integration
173 Ex_infr=ifftshift(ifft2(fftshift((Fieldxxr))).*exp(li*PhaseShift));
174 Exr=Ex_infr((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2));
175
176 %y component E_inf_r
177 Fieldxyr=(cx.*MTy1r.*E_incx.*exp(li*phi_incx)+cy.*MTy2r.*E_incy.*exp(li*phi_incy)).*Hxr;
178 %integration
179 Ey_infr=ifftshift(ifft2((fftshift(Fieldxyr))).*exp(li*PhaseShift));
180 Eyr=Ey_infr((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2));
181
182 %z component E_inf_r
183 Fieldzxr=(cx.*MTz1r.*E_incx.*exp(li*phi_incx)+cy.*MTz2r.*E_incy.*exp(li*phi_incy)).*Hxr;
184 %integration
185 Ez0r=ifftshift(ifft2((fftshift(Fieldzxr))).*exp(li*PhaseShift));
186 Ezr=Ez0r((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2));
187
188 %%%%%%%%
189 Ex(:,:,j)=Exf+Exr; % total field at the material n-2 incident+reflected
190 Ey(:,:,j)=Eyf+Eyr;
191 Ez(:,:,j)=Ezf+Ezr;
192
193 %%%%%%%%
194 %end
195
196 %%%%%%%%
197 %%%transmitted
198 else
199 % x component E_inf_t
200 Fieldxxt=(cx.*MTx1t.*E_incx.*exp(li*phi_incx)+cy.*MTx2t.*E_incy.*exp(li*phi_incy)).*Hzt;
201 %integration
202 Ex_inft=ifftshift(ifft2(fftshift((Fieldxxt))).*exp(li*PhaseShift));
203 Ext=Ex_inft((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2)); %cropped
204

```

```

205 %y component E_inf_t
206 Fieldxyt=(cx.*MTylt.*E_incx.*exp(li*phi_incx)+cy.*MTy2t.*E_incy.*exp(li*phi_incy)).*Hzt;
207 %integration
208 Ey_infT=ifftshift(ifft2((fftshift(Fieldxyt))).*exp(li*PhaseShift));
209 Eyt=Ey_infT((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2)); %cropped
210
211 %z component E_inf_t
212 Fieldxzt=(cx.*MTzlt.*E_incx.*exp(li*phi_incx)+cy.*MTz2t.*E_incy.*exp(li*phi_incy)).*Hzt;
213 %integration
214 Ez0t=ifftshift(ifft2((fftshift(Fieldxzt))).*exp(li*PhaseShift));
215 Ezt=Ez0t((L/2+1-xmax2):(L/2+1+xmax2),(L/2+1-xmax2):(L/2+1+xmax2)); %cropped
216
217 Ex (:,:,j)=Ext; %field beyond the boundary, transmitted
218 Ey (:,:,j)=Eyt;
219 Ez (:,:,j)=Ezt;
220
221 end
222
223 end
224
225
226 %%plot yz cross section through the center at xmax2+1
227 figure(1)
228 EEx=permute(Ex(xmax2+1,:,:),[3 2 1]);
229 EEy=permute(Ey(xmax2+1,:,:),[3 2 1]);
230 EEz=permute(Ez(xmax2+1,:,:),[3 2 1]);
231 I=abs(EEx).^2+abs(EEy).^2+abs(EEz).^2;
232 Imax=max(max(I));
233 In=I/Imax;
234 imagesc((In).^(1/4)); colormap(jet); colorbar

```