Numerical propagation in turbulent media

M. Gil de Oliveira

July 3, 2025

1 Propagation in free space

Although we aim to study the propagation of light in turbulent media, it is important to first understand how light propagates in free space. This will provide us with a solid foundation for understanding the effects of turbulence on light propagation.

1.1 The paraxial wave equation

Our starting point will be the wave equation for the electric field, which is derived from Maxwell's equations. In free space, the wave equation can be expressed as

$$\nabla^2 E - \frac{1}{c^2}\partial_t^2 E = 0, \tag{1}$$

where E is the electric field, which we assumed to have a fixed polarization, c is the speed of light in vacuum, and ∇^2 is the Laplacian operator in three-dimensional space.

A well known class of solutions are the plane waves $E(\mathbf{r}, t) = E_0 e^{i(\mathbf{k}\cdot\mathbf{r}-\omega t)}$ satisfies this equation, where $\mathbf{k}$ is the wave vector, ω is the frequency, and E_0 is a constant amplitude. The norm of the wave vector is related to the frequency ω by $ck = \omega$. These solutions have a well-defined direction of propagation defined by the wave vector $\mathbf{k}$, which we assume to be the z direction, i.e., $\mathbf{k} = k\hat{\mathbf{z}}$.

Nonetheless, this class of solutions is nonphysical, as the electric field is not localized in space and contains an infinite amount of energy. A more physically relevant solution can be obtained by considering a field of the form

$$E(\mathbf{r}, t) = u(\mathbf{r})e^{i(kz-\omega t)}, \tag{2}$$

where $u(\mathbf{r})$ is a complex function describing an envelope that modulates the plane wave. By substituting this expression into the wave equation (1), we obtain

$$\nabla^2 u + 2ik\partial_z u = 0. \tag{3}$$

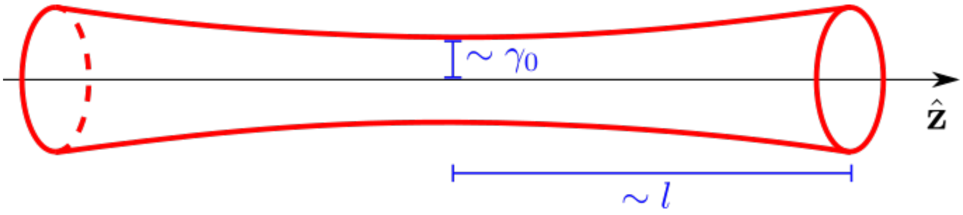

Figure 1: Envelope of a paraxial beam.

The typical envelope of a laser beam is shown in Fig. 1. The envelope $u(\mathbf{r})$ varies slowly along the z direction. More precisely, the variation of u in the z is much smaller than the scale of variation defined by the wavenumber k. Therefore, we will assume that $\partial_z^2 u \ll k\partial_z u$, which allows us to neglect the second derivative with respect to z in the wave equation. This leads us to the paraxial wave equation

$$\nabla_T^2 u + 2ik\partial_z u = 0, \tag{4}$$

where ∇_T^2 is the Laplacian operator restricted to the transverse plane, i.e., $\nabla_T^2 = \partial_x^2 + \partial_y^2$.

1.2 Solution of the paraxial wave equation

The paraxial wave equation (4) is a partial differential equation that describes the propagation of light beams in the paraxial approximation. We want to find a solution for this equation, which will allow us to understand how light propagates in free space. In particular, we want to find a solution for the electric field envelope $u(x, y, z)$ given an initial condition $u(x, y, 0)$ at $z = 0$.

As it the equation is, the most straightforward way to solve it is to use the Fourier transform. Let us then define the Fourier transform of $u(x, y, z)$ as

$$\tilde{u}(k_x, k_y, z) = \int_{\mathbb{R}^2} dxdy\; e^{-i(k_x x + k_y y)} u(x, y, z), \tag{5}$$

which can be inverted as

$$u(x, y, z) = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} dk_x dk_y\; e^{i(k_x x + k_y y)} \tilde{u}(k_x, k_y, z). \tag{6}$$

Substituting this expression into the paraxial wave equation (4), we obtain

$$\left(-k_x^2 - k_y^2 + 2ik\partial_z\right) \tilde{u}(k_x, k_y, z) = 0. \tag{7}$$

This is a simple ordinary differential equation in z for each fixed pair (k_x, k_y). The solution can be written as

$$\tilde{u}(k_x, k_y, z) = \tilde{u}(k_x, k_y, 0) e^{-i(k_x^2 + k_y^2)z/2k}. \tag{8}$$

This provides a step-by-step algorithm for solving the paraxial wave equation:

Algorithm 1 (Angular spectrum method)**.**

1. *Compute the Fourier transform* $\tilde{u}(k_x, k_y, 0)$ *of the initial condition* $u(x, y, 0)$ *using* (5).

2. *For each pair* (k_x, k_y)*, compute the solution* $\tilde{u}(k_x, k_y, z)$ *using the formula* (8).

3. *Finally, compute the inverse Fourier transform to obtain the solution* $u(x, y, z)$.

This is the angular spectrum method, which is a powerful technique for solving the paraxial wave equation. It allows us to compute the solution in the Fourier domain, where the equation becomes much simpler to solve. This is the basis for the method we will use to numerically simulate the propagation of light in turbulent media.

An equivalent way to express the solution of the paraxial equation, that is more simple from the analytical point of view, is obtained by substituting the expression (8) into the inverse Fourier transform (6) and applying the convolution theorem. [1] This leads to the following expression for the solution:

$$u(x,y,z) = \frac{k}{2\pi i z}\int_{\mathbb{R}^2} dx'dy'u(x',y',0)e^{ik\left[(x-x')^2+(y-y')^2\right]/2z}. \tag{9}$$

This is the so-called Fresnel diffraction integral [1, 2].

Exercise 1. *Apply the Fresnel diffraction integral to solve the paraxial wave equation for the initial condition*

$$u(x,y,0) = e^{-(x^2+y^2)/w^2}, \tag{10}$$

where w is a constant that defines the width of the beam (waist).

The Fresnel diffraction formula can be simplified when the distance z is sufficiently large. In fact, assume that our initial condition $u(x,y,0)$ is mostly localized in a disk of radius w around the origin, i.e., $u(x,y,0) \approx 0$ for $x^2+y^2 > w^2$. Then, when $z \gg kw^2/2$, the term $k[(x')^2+(y')^2]/2z$ in the exponent of (9) does not contribute to the integral, and we can approximate the Fresnel diffraction integral as

$$u(x,y,z) \approx \frac{ke^{ik\left(x^2+y^2\right)/2z}}{2\pi i z}\int_{\mathbb{R}^2} dx'dy'u(x',y',0)e^{ik(xx'+yy')/z}. \tag{11}$$

This is known as the Fraunhofer diffraction integral, and it describes the far-field propagation of light. We can then see that, within the Fraunhofer approximation, the propagation of light can be understood as a Fourier transform of the initial condition $u(x,y,0)$, modulated by a quadratic phase factor that depends on the distance z.

1.3 Analogy with quantum mechanics

The paraxial wave equation (4) is analogous to the Schrödinger equation in quantum mechanics

$$i\partial_t\psi(x,y,t) = -\frac{\hbar^2}{2m}\nabla^2\psi(x,y,t), \tag{12}$$

which describes the evolution of a free particle with wave function $\psi(x,y,t)$ in a two-dimensional space. The analogy is particularly useful because it allows us to use the same mathematical techniques to solve both equations. In fact, the angular spectrum method can be seen as a Fourier transform method for solving the Schrödinger equation, where the wave function ψ plays the role of the electric field envelope u.

If we assume that our wave propagates through a medium with a space-dependent refractive index $n(x,y)$, the paraxial wave equation becomes may be approximated as

$$\nabla_T^2 u + 2ik\partial_z u + k^2(n^2-1)u = 0, \tag{13}$$

which is analogous to the Schrödinger equation with a potential, i.e.,

$$i\partial_t\psi(x,y,t) = \left(-\frac{\hbar^2}{2m}\nabla^2 + V(x,y)\right)\psi(x,y,t). \tag{14}$$

This is what happens, for example, inside a waveguide, with the refractive index $n(x,y)$ being responsible for confining the light to a specific region of space, just like a potential well confines a quantum particle.

This analogy can also be extended to understand the imaging properties of lenses [3].

[1] This theorem states that the Fourier transform of convolution is the product of the Fourier transforms.

1.4 Numerical solution

From the analytical point of view, the Fresnel diffraction integral (9) would be the most straightforward way to compute the solution of the paraxial wave equation. However, it is often not practical to compute this integral directly. Instead, we will use a numerical method based on the angular spectrum method, which is often more efficient from a computational point of view.

To simplify the notation, we will assume a single transverse dimension, i.e., we will consider the paraxial wave equation in 1 transverse dimension, with a solution $u(x, z)$. The extension to two transverse is straightforward.

1.4.1 Representing the solution numerically

Although one is used to thinking of the solution $u(x, z)$ as a continuous function, in a computer we will have to represent it as a *discrete* and *finite* set of values. Usually, we represent the initial condition $u(x, 0)$ as an array of values $u_n(0) = u(x_n, 0)$, where $n = 0, \ldots, N-1$, and N is the total number of points in the array. The values of x are then given by $x_n = n\Delta x$, where Δx is the spacing between the points. Any numerical method will then compute (an approximation of) the solution at a later time z as an array of values $u_n(z) = u(x_n, z)$. Then, we can only have access to samples of the solution that lie inside the interval $[0, L]$, where $L = N\Delta x$.

1.4.2 Boundary conditions

We can interpret the restriction to a finite interval $[0, L]$ in two different ways: we either assume that this is just a finite portion of the solution, which exists in the entire real line, or we assume that we are only seeking to solve the paraxial wave equation in the interval $[0, L]$. The first interpretation makes more physical sense for free space propagation, while the second one is more practical for numerical simulations, as we will see. An important difference is that, that, in the second case, we will have to impose boundary conditions at the edges of the interval, as the paraxial wave equation in itself is no longer sufficient to determine the solution uniquely. This is true for any partial differential equation defined in a finite domain.

For example, a wave equation with one spatial dimension may model the propagation of a wave in a string. There are many possible behaviors for a wave pulse when it reaches the end of the string, and each of these behaviors corresponds to a different boundary condition. For example, if the string is fixed at the ends, we impose $u(0, t) = u(L, t) = 0$, which are the so-called Dirichlet boundary conditions. If the string is free at the ends, and we still want to have energy conservation, we impose the Neumann boundary conditions $\partial_x u(0, t) = \partial_x u(L, t) = 0$. These boundary conditions produce different solutions for the wave equation, whose differences are clearly seen in the reflection of a pulse on the edges: for Dirichlet boundary conditions, the reflected wave has a phase shift of π (it gains a minus sign), while, for Neumann boundary conditions, there is no phase shift.

For propagation in free space, the physical solution exists everywhere, and there is no boundary on which to set conditions. Even so, as already said, it is more efficient to find a solution only on the finite domain $[0, L]$, and to assume *periodic* boundary conditions $u(0, z) = u(L, z)$. For our case, this is indeed nonphysical, but, at is turns out, is a good approximation to the physical situation *as far as all the intensity is concentrated far away from the edges of the interval.* We will then always have to keep an eye to this, and increase L if necessary. This is a fair price to pay in order to be able to find an approximate solution efficiently.

1.4.3 First attempt: the Fourier series

As we are working on finite domain $[0, L]$, the Fourier transform (5) is simply not defined. Instead, as we assumed periodic boundary conditions, there is the close concept of a Fourier series

$$u(x,z) = \sum_{m=-\infty}^{\infty} \tilde{u}_m(z) e^{ik_m x}, \quad k_m = \frac{2\pi m}{L}, \tag{15}$$

where the Fourier coefficients $\tilde{u}_m(z)$ are given by

$$\tilde{u}_m(z) = \frac{1}{L} \int_0^L dx \; e^{-ik_m x} u(x,z). \tag{16}$$

This is the equivalent of the Fourier transform for periodic functions, and it is defined in terms of a discrete sum over the wave numbers k_m, instead of continuous integral over wavenumbers. The Fourier coefficients $\tilde{u}_m(z)$ are complex numbers that encode the amplitude and phase of each harmonic component of the solution $u(x,z)$.

We may then proceed just as the continuous case, and substitute the Fourier series (15) into the paraxial wave equation (4). This leads to a simple ordinary differential equation for each Fourier coefficient $\tilde{u}_n(z)$:

$$\left(-k_m^2 + 2ik\partial_z\right) \tilde{u}_m(z) = 0. \tag{17}$$

The solution of this equation is given by

$$\tilde{u}_m(z) = \tilde{u}_m(0) e^{-ik_m^2 z/2k}, \tag{18}$$

where $\tilde{u}_m(0)$ are the Fourier coefficients of the initial condition $u(x,0)$, which can be computed as

$$\tilde{u}_m(0) = \frac{1}{L} \int_0^L dx \; e^{-ik_m x} u(x,0). \tag{19}$$

All of these expressions actually provide exact solutions of the paraxial wave equation with periodic boundary conditions, and are analogous to the angular spectrum method.

Nonetheless, there are still problems that do not allow us to use this method numerically. Note that the hypothesis of a finite domain turned the wave numbers from continuous to discrete, which is definitely a gain. Nonetheless, there are still an infinite amount of them, which, of course, cannot be handled by a computer.

Another problem of this approach is the following: as mentioned, we only have access to the initial condition $u(x,0)$ in a finite set of points, which may be arranged in a vector $u_n(0) = u(x_n, 0)$. In order to compute the Fourier coefficients $\tilde{u}_m(0)$, we could try to approximate it as a Riemann sum

$$\tilde{u}_m(0) \approx \frac{1}{L} \sum_{n=0}^{N-1} e^{-ik_m x_n} u_n(0) \Delta x = \frac{1}{N} \sum_{n=0}^{N-1} e^{-i2\pi mn/N} u_n(0). \tag{20}$$

Note that, in this approximation, the Fourier components will form a periodic sequence: $\tilde{u}_n(0) = \tilde{u}_{n+N}(0)$. Then, the series (15) cannot converge, as the coefficients of the series do not decay to zero. This periodicity makes sense from an "informational point of view", as we give N points of information, and we can only expect to recover N Fourier coefficients.

Although this approach does not work with finite data, it is an important step in the right direction. The approach we present in the next section is motivated by this discussion.

1.4.4 Second attempt: the discrete Fourier transform

Formula (20) (omiting the $1/N$ factor, that is moved elsewhere) is actually the Discrete Fourier Transform (DFT) of the data $u_m(0)$, which is defined for a finite sequence $u_0, \ldots, u_{N-1}$ as

$$\tilde{u}_m = \sum_{m=0}^{N-1} u_n e^{-2\pi imn/N}, \quad m = 0, \ldots, N-1, \tag{21}$$

while its inverse, which reconstructs u_n, is given by

$$u_n = \frac{1}{N} \sum_{m=0}^{N-1} \tilde{u}_m e^{2\pi imn/N}. \tag{22}$$

Now, everything is finite, which makes the DFT a powerful tool for analyzing discrete signals, and it is widely used in signal processing and numerical analysis. The advantage of the DFT is that it can be computed efficiently using the Fast Fourier Transform (FFT) algorithm [4], which has a time complexity of $O(N \log N)$, compared to the $O(N^2)$ time complexity of the naive implementation.

The discussion that follows will be based on the notes [5], which provides a detailed explanation of how to calculate derivatives based on the DFT.

The idea now is to try to write our solution $u(x,z)$ using only a finite number of Fourier coefficients:

$$u(x,z) = \frac{1}{N} \sum_{m=0}^{N-1} \tilde{u}_m(z) e^{2\pi imx/L}. \tag{23}$$

Note that, as we have access to $u_n(0) = u(x_n, 0)$, we can make this function agree with the initial condition at the points x_n by computing the $\tilde{u}_m(0)$ using the DFT (21). This means that we are interpolating the solution $u(x,z)$ at the points x_n using a finite number of Fourier coefficients $\tilde{u}_m(z)$.

Now we make a subtle but extremely important observation: (21) is not the only interpolation formula that we can use. In fact, we can write the solution as

$$u(x,z) = \frac{1}{N} \sum_{m=0}^{N-1} \tilde{u}_m(z) e^{2\pi i(m+s_m N)x/L}, \tag{24}$$

where s_m is an integer that can be chosen arbitrarily. This will give exactly the same result as (23) (which corresponds to $s_m = 0$), at the sample points x_n, but will change the value of the function in between. This is because the exponential terms are simply completing a different number of oscillations between the sample points. Although the behavior at the sample points is the same, the interpolation between them is different, and this will lead to different results for the solution $u(x,z)$.

The question is then how to choose the integers s_m. The idea is, in the absence of any other information, to make the choice that minimizes the oscillations of the solution $u(x,z)$ between the sample points [5]. As the frequency of the oscillations is given by the term $2\pi(m + s_m N)/L$, we want to minimize $|m + s_m N|$ for each m, this is achieved by choosing $s_m = 0$ for $m \leq N/2$ and $s_m = -1$ for $m > N/2$. As an example, for $N = 5$, the corresponding wavenumbers should be $(0, 1, 2, -2, -1) \times \frac{2\pi}{L}$. Fortunately, most numerical libraries that implement the DFT also implement a function called fftfreq, which computes the frequencies of the Fourier coefficients in this way. This is simply a reordering of part of the sequence $(-2, -1, 0, 1, 2) \times \frac{2\pi}{L}$, which would correspond to defining the DFT (21) with a symmetric sum instead. As this is not the convention, one uses the fftfreq instead.

With this discussion in mind, we can finally write our numerical approximation to the solution of the paraxial wave equation as follows:

Algorithm 2 (Numerical angular spectrum method)**.**

1. *Compute the DFT* $\tilde{u}_m(0)$ *of the initial condition of the initial condition* $u_n(0) = u(x_n, 0)$ *at the sample points* $x_n = n\Delta x$*, where* $n = 0, \ldots, N-1$*.*

2. *For each* m*, compute the solution* $\tilde{u}_m(z)$ *using the formula* (18)*. Remember to obtain the wavenumbers* k_m *using the fftfreq function, which will automatically take care of the reordering of the frequencies.*

3. *Finally, compute the inverse DFT to obtain the solution* $u_n(z) = u(x_n, z)$*.*

2 Propagation in turbulent media

References

[1] J.W. Goodman. *Introduction to Fourier Optics.* W. H. Freeman, 2017.

[2] J.D. Schmidt. *Numerical Simulation of Optical Wave Propagation with Examples in MATLAB*. Press monograph. SPIE, 2010.

[3] David Stoler. Operator methods in physical optics. *JOSA*, 71(3):334–341, 1981.

[4] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[5] Steven G. Johnson. Notes on fft-based differentiation. https://math.mit.edu/~stevenj/fft-deriv.pdf, 2011.