Solving the Bessel equation and using the Beam Propagation Method to propagate a Bessel beam in one and two dimensions

Nicco Corduri
Université de Lorraine

December 2022

Abstract

Nowadays, when it comes to experiments using lasers, it is always preferable to be able to perform simulations before launching into real measurements. The simulation of light beam being more and more sophisticated, it allows to obtain results very close to reality and thus to predict the behavior of light during a real experiment. In this study, we are interested in unconventional Bessel beams. First, we compared the Bessel functions obtained in several ways. Then, we used a truncation factor to obtain a Bessel-Gauss function, closer to reality. Finally, we studied the Beam Propagation Method to simulate the propagation of this Bessel function in 1 and 2 dimensions.

Introduction

In the general case of light propagation, the most common form that a light beam describes is a Gaussian function. On the other hand, in other rarer cases, such as the case of non-conventional beams like the Airy or Bessel beams, it is necessary to describe them with functions specific to each of them. In our case, we are interested in the numerical simulation of the propagation of a Bessel beam. For this, we use the Beam Propagation Method on a Bessel-Gauss function in 1 and 2 dimensions. During this study, we used programming in Python language, because this language allows access to many libraries of function useful in our case, namely to solve differential equations, integrals or Fourier transforms.

1 Bessel function

The Bessel equation has the following form, see equation 1.

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 - n^2)y = 0 \tag{1}$$

With n the order of the Bessel function. By solving this equation, we then have access to the initial shape of our beam. However, it is not possible to solve directly a second order differential equation numerically. It is then necessary to separate the equation 1, in two differential equations of order 1, see equation 2.

$$\begin{aligned} v &= \frac{dy}{dx} \\ \frac{dv}{dx} &= \frac{-xv - (x^2 - n^2)y}{x^2} \end{aligned} \tag{2}$$

These equations can now be solved numerically. To do so, we use the « solve_ivp » function of the scipy.integrate library. This function uses the Runge-Kutta method of order 5 to solve these differential equations. Our range of calculation being centered in 0, it is then impossible to solve the equation 2 because it has a division by x. We therefore decide to have a range of calculation included in the interval [1e-5, b] in order not to have this problem. The Bessel function being symmetrical in 0, we invert the function obtained on the positive interval to obtain the part of the Bessel function in the negative calculation interval. The obtained Bessel function is then stored in the form of an array whose number of cells is the number of desired calculation points.

Analytical Bessel functions also exist. These are grouped in two species, the first kind of Bessel functions noted Jn(x) and the second kind of Bessel functions noted Yn(x). In our case, we use the integral form of the first kind Bessel functions, see equation 3, in order to be able to compare it with the Bessel function determined via the Bessel equation previously.

$$J_n(x) = \frac{1}{\pi}\int_0^{\pi} \cos(n\tau - x\sin(\tau))d\tau \tag{3}$$

With n the order of the Bessel function. In order to solve this integral, we use the "quad" function of the scipy.integrate library. Our data are then stored in an array whose number of cells is the number of desired calculation points. To have a last Bessel function for comparison, we use the Bessel function "jv" from the scipy.special library. Comparing the three Bessel functions obtained above, we then obtain the graph in Figure 1.

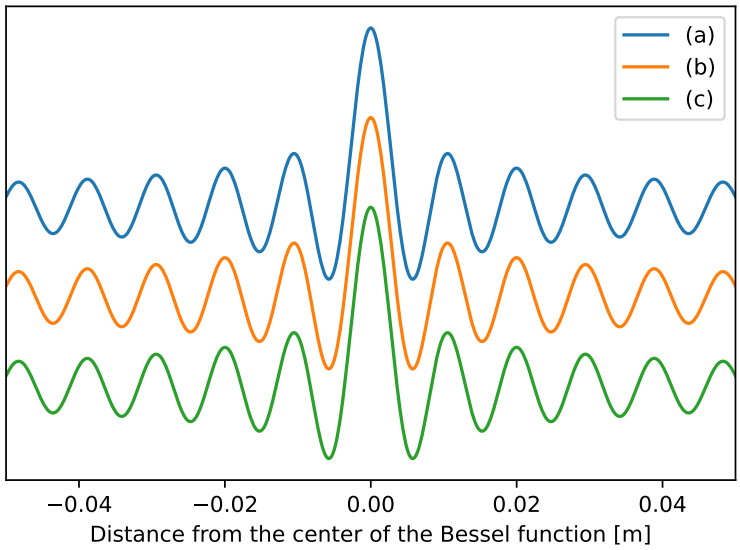


FIGURE 1 – Comparison of Bessel functions obtained in different ways : (a) Bessel function determined via the Bessel equation, (b) Analytical first kind Bessel function in integral form and (c) First kind Bessel function from the python library scipy.special.

With the help of the Figure 1, we can see that our three functions are similar at all points. We can therefore deduce that our Bessel function determined via the Bessel equation is correct. The intensity of this one determined via the equation 4, will serve us as the starting function of our Bessel bea

$$I = |E|^2 \tag{4}$$

With I the intensity of the function and E the amplitude of the function.

In order to determine the width of the main lobe of the Bessel function, we perform a fit on it using a Gaussian function such as the equation 5.

$$f(x) = exp\left(\frac{x^2}{\omega_0^2}\right) \tag{5}$$

This one is centered in 0, has a maximum intensity of 1 and a width determined via the value of ω0. We can therefore determine the width of the main lobe of our Bessel function by determining the Full width at half maximum (FWHM) of the Gaussian fit function. This width is determined using the equation 6.

$$FWHM = 2\sqrt{2\ln 2} * \omega_0 \tag{6}$$

In order to fit this function, we use the "curve_fit" function of the scipy.optimize library. Finally, considering the Bessel function (a) of Figure 1 we obtain Figure 2.

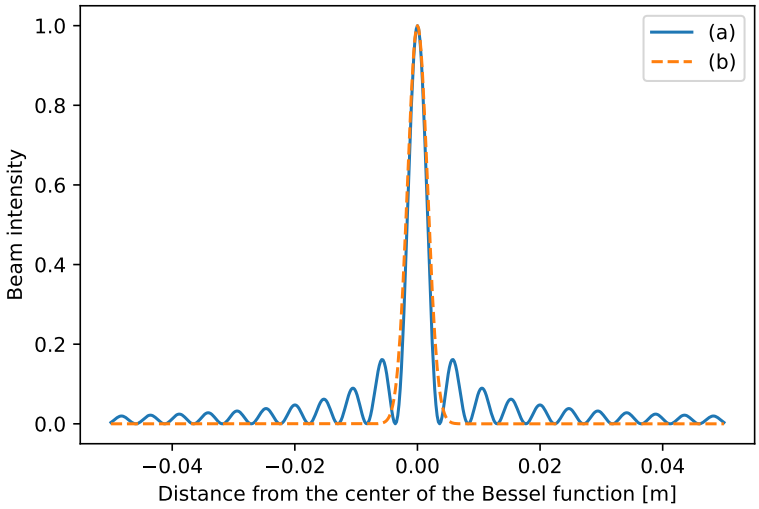


FIGURE 2 – (a) Intensity of the Bessel function and (b) Gaussian function to fit the intensity of the Bessel function.

We can therefore determine using the equation 6 that the Full width at half maximum of the main lobe for the Bessel function in Figure 2 is 5.42e-03 meters.

On the other hand, a Bessel beam with the intensity of Figure 2 is physically impossible, because it would require an infinite energy due to the non-zero intensity of the beam at large distances. This is why we must introduce a truncation factor to limit the intensity of the Bessel beam far from its center. This truncation factor is a Gaussian function such as the equation 5. This new function is called the Bessel-Gauss function [1] [2] [3]. We choose to limit the intensity of our Bessel beam for distances from the center of the beam greater than 20 times the width of the main lobe of the Bessel function. We then obtain the Figure 3. We have our starting Bessel-Gauss function, Figure 3, we must now propagate it numerically.

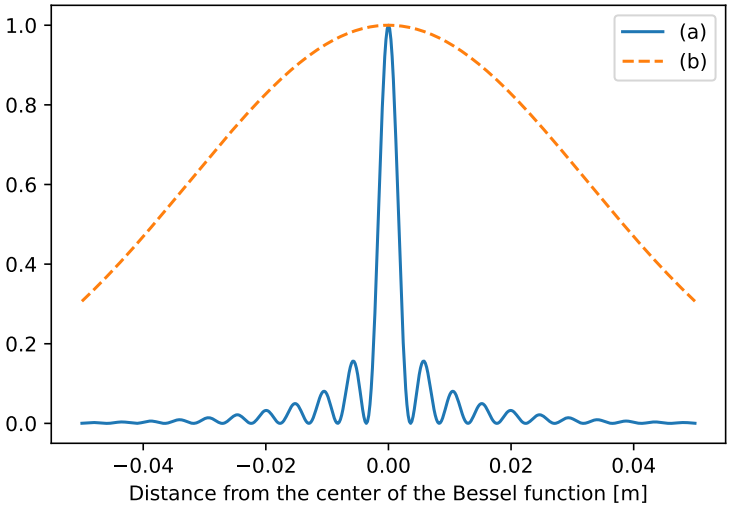


FIGURE 3 – (a) Bessel-Gauss function and (b) Gaussian truncation function of the Bessel function

2 Beam Propagation Method 1D

In order to propagate our initial Bessel-Gauss function, we use the Beam Propagation Method (BPM). This one allows to propagate an electromagnetic wave along a waveguide on a distance very large compared to the wavelength of the beam. This simulation is done by making changes to the propagated beam every dz displacement small compared to the total length of the propagation. The BPM uses Fourier transforms to perform these modifications. In order to use Fourier transforms, we use the function "fft" and the function "ifft" of the library numpy.fft to respectively perform a Fourier transform and an inverse Fourier transform. The first shift of dz/2 must be re-centered using the "fftshift" function of the numpy.fft library because it is possible that curve overlapping problems are created after the first Fourier transform if the basis function approaches a Dirac function. After performing each Fourier transform, we perform a modification of the beam by multiplying it by a function called "Reduced Frequency Matrix (MFR)". This is described by the equation 7.

$$MFR = exp\left(\frac{i}{2k}\left(\frac{2\pi}{L}\right)^2 x^2 dz\right) \tag{7}$$

With k the beam wave vector given by 2*pi/lambda, L the x-axis width, dz the length of each displacement of 0.001*Lr and Lr the Rayleigh length, the distance for which the beam width doubles. Using this reduced frequency matrix function in Fourier space, we can then determine the intensity of our beam after each

displacement of dz such that see equation 8.

$$I(z + dz) = I(z) * MFR \tag{8}$$

Using this propagation method, it is also possible to add a waveguide by simulating lenses after each displacement of dz. We can represent the BPM using Figure 4. The number of computational steps is calculated using the equation 9.

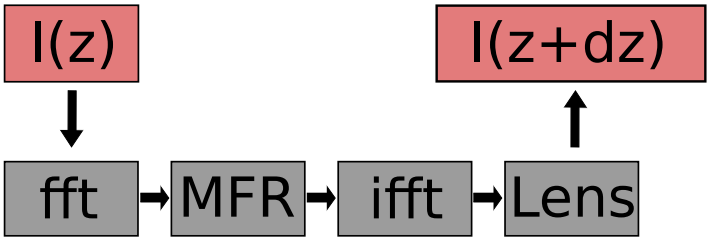


FIGURE 4 – Diagram of the operation of the BPM with I() beam intensity, fft Fourier transform, MFR reduced frequency matrix, ifft inverse Fourier transform and Lens.

$$NbEtape = NbRayleigh/0.001 \tag{9}$$

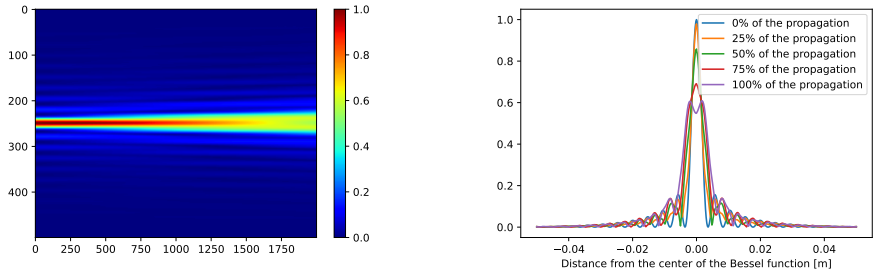
With NbRayleigh, the Rayleigh number being chosen by the user. We can also estimate the distance over which the wave will be propagated by the simulation via the equation 10.

$$D_{simulee} = NbRayleigh * Lr \tag{10}$$

With NbRayleigh, the Rayleigh number and Lr, the Rayleigh width. With such a method of simulation of wave propagation, we can then propagate any starting beam with a Gaussian function, an Airy function or as in our case a Bessel function. The beam data after each displacement will be stored in a matrix whose size is the number of calculation points in one direction and the number of calculation steps in the other.

We can then take the example Figure 5. On this figure, we can see the simulation of the propagation of a Bessel beam in 1+1D using the imshow() function of the library mathplotlib.pyplot with in x-axis the number of calculation steps, in y-axis the width of simulation and in color the intensity of the beam. In this case, we consider that the simulation width is 100e-3 m, the width of the main lobe of the Bessel is 5.4e-3 m, the order of the Bessel is 0, the wavelength is 663e-9 m and that the Rayleigh number is 2, which corresponds to a simulation on 52.50 meters We can first see with the help of Figure 5, that in the first few meters of propagation our beam is very little or not diffracted, which is the

behavior of a Bessel that was reported by Ismail Ouadghiri Idrissi [4], where he explains considering a Bessel beam radius of 2mm, the intensity of the central beam decreases only after 1m. Moreover, we can see that the central intensity of our beam decreases so as to create a hole of intensity in its center, which is well characteristic of the shape that a Bessel of order 0 has after a propagation.



(a) (b)

FIGURE 5 – (a) Propagation of a Bessel beam in 1+1D with in x-axis the number of calculation steps, in y-axis the width of the beam and in color the intensity of the beam and (b) representation of the intensity of the beam at different places of the propagation with in x-axis, the position with respect to the center of the beam and in y-axis the intensity of the beam for positions taken at 0, 25, 50, 75 and 100% of the propagation

3 Beam Propagation Method 2D

The Beam Propagation Method can also be used to simulate the propagation of a 2D beam. Indeed, by gathering the 1D computational data and combining them to have coordinates such as, see equation 11, we can then have the intensity of our starting beam in 2D, as well as the Reduced Frequency Matrix in 2D.

$$R = \sqrt{x^2 + y^2} \tag{11}$$

By using the functions "fft2" and "ifft2" of the numpy.fft library to perform 2D Fourier transforms and the same calculation sequence presented in Figure 4, we can perform 2D beam propagation. Our data will then be stored in the form of a 3-dimensional array whose size is the number of calculation points by the number of calculation points by the number of calculation steps. Such an array can take a lot of storage space so it is not possible to choose too large numbers of computation. By taking the same starting parameters as for Figure 5, we then obtain the 2D results of Figure 6.
With the help of Figure 6, we can see that the evolution of our Bessel beam in 1 dimension, Figure 5 is similar to that obtained in 2 dimensions. We can also see the circles characteristic to the propagation of a Bessel beam.

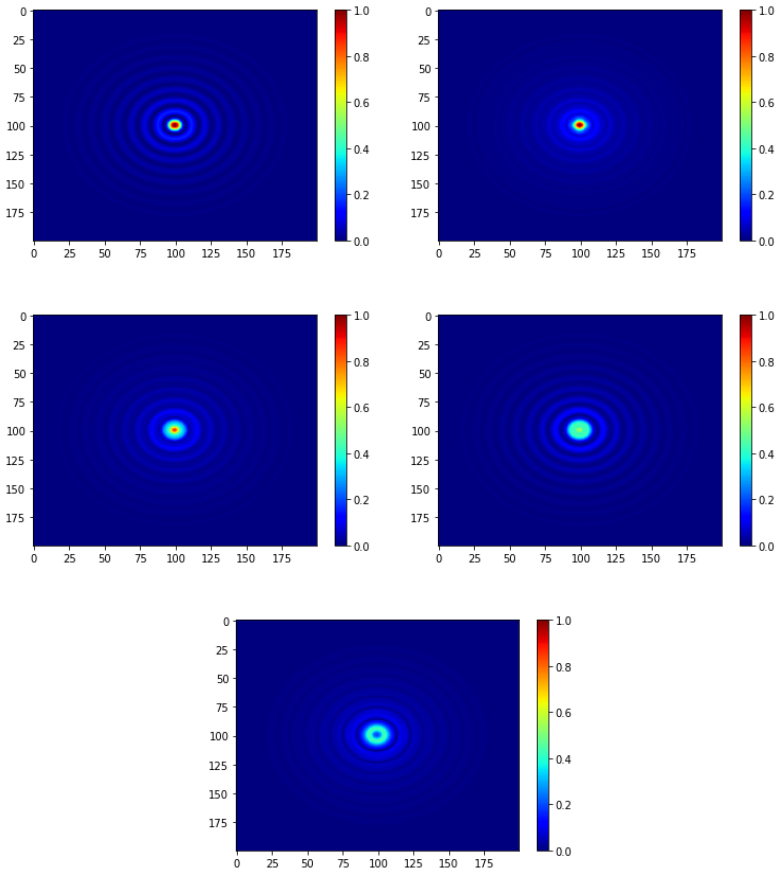


FIGURE 6 – Beam intensity for positions taken at 0, 25, 50, 75 and 100% of a 2D Bessel beam propagation.

Conclusion

The program that we used in this study allows us first to solve the Bessel equation in order to obtain a Bessel function. We checked that the shape of our function was coherent, we compared it with an analytical Bessel function of the first kind in the form of an integral and with a Bessel function already programmed. We were able to measure the width of the main lobe of the Bessel function by performing a fit on it with a Gaussian function. The Bessel function alone being a theoretical function impossible to use in a physically possible beam propagation, we added a truncation factor in order to obtain a Bessel-Gauss function which is physically possible. In a first step, we simulated our one dimensional Bessel-Gauss beam using the Beam Propagation Method. We observed the behavior that we expected for such a beam, namely a very weak

diffraction at short distance and then a diffraction at long distance resulting in a loss of intensity at the center of the beam and an increase in intensity at the edges. Subsequently, we performed this simulation in two dimensions. We observed similar results to those in one dimension, but we could observe the evolution of the circles formed by the Bessel function as the propagation progresses. In order to further complete this study, it would be interesting to observe if the self-healing properties that Bessel beams possess are well respected in the program that we used. Moreover, the use of waveguides would be judicious to study the behavior of this type of beam in guided spaces.

Références

[1] F. Gori, G. Guattari, and C. Padovani. Bessel-gauss beams. *Optics Communications*, 64(6) :491–495, 1987.

[2] Charles G. Durfee, John Gemmer, and Jerome V. Moloney. Phase-only shaping algorithm for gaussian-apodized bessel beams. *Opt. Express*, 21(13) :15777–15786, Jul 2013.

[3] Xiuxiang Chu, Quan Sun, Jing Wang, Pin Lü, Wenke Xie, and Xiaojun Xu. Generating a bessel-gaussian beam for the application in optical engineering. *Scientific Reports*, 5(1), December 2015.

[4] Ismail Ouadghiri Idrissi. *Nonlinear instabilities and filamentation of Bessel beams*. Theses, Université Bourgogne Franche-Comté, December 2018.