A TRIDENT SCHOLAR PROJECT REPORT

NO. 520

An Analytical and Computational Study of the Paraxial Wave Equation with Applications to Laser Beam Propagation

by

Midshipman 1/C Kyle G. Jung, USN

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

USNA-1531-2

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
5-16-22		

4. TITLE AND SUBTITLE

An Analytical and Computational Study of the Paraxial Wave Equation with Applications to Laser Beam Propagation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Kyle G. Jung

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Naval Academy
Annapolis, MD 21402

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Trident Scholar Report no. 520 (2022)

12. DISTRIBUTION / AVAILABILITY STATEMENT

This document has been approved for public release; its distribution is UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this project, we approximate solutions to the Paraxial Wave Equation by posing an initial boundary value problem (IBVP). The Paraxial Wave Equation is a model of laser beam propagation. A variable refractive index term is introduced within this partial differential equation to account for a nonhomogeneous medium. We apply Spectral methods to approximate the transverse Laplacian operator and an adaptive Runge-Kutta method using MATLAB's ordinary differential equation solvers to propagate the beam forward in space. Three Spectral methods are considered: a Fourier Galerkin method, a Fourier collocation method, and a Chebyshev collocation method. These methods are verified in two ways: (1) by comparing the numerical IBVP solution to the exact solution in unbounded space for a Gaussian beam propagating in homogeneous media and (2) by applying the method of manufactured solutions. We apply a Fourier collocation method to model laser beam propagation through a nonhomogeneous medium.

15. SUBJECT TERMS
Paraxial Wave Equation, Laser Beam Propagation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**		57	**19b. TELEPHONE NUMBER** *(include area code)*

U.S.N.A. --- Trident Scholar project report; no. 520 (2022)

AN ANALYTICAL AND COMPUTATIONAL STUDY OF THE PARAXIAL WAVE EQUATION WITH APPLICATIONS TO LASER BEAM PROPAGATION

by

Midshipman 1/C Kyle G. Jung
United States Naval Academy
Annapolis, Maryland

_____
(signature)

Certification of Advisers Approval

Professor Reza Malek-Madani
Mathematics Department

_____
(signature)

_____
(date)

Professor Svetlana Avramov-Zamurovic
Weapons, Robotics, and Control Engineering Department

_____
(signature)

_____
(date)

Acceptance for the Trident Scholar Committee

Professor Maria J. Schroeder
Associate Director of Midshipman Research

_____
(signature)

_____
(date)

USNA-1531-2

Contents

1 Abstract

In this project, we approximate solutions to the Paraxial Wave Equation by posing an initial boundary value problem (IBVP). The Paraxial Wave Equation is a model of laser beam propagation. A variable refractive index term is introduced within this partial differential equation to account for a nonhomogeneous medium. We apply Spectral methods to approximate the transverse Laplacian operator and an adaptive Runge-Kutta method using MATLAB's ordinary differential equation solvers to propagate the beam forward in space. Three Spectral methods are considered: a Fourier Galerkin method, a Fourier collocation method, and a Chebyshev collocation method. These methods are verified in two ways: (1) by comparing the numerical IBVP solution to the exact solution in unbounded space for a Gaussian beam propagating in homogeneous media and (2) by applying the method of manufactured solutions. We apply a Fourier collocation method to model laser beam propagation through a nonhomogeneous medium.

2 Introduction

Lasers can be found in a broad range of scientific, industrial, and military applications. Their applications in directed energy weapons and communication systems are of particular interest to the United States Navy [17], [5]. However, the Navy's operational domain within the maritime environment poses various challenges to laser beam propagation. The maritime environment encompasses the atmosphere directly above and the water below the surfaces of oceans and their littoral regions. Variations in environmental parameters such as temperature, fluid density, pressure, and scattering particle distributions result in various nonlinear effects which distort propagating laser beams in the maritime environment. The dynamical nature of the changes within the maritime domain exacerbates the distortions to the phase and amplitude of laser beam propagation. The full spectrum of these effects fall beyond the scope of this paper. Various fields such as oceanography and engineering independently study these effects based on the fluid dynamics of the region. Extensive study and experimentation have been done within the field of optics to study the effects of the maritime environment on laser beam propagation [10].

The interaction of light and the medium could be characterized by the spatial and temporal changes in the refractive index within the medium, which is called optical turbulence. The refractive index is a single parameter that models nonhomogeneity in a medium, which accounts for effects such as temperature, salinity, and pressure variations that are distributed unevenly in a medium. We note that refractive index changes do not necessarily apply to media with particulates that causes significant scattering or account for the effects of attenuation.

This paper mainly focuses on constructing a numerical framework to simulate laser beam propagation by finding approximate solutions to the Paraxial Wave Equation(PWE). The PWE is a model of laser beam propagation in optical turbulence or vacuum [1]. A variable refractive index coefficient is introduced within this complex-valued partial differential equation to account for a nonhomogeneous medium. In our study, we consider a varying refractive index converted from temperature fluctuations from a numerical study of natural convection in a Rayleigh-Benard volume [10].

We apply Spectral methods to find solutions to the PWE. Spectral methods are a broad class of numerical methods based on approximating functions using their series expansions. By finding the derivatives of their series expansion, the derivative of the function can be approximated. Spectral methods are *global* methods in that the derivative of a function at a single point is determined by the values of the function at all of its surrounding points. The resulting benefits are high spatial resolution relative to computational expenditure and phase stability over long time integration [9]. High spatial resolution may be conducive for approximating the behavior of wavelengths of light more accurately. Phase stability over long time integration allows us to study laser beams propagating over long distances. These benefits are advantageous to simulating and understanding laser beam propagation.

We pose the laser beam propagation as an initial-boundary value problem

(IBVP) for the PWE. The numerical scheme in this paper applies a Spectral method to model the transverse spread of the beam and an adaptive Runge-Kutta Method through MATLAB's ODE45 solver [14] to propagate the beam forward. The accuracy of the Spectral method is verified in two ways: (1) by comparing the numerical IBVP solutions to analytical solutions in unbounded space for laser propagation in a homogeneous medium and (2) by applying the Method of Manufactured Solutions.

3 The Paraxial Wave Equation

The partial differential equation of interest is the Paraxial Wave Equation, a model of laser beam propagation. Laser beams are a special form of monochromatic light whose equation can be traced directly from Maxwell's equations. The governing equations of electromagnetic fields reduce to the Paraxial Wave Equation as shown in Appendix A for the complex-valued function u, where u is any component of the electric or magnetic field. Time-dependence is separated leaving a function $u(x, y, z)$ of three spatial variables,

$$\Delta_\perp u + 2ik\frac{\partial u}{\partial z} + k^2(\frac{n_r^2}{n_0^2} - 1)u = 0, \tag{1}$$

where $\Delta_\perp u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ is the transverse Laplacian operator, $i = \sqrt{-1}$, and $k = \frac{2\pi}{\lambda}$ is the wave number with the light's wavelength λ. The spatially varying refractive index $n_r(x, y, z)$ and the reference refractive index constant n_0 characterize the laser beam's medium of propagation.

Spatially, the laser beam propagates along the z-axis where z=0 is the location of the beam leaving the laser aperture. The beam spreads transversely in the x, y directions, presenting the orientation of the laser beam seen when it hits a surface like a wall.

The square modulus and argument of the complex-valued solution u give the intensity I and phase Φ of the laser beam,

$$I = |u|^2, \tag{2}$$

and

$$\Phi = \tan^{-1}\left(\frac{\mathrm{Im}(u)}{\mathrm{Re}(u)}\right). \tag{3}$$

For laser beams propagating in a homogeneous medium, we let $n_r = n_0$, resulting in the well-studied form of the PWE,

$$\Delta_\perp u + 2ik\frac{\partial u}{\partial z} = 0. \tag{4}$$

3.1 Gaussian Beam Solution

Exact solutions for various beam forms have been extensively studied for the homogeneous PWE in (4). This section follows the work of Andrews and Phillips [1]. We can model a laser beam by posing an initial-value problem for the unbounded half-space $z > 0$. The initial conditions model a beam leaving the laser's aperture. The simplest beam is the Gaussian beam. As denoted by its name, the initial condition for it uses a modified Gaussian function to model a spherical wave front,

$$u(r, 0) = a_0 \exp(-\frac{1}{2}\alpha_0 k r^2), \tag{5}$$

where $r^2 = x^2 + y^2$, $\alpha_0 = \frac{2}{kw_0^2} + i\frac{1}{F_0}$, and a_0 is the initial amplitude. The beam's input parameters λ, F_0, and w_0 are the wavelength, initial radius of curvature, and initial beam radius in meters. Typical values are $\lambda = 633E-9$ [m], $F_0 = 500$ [m], and $w_0 = 0.03$ [m]. The initial amplitude is set to $a_0 = 1$.

The homogeneous PWE in (4) supplemented with the Gaussian initial conditions in (5) poses an initial-value problem for the Gaussian beam. The exact solution of the initial value problem can be found by assuming the solution is of the form,

$$u(r, z) = A(z) \exp(\frac{1}{p(z)} \frac{\alpha_0 k r^2}{2}), \tag{6}$$

where $p(z)$ and $A(z)$ are two unknown functions. We solve for these unknown functions by applying our initial condition in (5), recognizing $p(0) = 0$ and $A(0) = a_0 = 1$. Then, we substitute the template in (6) into the homogeneous PWE, separate terms, and solve the resulting ordinary differential equation for $p(z)$ and $A(z)$. More details for this direct solution method can be found in Andrews and Phillips [1]. Gbur [6] derives the higher order Laguerre-Gaussian beam using the same method.

Alternatively, our initial-value problem can be solved by identifying the Green's function for the Paraxial Wave Equation and evaluating the convolution of the Green's function with the Gaussian beam's initial condition. This integral is also referred to as the Huygen-Fresnel integral [1]. This method can be generalized to more complex beam forms by changing the problem's initial conditions.

Both the direct solution method and Green's function method result in the exact solution for the Gaussian beam in the unbounded half space $z > 0$,

$$u(r, z) = \frac{1}{1 + i\alpha_0 z} \exp(\frac{ik}{2z}(\frac{i\alpha_0 z}{1 + i\alpha_0 z})r^2). \tag{7}$$

The Gaussian beam solution can be rewritten in terms of the beam parameters,

$$\Theta_0 = 1 - \frac{z}{F_0}, \qquad \Lambda_0 = \frac{2z}{kw_0^2}.$$

Then, the exact solution becomes

$$u(x, y, z) = \frac{1}{\sqrt{\Theta_0^2 + \Lambda_0^2}} \exp(-\frac{x^2 + y^2}{W(z)^2}) \exp[i(-\varphi - \frac{k(x^2 + y^2)}{2F(z)})], \tag{8}$$

where

$$\varphi = \tan^{-1}(\frac{\Lambda_0}{\Theta_0}), \qquad W(z) = w_0\sqrt{\Theta_0^2 + \Lambda_0^2}, \qquad F(z) = \frac{F_0(\Theta_0^2 + \Lambda_0^2)(\Theta_0 - 1)}{\Theta_0^2 + \Lambda_0^2 - \Theta_0}.$$

Figures 1, 2, 3 show the intensity cross-section of a Gaussian beam at three distances down its propagation path $z = 0, 500, 1200$[m]. The intensity peaks at the center where $(x, y) = (0, 0)$ and decays in the famous bell-curve shape as one moves radially away from it. With the initial radius of curvature $F_0 > 0$, the beam starts off converging until it reaches its collimation distance where the beam begins to diverge. Figure 2 is a plot of the beam at its collimation point where the beam is at its smallest radius and highest intensity.

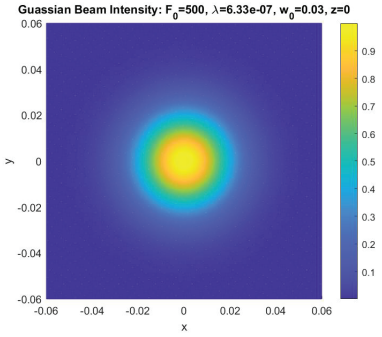

Figure 1: Gaussian Beam at z=0[m]

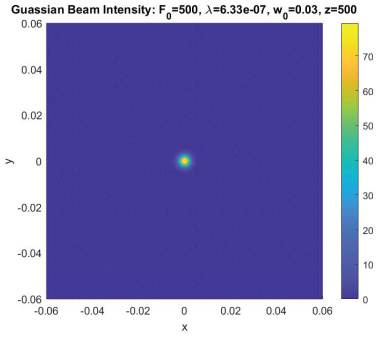

Figure 2: Gaussian Beam at z=500[m]

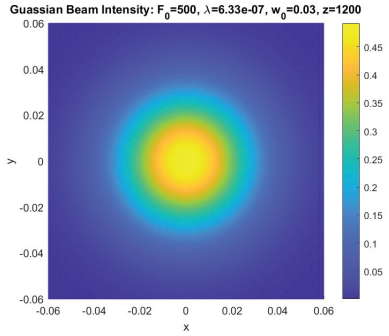

Figure 3: Gaussian Beam at z=1200[m]

Along the propagation axis, the intensity increases until it peaks at the F_0 distance along the propagation path. Figure 4 shows this intensity relation along the propagation path z at the beam center.

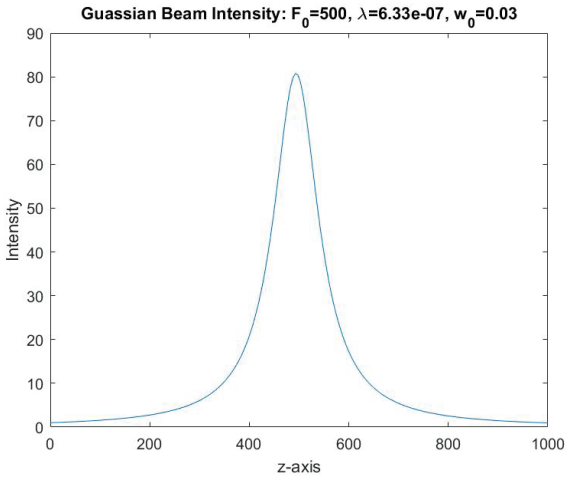

Figure 4: Gaussian beam intensity at the center of the beam where $(x, y) = (0, 0)$ and along the propagation axis from $0 < z < 1000$ meters where $F_0 = 500$ meters, $w_0 = 0.03$ meters, and $\lambda = 6.33 \times 10^{-7}$ meters. Intensity can be measured in milliwatts per meter squared.

We will use the exact solution of the Gaussian beam in unbounded space to compare with our numerical solution in bounded space and calibrate our Spectral method scheme.

3.2 Non-dimensional PWE System

In our study, we convert Equation (1) to a real-valued system of equations where x, y, z are now non-dimensional variables and $\bar{x}, \bar{y}, \bar{z}$ denote physical distances in meters. We let

$$x = \frac{\bar{x}}{w_0}, \qquad y = \frac{\bar{y}}{w_0}, \qquad z = \frac{\bar{z}}{F_0}. \tag{9}$$

Then, the PWE can be rewritten in terms of non-dimensional variables,

$$\Delta_\perp u + \frac{2ikw_0^2}{F_0}\frac{\partial u}{\partial z} + w_0^2 k^2 (\frac{n_r^2}{n_0^2} - 1)u = 0. \tag{10}$$

We note that $u = U + iV$ is a complex-valued function. Substituting $U + iV$ into Equation (1), we obtain a real-valued, coupled system of PDEs where U and V satisfy

$$\begin{cases} \Delta_\perp U - \alpha \frac{\partial V}{\partial z} + \beta U = 0, \\ \Delta_\perp V + \alpha \frac{\partial U}{\partial z} + \beta V = 0, \end{cases} \tag{11}$$

for $\alpha = \frac{2kw_0^2}{F_0}$ and $\beta(x, y, z) = w_0^2 k^2 (\frac{n_r^2}{n_0^2} - 1)$.

We begin our numerical study by first concentrating on propagation in a homogeneous medium, where $n(x, y, z) = n_0$, where equation (11) reduces to

$$\begin{cases} \Delta_\perp U - \alpha \frac{\partial V}{\partial z} = 0, \\ \Delta_\perp V + \alpha \frac{\partial U}{\partial z} = 0. \end{cases} \tag{12}$$

4 Spectral Methods

A differential equation forms a relationship between a function f and its derivatives. We say f satisfies the differential equation if the equality holds true for all values of the independent variables. For example, let us consider the first order ordinary differential equation (ODE), $f' + f = 0$. A solution to this differential equation is $f = e^{-x}$, because $f' + f = -e^{-x} + e^{-x} = 0$. In this case, we know the exact function that satisfies the differential equation. A wide range of analytical techniques can be applied to various differential equations to find exact solutions. However for many differential equations, few techniques are available to find exact solutions. The list of analytical techniques decrease as we get into the class of partial differential equations (PDEs), which describe differential equations with functions of more than one variable. Numerical methods are the dominant approach to finding solutions to PDEs and certain classes of ODEs.

4.1 Series Expansions

Spectral methods are one class of numerical methods that approximate the solution to a differential equation as a series expansion. We can approximate any function f as linear combinations of simpler basis functions ϕ_n weighted with expansion coefficients a_n,

$$f(x) = \sum_{n=0}^{\infty} a_n \phi_n(x). \tag{13}$$

A popular series expansion comes in the form of the Maclaurin series where the coefficients are explicitly defined by $a_n = \frac{f^{(n)}(0)}{n!}$ and the basis functions are the set of polynomials, $\phi = \{x^n | n \in \mathbb{Z}^+\}$. In contrast, Spectral methods make use of an orthogonality relation for a set of functions to compute the expansion coefficients.

Typical basis functions for Spectral methods come in the form of trigonometric functions and orthogonal polynomials. This paper presents a Fourier expansion and a Chebyshev expansion. These basis functions form a set of special functions that belong to a Hilbert space, a special form of an inner product space. An inner product space is a normed vector space with a defined inner product.

Definition 4.1 (Inner Product)**.** Let H be a vector space on $\mathbb{C}$. An inner product is a function $\langle \cdot, \cdot \rangle$ from $\mathbb{C}^2$ to $\mathbb{C}$ such that for all $f, g, h \in H$ and $c \in \mathbb{C}$,

1. $\langle f, g\rangle = \overline{\langle f, g\rangle}$,
2. $\langle f + h, g\rangle = \langle f, g\rangle + \langle h, g\rangle$,
3. $\langle cf, g\rangle$=c $\langle f, g\rangle$
4. $\langle f, f\rangle \geq 0$, $\forall f \in H$ and $\langle f, f\rangle = 0$ if and only if $f = 0$.

A Hilbert space is a complete inner product space.

Definition 4.2 (Hilbert space)**.** An inner product space H is said to be a Hilbert space if it is complete, meaning every Cauchy sequence in H converges to a function in H. That is, given a sequence $\{x_n\}_{n\geq 1} \in H$, if $||x_n - x_m|| \to 0$ as $n, m \to \infty$, then $\exists x \in H$ such that $||x - x_n|| \to 0$ as $n \to \infty$.

Definition 4.3 (Norm)**.** A **norm** $||\cdot||$ is a function from a vector space onto $\mathbb{R}^+$ that satisfies

1. $||f|| = 0$ iff $f = 0$,
2. $||cf|| = |c|\,||f||$ for $c \in \mathbb{R}$ and $f \in H$,
3. $||f + g|| \leq ||f|| + ||g||$, $\forall f, g \in H$.

Norms are measures of distances within a space. They will be extensively used later on to study error.

In this paper, we utilize the following inner product and the associated L^2 norm,

$$\langle f, g\rangle = \int_\Omega f(x)\overline{g(x)}dx. \tag{14}$$

$$||f|| = \sqrt{\int_\Omega |f(x)|^2 dx}. \tag{15}$$

It can be shown that (14) and (15) satisfy the properties of an inner product and a norm on a Hilbert space.

4.2 Orthogonality

Orthogonality is a critical property for the selection of basis functions in Spectral methods.

Definition 4.4 (Orthogonality)**.** For some vector space V, two functions $f, g \in V$ are said to be orthogonal if $\langle f, g\rangle = 0$. The set S is said to be orthogonal if $\langle f, g\rangle = 0$ holds true $\forall f, g \in S$.

Spectral methods rely on orthogonal sets of functions to define the expansion coefficients a_n for a series expansion approximation. To find the expansion

coefficient, we take the inner product of Equation (13) with a fixed but arbitrary basis function ϕ_m for $m \in \mathbb{Z}^+$,

$$\langle f, \phi_m \rangle = \sum_{n=0}^{\infty} \langle a_n \phi_n, \phi_m \rangle. \tag{16}$$

Since $\langle \phi_m, \phi_n \rangle = 0$ for $n \neq m$, equation (13) reduces to

$$\langle f, \phi_m \rangle = a_m \langle \phi_m, \phi_m \rangle, \tag{17}$$

which can be rewritten in terms of a_m,

$$a_m = \frac{1}{\langle \phi_m, \phi_m \rangle} \langle f, \phi_m \rangle = \frac{1}{\int_\Omega |\phi_m(x)|^2 dx} \int_\Omega f(x) \overline{\phi_m(x)} dx. \tag{18}$$

For example, let us define the complex Fourier series for some function $g(x)$ on the interval $\Omega = [0, 2\pi]$ and some expansion coefficient $b_n \in \mathbb{C}$,

$$g(x) = \sum_{n=-\infty}^{\infty} b_n e^{inx}. \tag{19}$$

From (18), the expansion coefficients are computed through the integral,

$$b_n = \frac{1}{\int_0^{2\pi} (e^{inx} e^{-inx}) dx} \int_0^{2\pi} f(x) e^{-inx} dx = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-inx} dx. \tag{20}$$

Sets of functions are also orthogonal with respect to some weight function w. For example, the set of Chebyshev polynomials are orthogonal with respect to $w = \frac{1}{\sqrt{1-x^2}}$. Let us consider the Chebyshev series expansion on the interval $[-1, 1]$,

$$h(x) = \sum_{n=-\infty}^{\infty} c_n T_n(x), \tag{21}$$

where T_n is the n^{th} Chebyshev polynomial. Chebyshev polynomials can be defined by the recurrence relation,

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1} = 2xT_n(x) - T_{n-1}(x). \tag{22}$$

It can be shown for any $n, m \in \mathbb{Z}^+$ such that $n \neq m$,

$$\int_{-1}^{1} T_n(x) T_m(x) \frac{1}{\sqrt{1-x^2}} dx = 0. \tag{23}$$

It follows

$$c_n = \frac{1}{\int_{-1}^{1} |T_n(x)|^2 w(x) dx} \int_{-1}^{1} T_n(x) h(x) w(x) dx. \tag{24}$$

The treatment of computing the integrals in (18) is one distinct difference in the implementation of the Galerkin and collocation methods, two forms of Spectral methods which we will describe later in this paper.

4.3 Time Dependent PDEs

We consider time-dependent PDEs formulated as an initial boundary value problem on the domain $\Omega \subseteq \mathbb{R}$ with the differential and boundary operator, $\mathcal{L}$ and $\mathcal{B}$,

$$\begin{aligned} \frac{\partial u}{\partial t} &= \mathcal{L}[u], \qquad x \in \Omega, \\ \mathcal{B}u(x,t) &= 0, \qquad x \in \partial\Omega, \\ u(x,0) &= 0, \qquad x \in \Omega. \end{aligned} \tag{25}$$

We assume the solution u belongs to a Hilbert space and can be written as a linear combination of basis functions,

$$u(x,t) = \sum_{n=1}^{\infty} a_n(t)\phi_n(x). \tag{26}$$

The approximation $u_N(x,t)$ of $u(x,t)$ can be found by truncating the infinite series expansion of u to N number of terms.

4.3.1 Spectral Galerkin Method

The Galerkin method requires a residual R_N to equal 0, where the residual can be defined as the difference between the function and the approximation,

$$R_N = u - u_N. \tag{27}$$

Since u is a solution to the time-dependent PDE in (25), then the residual can be defined by

$$R_N = \frac{\partial u_N}{\partial t} - \mathcal{L}[u_N]. \tag{28}$$

To force the residual to be 0, we take the inner product of R_N with a test function ψ. In the Galerkin method, we choose test functions within the set of basis functions for our series expansion approximation defined in equation (26), so we choose $\psi = \phi$. We take the inner product of R_N with test functions:

$$\langle R_N, \phi_n \rangle = 0. \tag{29}$$

The orthogonality property of our set of basis functions reduces the PDE in (29) to an ODE comprised of the n^{th} coefficient function, $a_n(t)$. The coefficients for the series approximation are returned by solving the ODEs for each coefficient $\{a_0(t), a_1(t), a_2(t), \ldots a_N(t)\}$.

Example:

Let us consider the standard wave equation with an added variable coefficient term $f(x)$ in the following IBVP,

$$u_{tt} - c^2 u_{xx} + f(x)u = 0, \tag{30}$$

with Dirichlet boundary conditions and initial conditions

$$\begin{cases} u(0,t) = u(L,t) = 0, \\ u(x,0) = u_0(x), \quad u_t(x,0) = u_1(x), \end{cases}$$

in a domain of $0 \leq x \leq L$ for some functions $f(x)$, $u_0(x)$, and $u_1(x)$.

To apply a Galerkin method, we select a truncated sine series to approximate our solution function u,

$$u_N(x,t) = \sum_{n=0}^{N} a_n(t) \frac{2}{L} \sin(\frac{n\pi x}{L}). \tag{31}$$

The set of basis functions is defined by

$$\{\phi(x)\} = \left\{ \frac{2}{L} \sin(\frac{n\pi x}{L}) \quad | \quad n = 1, 2, 3, \ldots, N \right\}. \tag{32}$$

Each pair of functions in the set is orthogonal since for any $n, m \in \mathbb{N}$ such that $n \neq m$,

$$\langle \phi_n, \phi_m \rangle = \int_0^L [\frac{2}{L} \sin(\frac{n\pi x}{L})][\frac{2}{L} \sin(\frac{m\pi x}{L})] dx = 0. \tag{33}$$

Thus, the set of functions creates an orthogonal basis that spans the finite dimensional space,

$$S_N = \text{span}\{\frac{2}{L} \sin(\frac{n\pi x}{L}) \quad | \quad n = 1, 2, 3, \ldots, N\}. \tag{34}$$

Additionally, there exist an eigenfunction relationship for each function in our set of basis functions,

$$\mathcal{L}[\phi_n] = \frac{n^2\pi^2}{L^2} [\frac{2}{L} \sin(\frac{n\pi x}{L})] = \lambda_n \phi_n,$$

for $\mathcal{L}[u] = -\frac{\partial^2 u}{\partial x^2}$.

Returning to solving our problem in equation (30), we substitute the series expansion in equation (31) into the PDE in equation (30):

$$\sum_{n=1}^{N} [a_n''(t)\phi_n(x) + c^2 \lambda_n a_n(t) \phi_n(x) + f(x) a_n(t) \phi_n(x)] = 0. \tag{35}$$

Next, we take the inner product as defined in equation (14) with fixed test functions ψ_m for $m = 1, 2, 3, \ldots N$. As mentioned previously, the test and basis functions are chosen to be the same in the Galerkin method. Because the set of basis and test functions $\{\phi\}$ are chosen to be orthonormal, we have the relationship,

$$\langle \phi_n, \phi_m \rangle = \delta_{m,n} = \begin{cases} 0 & \text{if } n \neq m, \\ 1 & \text{if } n = m, \end{cases} \tag{36}$$

where δ is the Kronecker delta function.

Simplifying equation (35) using the orthogonality and eigenfunction relationship, we get

$$a_m''\langle\phi_m,\phi_m\rangle + c^2\lambda_m a_m\langle\phi_m,\phi_m\rangle + \sum_{n=1}^{N} a_n\langle f(x)\phi_n,\phi_m\rangle = 0. \tag{37}$$

Notice the indices change because all nonzero sums occur only when $n = m$. Equation (37) further simplifies to

$$a_m'' + c^2\lambda_m a_m + \sum_{n=1}^{N} a_n\langle f(x)\phi_n,\phi_m\rangle = 0, \qquad m = 1, 2, ... \tag{38}$$

Now the problem becomes solving the N number of ordinary differential equations (ODEs) supplemented with initial conditions from the PDE,

$$\begin{cases} a_m(0) = \langle u_0(x),\phi_m(x)\rangle, \\ a_m'(0) = \langle u_1(x),\phi_m(x)\rangle. \end{cases} \tag{39}$$

Solving these ODEs produces the $a_m(t)$ coefficient values. This leaves N number of second order ODEs to solve. In our example, $a_m(t)$ coefficients are solved using MATLAB's ODE45 solver. Using ODE45 requires converting the second order ODEs to first order ODEs, doubling the equation count to $2N$. The approximate solution to the IBVP can now be reconstructed using the series expansion given in equation (31).

Presented in Figure 5 is the solution to the IBVP with the specially defined function values,

$$f(x) = \sin(x) - 5\sin(30x) + 10\sin(40x),$$

$$u_0 = e^{-200(x-0.5)^2}, \quad u_1 = -400(x-0.5)e^{-200(x-0.5)^2}.$$

Small amplitude oscillations can be seen within the plot as the initial Gaussian shape moves and reflects off the left boundary. These oscillations are a result of the variable coefficient term $f(x)$ in the modified wave equation.

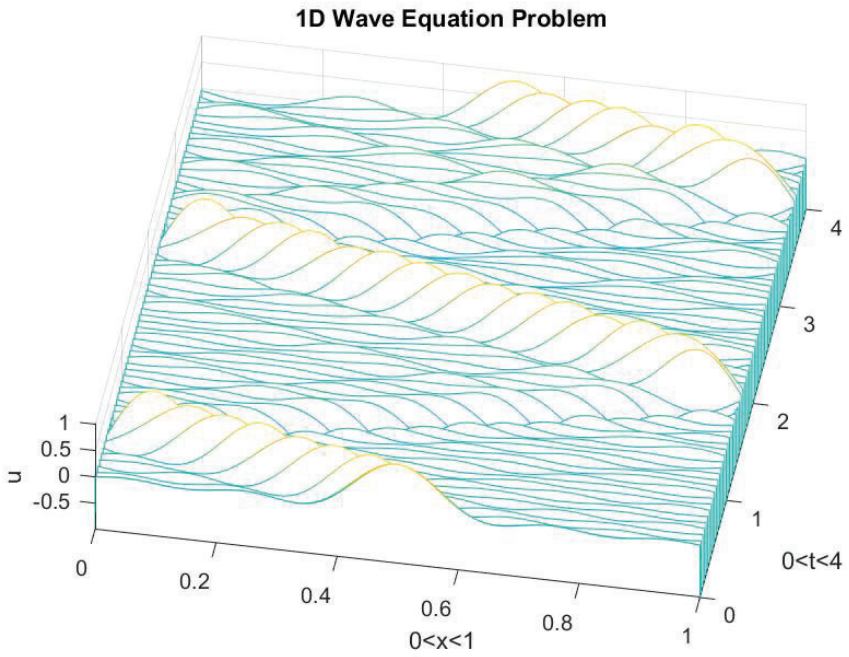

Figure 5: 1D Wave Equation Problem from $0 < t < 4$

The MATLAB script is given in Appendix B.

4.3.2 Spectral Collocation Method

The Spectral collocation method follows similarly to the Galerkin method, but the test functions are taken to be shifted delta functions. The delta function has the unique property that for some domain $\Omega \subseteq \mathbb{R}$ and some function $f(x) \in \Omega$, the convolution with a shifted delta function $\delta(x - x_i)$ for $x_i \in \Omega$ is

$$\int_{\Omega} f(x)\delta(x - x_i)dx = f(x_i). \tag{40}$$

Within the same framework of the method of weighted residuals, we define a set of test functions,

$$\{\psi\} = \{\psi_i = \delta(x - x_i) \quad | \quad i = 0, 1, 2, 3, \ldots N\}. \tag{41}$$

Then, we force the residual R_N to be 0 by taking the inner product with our set of test functions to create a system of $N + 1$ equations, where each equation is defined by

$$\langle R_N, \delta(x - x_i)\rangle = 0. \tag{42}$$

This can be rewritten as

$$\frac{\partial u_N(x_i,t)}{\partial t} - \mathcal{L}_N[u_N(x_i,t)] = 0, \tag{43}$$

where $u_N(x_i,t)$ is the series expansion evaluated at the collocation points x_i and $\mathcal{L}_N$ is the discrete approximation of the operator $\mathcal{L}$. Spectral collocation methods deal with approximating this differential operator. A dense matrix is commonly obtained to compute the derivative of the approximate solution at an evaluated point. The developers of Spectral Methods, David Gottlieb and Steven Orszag [7] [9], approach this problem by continuing the theme of orthogonal series expansions and applying analytical quadrature methods to define differentiation matrices. In contrast, Trefethen [16] approaches the problem with the intuition of interpolation. Both approaches are explored within this paper and cover the topic of collocation grids in different ways.

This paper considers two series expansions, the Chebyshev and Fourier expansions. The Chebyshev expansion considers Dirichlet boundary conditions whereas the Fourier expansion considers periodic boundary conditions. A Chebyshev differentiation matrix is constructed using the perspective of interpolation on the non-uniform Gauss-Lobatto grid and Lagrange polynomials. Alternatively, the Chebyshev differentiation matrix can be defined by applying the Gauss-Lobatto quadrature to series expansion. For the case of Fourier series, this paper constructs a Fourier differentiation matrix using Fourier series and the trapezoid-rule quadrature on a uniform grid.

After defining the differentiation matrix and evaluating the PDE at collocation points, the resulting system of ODEs in (43) can be solved using a numerical ODE solver such as MATLAB's adaptive fourth order Runge-Kutta solver, ODE45. We will see later on that the resulting set of ODEs for the PWE problem is much stiffer for a Chebyshev spectral method.

It should be noted that the Spectral collocation method for solving time dependent IBVPs is akin to the procedure for the Method of Lines (MOL).

5 Numerical Method For Laser Beam Propagation

The method of lines is implemented to solve an initial boundary-value problem (IBVP) formulation of the PWE. The initial condition gives the transverse shape of the beam leaving the laser's aperture. The laser is propagating forward using time integration schemes. In this paper, the boundaries on the x, y domain are set far from the beam center in an attempt to mitigate their effect on the beam.

The method of lines is implemented by first discretizing the x,y domain Ω using a finite set of discrete points $(x_i, y_j) \in \Omega$ for $i, j = 0, 1, 2, 3 \ldots N$ for some $N \in \mathbb{N}$. A Spectral method is used to approximate the derivatives with respect to x, y of the solution $u(x_i, y_i, z)$ at each discrete point. The resulting semi-discrete system of ordinary equations are then solved using time-integration

methods. This paper implements an adaptive 4th order Runge-Kutta method through MATLAB's ODE45 solver for time integration.

5.1 Differentiation Matrix

To approximate the derivative at discrete points, a differentiation matrix is applied. Unlike finite difference methods, Spectral methods use the data from every available point in the domain to approximate the derivative at a single point. This results in a dense differentiation matrix D. Differentiation can be approximated using matrix operations with D, so for some function f and some set of discrete points $\vec{x}$,

$$\left.\frac{\partial f}{\partial x}\right|_{x=\vec{x}} \approx Df(\vec{x}). \tag{44}$$

Higher order derivatives can be computed by multiplying the differentiation matrix D by itself. For example, the second derivative $\frac{\partial^2}{\partial x^2}$ is approximated by D^2. The discrete two-dimensional Laplacian operator can be approximated for a multi-variable function $g(x, y)$ with the following matrix operation,

$$\Delta g \implies D^2\vec{G} + (D^2\vec{G}^T)^T = D^2\vec{G} + \vec{G}(D^2)^T \tag{45}$$

where $G_{ij} = g(x_i, y_j)$ is a matrix of data points evaluated at the grid points x_i, y_j,

$$\vec{G} = \begin{bmatrix} g(x_0, y_0) & g(x_0, y_1) & \dots & g(x_0, y_{N-1}) \\ g(x_1, y_0) & g(x_1, y_1) & \dots & g(x_0, y_{N-1}) \\ \vdots & \ddots & \ddots & \vdots \\ g(x_{N-1}, y_0) & g(x_{N-1}, y_1) & \dots & g(x_{N-1}, y_{N-1}) \end{bmatrix}. \tag{46}$$

Applying our discrete Laplacian operator, we approximate the Paraxial Wave Equation's spatial differential operator $\mathcal{L}[g(x, y)] = \frac{1}{\alpha}(\Delta g(x, y) + \beta g(x, y))$ at our grid points x_j, y_j,

$$\mathcal{L}_N(G) = \frac{1}{\alpha}(D^2\vec{G} + \vec{G}(D^2)^T + \vec{\beta} \cdot \vec{G}) \tag{47}$$

This approximation for L will be used later on to solve the Paraxial Wave Equation using D and ODE45.

5.2 Chebyshev Collocation Method

The first method we present is a Chebyshev collocation method for approximating our differential operator $\mathcal{L}$ on a non-uniform grid using a differentiation matrix D. This method is appropriate for Dirichlet boundary conditions. Our grid points on $(-1, 1)$ are defined by the Chebyshev points,

$$x_j = \cos(\frac{j\pi}{N}), \qquad j = 0, 1, 2, \cdots, N. \tag{48}$$

Chebyshev points can be geometrically interpreted as projections of equidistant points on a unit circle as shown in Figure 6.

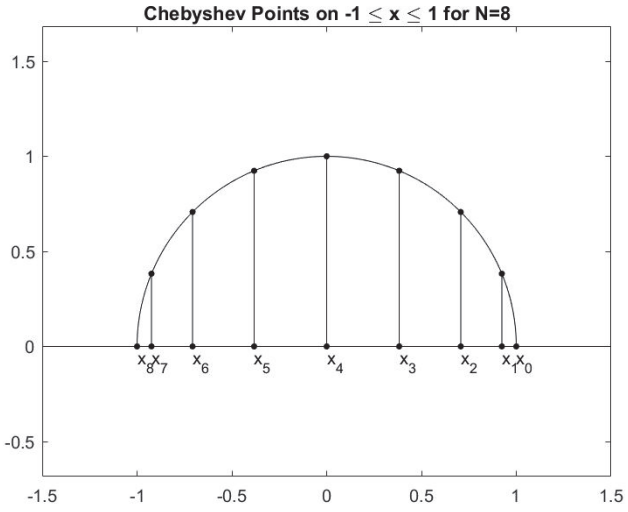

Figure 6: Chebyshev points can be interpreted as projections of equidistant points on a unit circle.

The grid in (48) for the interval $(-1, 1)$ can be generalized to (a, b) by

$$x_j = \frac{b-a}{2}\cos(\frac{j\pi}{N}) + \frac{a+b}{2}, \qquad j = 0, 1, 2, \ldots, N. \tag{49}$$

These special grid points are chosen to reduce a numerical instability called the Runge phenomenon when applying polynomial interpolation on a uniform grid.

5.2.1 Polynomial Interpolation

The idea of interpolation is a vital concept in applying Spectral methods. We will see in the next section how interpolation can be applied to approximate differentiation. This section will explore polynomial interpolation on a uniform grid.

The general form of an n^{th} degree polynomial $p(x)$ is given by

$$p(x) = b_0 + b_1 x + b_2 x^2 + ... + b_{n-1}x^{n-1} + b_n x^n, \tag{50}$$

where b_j is the expansion coefficient for the j^{th} term. Given data points in a vector $\vec{v}$ corresponding to grid points in $\vec{x}$, we can define an interpolating polynomial by solving for the b coefficients for a polynomial that satisfies each

component v_j in $\vec{v}$ and x_j in $\vec{x}$:

$$\begin{aligned} p(x_0) &= b_0 + b_1 x_0 + b_2 x_0^2 + ... + b_{n-1} x_0^{n-1} + b_n x_0^n = v_0, \\ p(x_1) &= b_0 + b_1 x_1 + b_2 x_1^2 + ... + b_{n-1} x_1^{n-1} + b_n x_0^n = v_1, \\ &\vdots \\ p(x_n) &= b_0 + b_1 x_n + b_2 x_n^2 + ... + b_{n-1} x_n^{n-1} + b_n x_n^n = v_n. \end{aligned} \tag{51}$$

This linear system can be written in matrix form,

$$A\vec{b} = \vec{v}, \tag{52}$$

where $\vec{b}$ is a vector composed of the unknown b values and A is the Vandermonde matrix corresponding to the grid points,

$$A = \begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & x_2^3 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 1 & x_n & x_n^2 & x_n^3 & \dots & x_n^n \end{bmatrix}. \tag{53}$$

As an example, let us consider the following function,

$$f(x) = \frac{1}{1 + 16x^2}, \qquad -1 \leq x \leq 1. \tag{54}$$

We first need to discretize the domain, so we divide $[-1, 1]$ into N intervals with $N + 1$ points. An intuitive approach to discretizing the domain would be to define equidistant points where

$$x_j = -1 + \frac{2j}{N}, \qquad j = 0, 1, 2 \dots N. \tag{55}$$

Given in Figure 7 is an interpolation of equation (54). We see that using equidistant grid points with polynomial interpolation causes a type of numerical instability called the Runge phenomenon [16]. The Runge phenomenon creates large oscillations close to the boundary, which is exacerbated further with a refined grid.

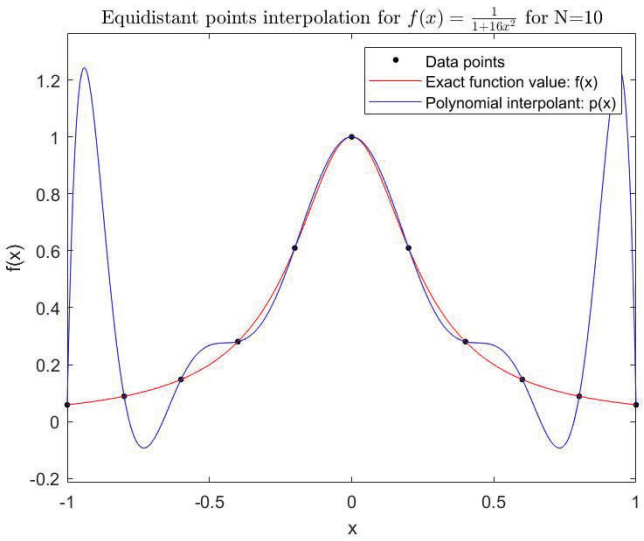

Figure 7: Interpolation of equation (54) using equidistant points.

To counteract the Runge phenomena, we apply Chebyshev points defined in equation (48).

Given in Figure 8 is a polynomial interpolation of equation (54) with Chebyshev points. We can see the Runge phenomena has been successfully reduced. Unlike equidistant points, the polynomial interpolant $p(x)$ will converge to $f(x)$ as more Chebyshev points are added.

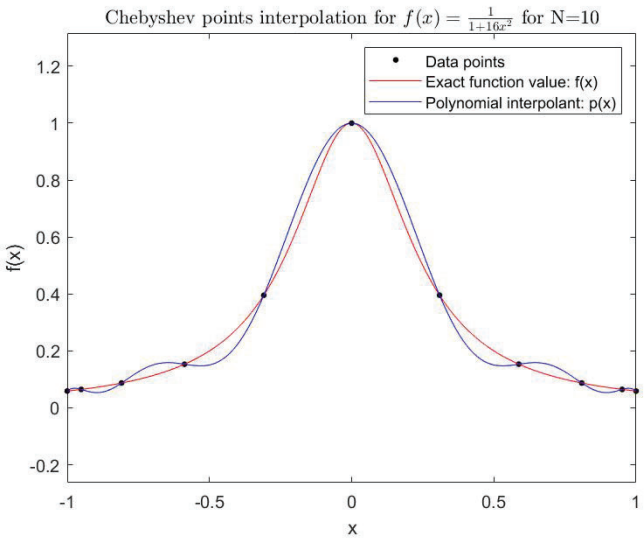

Figure 8: Interpolation of equation (54) using Chebyshev points.

An analytical proof of selecting Chebyshev points can be shown by minimizing the L_∞ norm for polynomial interpolation [2].

Differentiation can be approximated through interpolation. We can approximate the derivative of a function by differentiating its interpolating polynomial,

$$p'(x) = 0 + b_1 + b_2 2x + ... + b_{n-1} n x^{n-1}, \tag{56}$$

which results in the following matrix,

$$B = \begin{bmatrix} 0 & 1 & 2x_0 & 3x_0^2 & \dots & nx_0^{n-1} \\ 0 & 1 & 2x_1 & 3x_1^2 & \dots & nx_1^{n-1} \\ 0 & 1 & 2x_2 & 3x_2^2 & \dots & nx_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 1 & 2x_n & 3x_n^2 & \dots & nx_n^{n-1} \end{bmatrix}. \tag{57}$$

For B, we can find the discrete derivative $\vec{w}$ of $\vec{v}$ with the following relationship:

$$\vec{w} = B\vec{b} \tag{58}$$

From Equation (52), we see that

$$\vec{b} = A^{-1}\vec{v}. \tag{59}$$

Now we have the relationship,

$$\vec{w} = BA^{-1}\vec{v}. \tag{60}$$

Therefore we can define the differentiation matrix D as

$$D = BA^{-1}. \tag{61}$$

5.2.2 Chebyshev Differentiation Matrix

The previous section showed that to derive the differentiation matrix D in (44), we first find a polynomial interpolant $p(x)$ of $f(\vec{x})$ for some function $f(x)$ and some grid $\vec{x}$. Then, we analytically differentiate $p(x)$ to approximate $f'(x)$.

Another way to represent this process is to use Lagrange polynomials where the interpolant can be constructed by

$$p(x) = \sum_{j=0}^{N} f(x_j) p_j(x), \qquad p_j(x) = \prod_{k=0, k\neq j}^{N} \frac{x - x_j}{x_k - x_j}. \tag{62}$$

Differentiating the Lagrange polynomial, we get

$$p'(x) = \sum_{j=0}^{N} f(x_j) p_j'(x), \qquad p_j'(x) = p_j(x) \sum_{k=0, k\neq j}^{N} \frac{1}{x - x_k}. \tag{63}$$

The derivative of x_i can be found by evaluating $p'(x_i)$. It is worth noting Chebyshev points hold the following relationship $x_j = -x_{N-j}$, so $p_j'(x_i) = p_j \frac{1}{x_i - x_j}$ for $i \neq j$ and $p_j(x_j) = \sum_{k=0, k\neq j}^{N} \frac{1}{x_j - x_k}$. It follows that our differentiation matrix D can be defined by

$$p'(\vec{x}) = \begin{bmatrix} p'(x_0) \\ p'(x_1) \\ \vdots \\ p'(x_N) \end{bmatrix} = \begin{bmatrix} d_{00} & \frac{p_1(x_0)}{x_0 - x_1} & \cdots & \frac{p_N(x_0)}{x_0 - x_N} \\ \frac{p_0(x_1)}{x_1 - x_0} & d_{11} & \cdots & \frac{p_N(x_1)}{x_1 - x_N} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{p_0(x_N)}{x_N - x_0} & \frac{p_2(x_N)}{x_N - x_1} & \cdots & d_{NN} \end{bmatrix} \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix} = D\vec{f}, \tag{64}$$

where

$$d_{jj} = \sum_{k=0, k\neq j}^{N} \frac{1}{x_j - x_k}. \tag{65}$$

The matrix D can be defined more explicitly with closed form formulas as given by Trefethen [16]. The main diagonal entries are given by

$$D_{NN} = -\frac{2N^2 + 1}{6}, \qquad D_{00} = \frac{2N^2 + 1}{6}, \tag{66}$$

and the off diagonal entries given by,

$$D_{jj} = \frac{-x_j}{2N - 1}, \qquad j = 1, ..., N - 1, \tag{67}$$

and edges of the matrix given by,

$$D_{ij} = \frac{c_i}{c_j} \frac{(-1)^{i+j}}{x_i - x_j}, \qquad i \neq j, c_i = \begin{cases} 2, & i = 0, N \\ 1. & \text{otherwise} \end{cases} \tag{68}$$

Trefethen codes this matrix succinctly in his *cheb.m* function,

```
% CHEB  compute D = differentiation matrix, x = Chebyshev grid
function [D,x] = cheb(N)
if N==0, D=0; x=1; return, end
x = cos(pi*(0:N)/N)';
c = [2; ones(N-1,1); 2].*(-1).^(0:N)';
X = repmat(x,1,N+1);
dX = X-X';
D  = (c*(1./c)')./(dX+(eye(N+1)));      % off-diagonal entries
D  = D - diag(sum(D'));                 % diagonal entries
```

5.3 Fourier Collocation Method

A Fourier Spectral method approximates the derivative of a function using trigonometric polynomials as the basis functions for the series expansion approximation. To derive D, we consider a Fourier series expansion of some function $f(x)$ on a periodic domain $[0,\, 2\pi)$,

$$f(x) = \sum_{n=-\infty}^{\infty} a_n e^{inx}, \tag{69}$$

where $a_n \in \mathbb{C}$ are expansion coefficients and e^{inx} for $n \in \mathbb{Z}$ are a set of basis functions. These functions are orthogonal on the interval $[0,\, 2\pi)$. The expansion coefficients can be computed by

$$a_n = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-inx} dx, \tag{70}$$

which is a direct result of the orthogonality property of our set of basis functions.

Now, we approximate a_n on an equidistant grid on $[0,\, 2\pi]$ where $x_j = hj$ for $j = 0, 1, 2, 3, \ldots N$ and $h = \frac{2\pi}{N}$ for an even number N. The odd case follows in a similar manner. We use the trapezoid quadrature formula given by

$$\int_0^{2\pi} g(x) dx \approx h(\sum_{j=1}^{N-1} g(x_j) + \frac{g(0)}{2} + \frac{g(2\pi)}{2}). \tag{71}$$

Then, it follows from (71), the expansion coefficient a_n can be approximated by

$$\hat{a}_n = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) e^{-inx_j}, \tag{72}$$

resulting from the periodicity of f where $f(0) = f(2\pi)$.

Now, our problem can essentially be seen as an interpolation problem using trigonometric polynomials. Given data points evaluated at $f(x_j)$, we can find an interpolant $p(x)$ by truncating the Fourier series expansion of f using N number of Fourier modes where N is even,

$$p(x) = \sum_{n=-N/2}^{N/2} \frac{\hat{a}_n}{c_n} e^{inx}, \qquad c_n = \begin{cases} 2 & n = \pm\frac{N}{2}, \\ 1 & \text{else}, \end{cases} \tag{73}$$

where c_n is introduced for numerical stability when truncating the Fourier series expansion [9]. Expanded out (73) becomes

$$p(x) = \sum_{n=-N/2}^{N/2} \frac{1}{c_n N} \sum_{j=0}^{N-1} f(x_j) e^{-inx_j} e^{inx}. \tag{74}$$

We first evaluate the external sum,

$$p(x) \quad = \quad \sum_{j=0}^{N-1} f(x_j) \sum_{n=-N/2}^{N/2} \frac{1}{c_n N} e^{in(x-x_j)}.$$

We can sum the following as a geometric series rewritten as

$$\begin{aligned}
\sum_{n=-N/2}^{N/2} \frac{1}{c_n N} e^{in(x-x_j)} &= \frac{1}{2N} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} f(x_j) e^{in(x-x_j)} + \frac{1}{2N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_j) e^{in(x-x_j)} \\
&= \frac{e^{-i\frac{N}{2}(x-x_j)} - e^{i\frac{N}{2}(x-x_j)} + e^{i(-\frac{N}{2}+1)(x-x_j)} - e^{i(\frac{N}{2}+1)(x-x_j)}}{1 + e^{i(x-x_j)}} \\
&= \frac{-e^{i(x-x_j)\frac{1}{2}}(e^{i(x-x_j)\frac{1}{2}} + e^{-i(x-x_j)\frac{1}{2}})(e^{i(x-x_j)\frac{N}{2}} - e^{-i(x-x_j)\frac{N}{2}})}{-Ne^{i(x-x_j)\frac{1}{2}}(e^{i(x-x_j)]\frac{1}{2}} - e^{-i(x-x_j)\frac{1}{2}})}.
\end{aligned}$$

Using trigonometric identities, the summation becomes the real-valued periodic Sinc function,

$$S_N(x - x_j) = \frac{1}{N} \cot(\frac{x - x_j}{2}) \sin(N\frac{x - x_j}{2}). \tag{75}$$

Now, we can rewrite $p(x)$ in the form of an interpolation function that approximates $f(x)$ using the uniformly distributed grid points x_j,

$$p(x) = \sum_{j=0}^{N-1} f(x_j) S_N(x - x_j) \tag{76}$$

We can approximate $\frac{df}{dx}$ by differentiating its truncated Fourier series approximation $p(x)$,

$$f'(x) \approx p'(x) = \sum_{j=0}^{N-1} f(x_j) S'_N(x - x_j) \tag{77}$$

The differentiation matrix D is defined by evaluating at a set of grid points x_k for $k = 0, 1, 2, 3, \ldots N$,

$$p'(\vec{x}) = \begin{bmatrix} S'_N(0) & S'_N(x_1 - x_0) & \ldots & S'_N(x_N - x_0) \\ S'_N(x_0 - x_1) & S'_N(0) & \ldots & S'_N(x_N - x_1) \\ \vdots & \ddots & \ddots & \vdots \\ S'_N(x_0 - x_N) & S'_N(x_1 - x_N) & \ldots & S'_N(0) \end{bmatrix} \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}. \tag{78}$$

For our case, we define our grid points using the same uniform mesh as our expansion coefficient approximation: $x_k = \frac{2\pi k}{N}$. These grid points produce the following relationship: $S_N(x_k - x_j) = S_N(x_{k-j})$. Coupled with the periodicity of $p(x)$, $S_N(x_{-j}) = S_N(x_{N-j})$, it can be shown that D is a Toeplitz matrix,

$$\begin{bmatrix} S_N''(0) & S_N''(x_1) & \dots & S_N''(x_N) \\ S_N''(x_{N-1}) & S_N''(0) & \ddots & S_N''(x_0) \\ \ddots & \ddots & \ddots & \ddots \\ S_N''(x_0) & S_N''(x_{N-1}) & \dots & S_N''(0) \end{bmatrix}. \tag{79}$$

The D^2 matrix can be found by differentiating S_N twice and evaluating at our grid points. The entries of our Fourier D^2 matrix is given by Trefethen [16] and Hesthaven [9],

$$S_N''(x_j) = \begin{cases} -\frac{\pi^2}{3h^2} - \frac{1}{6} & j = 0, N, \\ \frac{(-1)^j}{2\sin^2(jh/2)} & \text{else.} \end{cases} \tag{80}$$

6 Verification of Method

We verify our numerical schemes in two ways. We first study laser beam propagation in free-space. An exact solution to the PWE exists for the Gaussian beam for an unbounded domain as introduced in section 3.1. The application of our numerical scheme necessitates the formulation of an IBVP on a bounded domain with boundary conditions. We compare our numerical solution to the IBVP to the unbounded Gaussian beam solution. To mitigate boundary condition effects, we select a domain such that the boundaries are sufficiently far enough from the laser beam center. We will see later from computations that a boundary that is set twice the distance of the beam radius away from the beam center is sufficient to mitigate boundary condition effects. The boundary becomes a more significant factor as the beam expands. The drawback to selecting a large domain is beam resolution. A larger domain requires finer discretizations to resolve a laser beam, which increases the computational demand of the method. A balance needs to be found between preventing boundary effects and computational expenditure.

We consider a Fourier Galerkin Method, a Chebyshev collocation method, and a Fourier collocation method in our free-space study to resolve the transverse spread of the beam. An adaptive Runge-Kutta Method is applied using MATLAB's ODE45 solver to propagate the beam forward along the z-axis. Out of the three approximations for the transverse spread of the beam, the Fourier collocation method is the most efficient when coupled with MATLAB's ODE45 solver. MATLAB's ODE solvers struggle for refined Chebyshev approximation grids for this laser problem. The incompatibility between the Spectral approximation and time integration method falls beyond the scope of this study. The

inefficiencies from the Galerkin method are a direct result of the integral computation to find the initial coefficient function values for the series expansion.

We then proceed with a refractive index coefficient by applying the method of manufactured solutions.

6.1 Propagation in Free Space

To numerically simulate laser beam propagation in a homogeneous medium, we present an Initial-Boundary Value Problem (IBVP) for a Gaussian beam. Our numerical scheme applies the method of lines by discretizing the x, y domain on $\Omega = [a, b] \times [a, b]$ and applying a time stepping scheme for the z-axis. Then, our IBVP can be presented as

$$\frac{\partial V}{\partial z} = \mathcal{L}[U], \qquad \frac{\partial U}{\partial z} = -\mathcal{L}[V], \tag{81}$$

where $\mathcal{L}[f] = \frac{1}{\alpha}(\Delta f + \beta f)$ is the spatial differential operator for the x, y domain of the PWE.

Our initial conditions model a Gaussian beam leaving the laser's aperture using the unbounded exact solution of the Gaussian Beam,

$$U(x, y, 0) = e^{-r^2} \cos(\frac{\alpha}{4} r^2), \qquad V(x, y, 0) = -e^{-r^2} \sin(\frac{\alpha}{4} r^2), \tag{82}$$

where $r^2 = x^2 + y^2$. We can find the exact unbounded solution in terms of U, V by separating the real and imaginary parts of the exact solution in equation (8):

$$\begin{cases} U(\bar{x}, \bar{y}, \bar{z}) = \frac{1}{\sqrt{\Theta_0^2 + \Lambda_0^2}} e^{-\frac{\bar{r}^2}{W(\bar{z})^2}} \cos(\varphi + \frac{k\bar{r}^2}{2F(\bar{z})}), \\ V(\bar{x}, \bar{y}, \bar{z}) = -\frac{1}{\sqrt{\Theta_0^2 + \Lambda_0^2}} e^{-\frac{\bar{r}^2}{W(\bar{z})^2}} \sin(\varphi + \frac{k\bar{r}^2}{2F(\bar{z})}), \end{cases} \tag{83}$$

for the physical variables $\bar{r}^2 = \bar{x}^2 + \bar{y}^2$ and $\bar{z}$ in units of meters.

Boundary conditions are either homogeneous Dirichlet or periodic boundary conditions. We attempt to mitigate boundary effects by setting the boundary far from the beam center.

Now, we can apply the method of lines to our IBVP to simulate a propagating Gaussian beam. We first approximate U, V by evaluating at our grid points $U(x_i, y_i, z), V(x_i, y_i, z)$, creating a semi-discrete set of $(N + 1)^2$ equations to solve:

$$\frac{\partial \vec{V}}{\partial z} = \mathcal{L}_N[\vec{U}], \qquad \frac{\partial \vec{U}}{\partial z} = -\mathcal{L}_N[\vec{V}]. \tag{84}$$

where $\vec{U_{ij}} = U(x_i, y_i, z)$ and $\vec{V_{ij}} = V(x_i, y_i, z)$ are matrices,

$$\vec{U} = \begin{bmatrix} U(x_0, y_0, t) & U(x_0, y_1, t) & \dots & U(x_0, y_{N-1}, t) \\ U(x_1, y_0, t) & U(x_1, y_1, t) & \dots & U(x_0, y_{N-1}, t) \\ \vdots & \ddots & \ddots & \vdots \\ U(x_{N-1}, y_0, t) & U(x_{N-1}, y_1, t) & \dots & U(x_{N-1}, y_{N-1}, t) \end{bmatrix}, \tag{85}$$

$$\vec{V} = \begin{bmatrix} V(x_0, y_0, t) & V(x_0, y_1, t) & \dots & V(x_0, y_{N-1}, t) \\ V(x_1, y_0, t) & V(x_1, y_1, t) & \dots & V(x_0, y_{N-1}, t) \\ \vdots & \ddots & \ddots & \vdots \\ V(x_{N-1}, y_0, t) & V(x_{N-1}, y_1, t) & \dots & V(x_{N-1}, y_{N-1}, t) \end{bmatrix}. \tag{86}$$

Similarly, we construct our initial condition matrix by evaluating at our grid points,

$$\vec{U}_{0_{ij}} = U(x_i, y_j, 0), \qquad \vec{V}_{0_{ij}} = V(x_i, y_j, 0).$$

For time stepping, we apply MATLAB's ODE45 to solve our system of N^2 ODEs. MATLAB's ODE45 solver uses an adaptive Runge-Kutta method. The solver uses its own internal steps to compute solutions. If manual times steps are inputted, the manual time steps are approximated from the internal adaptive steps [14]. The following results in this paper manually sets time steps from $z = 0$ to $z = 2$ with 1000 uniformly distributed points.

6.1.1 Fourier Galerkin Method Solution

We implement a Fourier Galerkin Method, a special form of the Method of Weighted Residuals (MWR), to solve our IBVP. We first construct two approximations of U, V in the form of truncated sine series expansion, $\tilde{U}$ $\tilde{V}$ where

$$\tilde{U} = \sum_{m=1}^{N} \sum_{n=1}^{N} a_{mn}(z)\phi_{mn}(x, y), \tag{87}$$

$$\tilde{V} = \sum_{m=1}^{N} \sum_{n=1}^{N} b_{mn}(z)\phi_{mn}(x, y), \tag{88}$$

with expansion coefficients a_{mn} and b_{mn}. For our basis functions in our function approximation also known as trial functions in the MWR, we use a Fourier sine series,

$$\phi_{mn}(x, y) = \sin(\frac{n\pi x}{L}) \sin(\frac{m\pi y}{L}) \qquad m, n \in \mathbb{N}. \tag{89}$$

This set of basis functions contain two important properties. The first property is the set of functions' orthogonality property. We define an inner product for functions $g(x, y)$ and $h(x, y)$ defined on [0, L] as

$$\langle h(x, y), g(x, y) \rangle = \int_0^L \int_0^L h(x, y)g(x, y)dxdy. \tag{90}$$

For our particular choice of basis functions, it can be shown for every $m', n', m, n \in \mathbb{N}$, the following orthogonality relationship holds in equation (91). Other sets of functions sometimes requires a weight function $w(x)$ to be orthogonal.

$$\langle \phi_{mn}(x, y), \phi_{m'n'}(x, y) \rangle = \begin{cases} 0 & \text{if } n \neq n' \text{ or } m \neq m', \\ \frac{L^2}{4} & \text{if } n = n' \text{ and } m = m'. \end{cases} \tag{91}$$

The second important property is the set of functions contains an eigenvalue-eigenfunction relationship in respect to the linear operator $\mathcal{L} = -\Delta$:

$$\mathcal{L}[\phi_{mn}] = -\Delta[\phi_{mn}] = \lambda_{mn}\phi_{mn} \tag{92}$$

where the eigenvalues are defined by $\lambda_{mn} = \frac{n^2\pi^2}{L^2} + \frac{m^2\pi^2}{L^2}$. In addition, it can be shown the basis functions satisfy our IBVP's Dirichlet boundary conditions, so we do not have to worry about imposing them later.

$$\phi_{mn}(0, y) = \phi_{mn}(L, y) = \phi_{m.n}(x, 0) = \phi_{m.n}(x, L) = 0$$

Now, we need to define the residuals R_1, R_2. In the context of differential equations, the residuals are the difference between the exact solutions U, V and the approximate solutions $\tilde{U}$, $\tilde{V}$:

$$\begin{cases} R_1 = (\Delta U - \alpha\frac{\partial V}{\partial z}) - (\Delta\tilde{U} - \alpha\frac{\partial \tilde{V}}{\partial z}) \\ R_2 = (\Delta V + \alpha\frac{\partial U}{\partial z}) - (\Delta\tilde{V} + \alpha\frac{\partial \tilde{U}}{\partial z}) \end{cases} \tag{93}$$

The idea of the MWR is to average out the residual R_1, R_2 over the interval $[0\ L]$ by taking the inner product with test functions ψ and setting them equal to 0. In the Galerkin method, the test functions ψ are the same as the basis functions, $\psi = \phi$.

$$\begin{cases} \langle R_1, \phi_{mn}\rangle = 0, \\ \langle R_2, \phi_{mn}\rangle = 0. \end{cases} \tag{94}$$

where $m, n = \{1, 2, 3, ..., N\}$. Separating using the linearity of the inner product, we get

$$\begin{cases} \langle(\Delta\tilde{U} - \alpha\frac{\partial \tilde{V}}{\partial z}), \phi_{mn}\rangle = \langle(\Delta U - \alpha\frac{\partial V}{\partial z}), \phi_{mn}\rangle, \\ \langle(\Delta\tilde{V} - \alpha\frac{\partial \tilde{U}}{\partial z}), \phi_{mn}\rangle = \langle(\Delta V - \alpha\frac{\partial U}{\partial z}), \phi_{mn}\rangle. \end{cases}$$

From the original PDE, we see the right side of the equation is equal to 0 and we get $2N^2$ Galerkin equations,

$$\begin{cases} \langle(\Delta\tilde{U} - \alpha\frac{\partial \tilde{V}}{\partial z}), \phi_{mn}\rangle = 0, \\ \langle(\Delta\tilde{V} - \alpha\frac{\partial \tilde{U}}{\partial z}), \phi_{mn}\rangle = 0. \end{cases} \tag{95}$$

Now, we substitute our series expansion for $\tilde{U}$ $\tilde{V}$ and simplify using the orthogonality of ϕ and its eigenvalue-eigenfunction relationship. Now, we have to find the expansion coefficients a_{mn} and b_{mn} by solving the following $2N^2$ coupled first-order ODEs:

$$\begin{cases} a_{mn}(z)\lambda_{mn} - \alpha b'_{mn}(z) = 0, \\ b_{mn}(z)\lambda_{mn} + \alpha a'_{mn}(z) = 0, \end{cases} \tag{96}$$

with initial conditions,

$$\begin{cases} a(0) = \frac{4}{L^2}\langle U(x, y, 0), \phi_{mn}(x, y)\rangle, \\ b(0) = \frac{4}{L^2}\langle V(x, y, 0), \phi_{mn}(x, y)\rangle. \end{cases} \tag{97}$$

The numerical results of the Galerkin method using a sine series expansion for our Gaussian beam problem solved on a square domian $[0, L] \times [0, L]$ were promising. To resolve the initial beam, we needed at least N=64 Fourier modes as shown in the Figures below. Note the figures were all plotted with the same resolution (100 by 100 points).

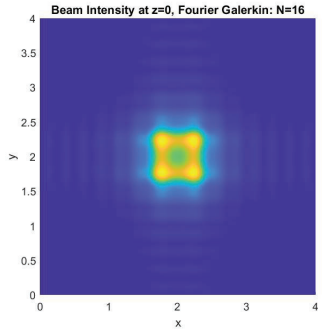

Figure 9: Gaussian beam intensity at z=0 for the Galerkin Method with N=16 modes.

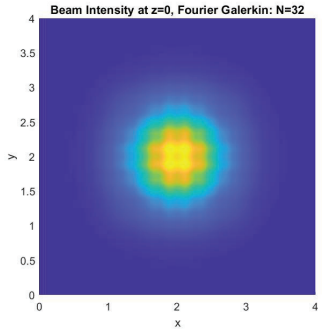

Figure 10: Gaussian beam intensity at z=0 for the Galerkin Method with N=32 modes.

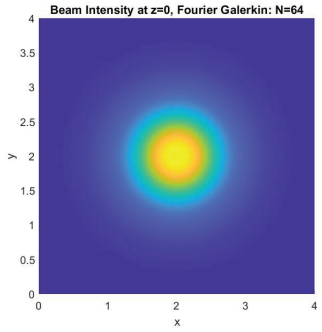

Figure 11: Gaussian beam intensity at z=0 for the Galerkin Method with N=64 modes.

The square shape of the first two approximations $N = 16$ and $N = 32$ are a result of low convergence of the sine series approximation for our Gaussian beam problem. The following plots shows the convergence of the Fourier series for $U(r, 0)$ and $V(r, 0)$.

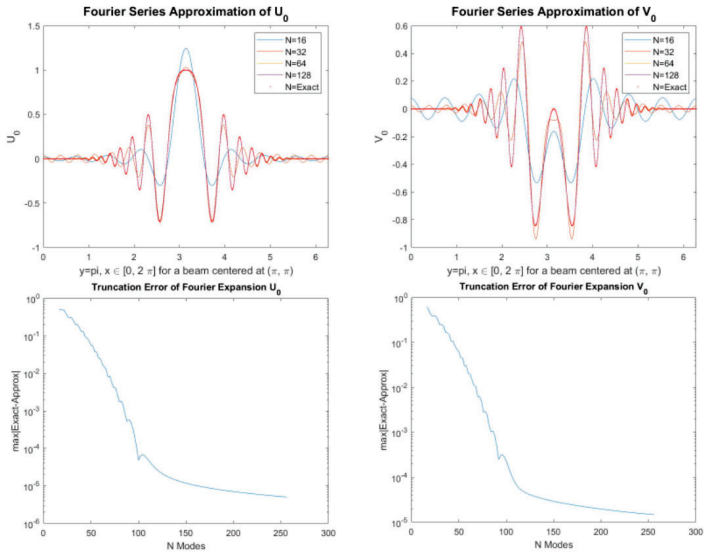

Figure 12: Convergence of the Fourier series approximation of $U(r,0)$ (left) and $V(r,0)$ (right). The bottom plots shows the max error between the exact function and the truncated expansion using N modes.

The series expansion requires a relatively high truncation number to resolve the beam. The following table gives approximate run times for sampled N values:

$N = 16$	$N = 32$	$N = 64$
9 seconds	400 seconds	3500 seconds

The majority of the run time was spent on resolving the $2N^2$ number of initial condition integrals in (97). These integrals were computed using MATLAB's built-in *integral2* numerical integration function with default settings and tolerances. For $N = 64$, MATLAB had trouble resolving these integrals accurately. When approximating the integrals in MATLAB, the max number of function evaluations were reached before reaching the default error tolerance.

Future studies can look at the more efficient computation of these integrals. The integrals in (97) can be computed independently, so high performance computing methods can be implemented to compute the integrals in parallel. Within this avenue, advanced numerical quadrature techniques can be explored for more efficient and accurate evaluations of the initial condition coefficients. Alternatively, a different basis can be implemented for the series expansion, which may positively affect the convergence of the numerical quadrature and the accuracy of the solution approximation.

Since the refractive index term U and V are linear variable coefficient terms, the Galerkin method scheme can be applied to the nonhomogeneous problem as well. However, the integral computation will become even more strenuous.

We will see in the next sections that the collocation methods are much less computationally intense. The computational times are fractions of the time of our implementation of the Galerkin method. This is a direct result of how the integrals for the expansion coefficients were treated. In previous sections, we saw that these integrals were simplified analytically for the differentiation matrix in our implementation of the Fourier collocation method. The analytically simplified quadrature resulted in the special Sinc function in (75). The same analytical simplification can be shown using the Chebyshev polynomial basis using a Gauss-Lobatto quadrature [9]. In comparison to the Collocation methods, the Galerkin method is more computationally intense.

6.1.2 Chebyshev Collocation Method Solution

We apply our numerical scheme using a Chebyshev differentiation matrix to solve the Gaussian beam IBVP. Homogeneous Dirichlet boundary conditions are enforced by setting the exterior of $\vec{U}, \vec{V}$ to 0, which the Gaussian initial condition nearly satisfies automatically. To more strictly enforce the boundary condition and decrease computational expenditure, we take the interior of D and perform operations only on the interior grid points $[x_1, x_2, ...x_{N-1}] \times [y_1, y_2, ...y_{N-1}]$.

Two plots are presented. Both plots give the intensity $I = U^2 + V^2$ at the center of our Gaussian beam. The first plot shows a solution to the IBVP solved on the domain $[-1, 1] \times [-1, 1]$ while the second plot shows the solution solved on $[-2, 2] \times [-2, 2]$ for sampled N values. The first plot reveals some sort of boundary effect, which worsens as N increases. When we push the boundary further away, we can see the solution converges for increasing N values.

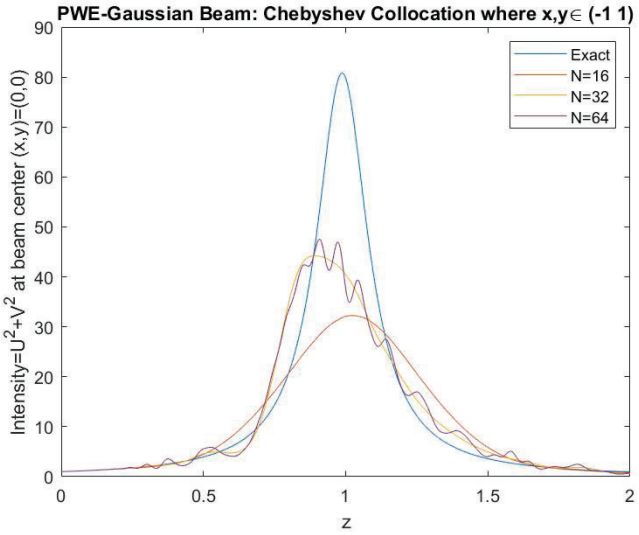

Figure 13: Intensity at the beam center (x,y)=(0,0) varying along the z-axis. PWE solved on the grid [-1, 1] × [-1,1].

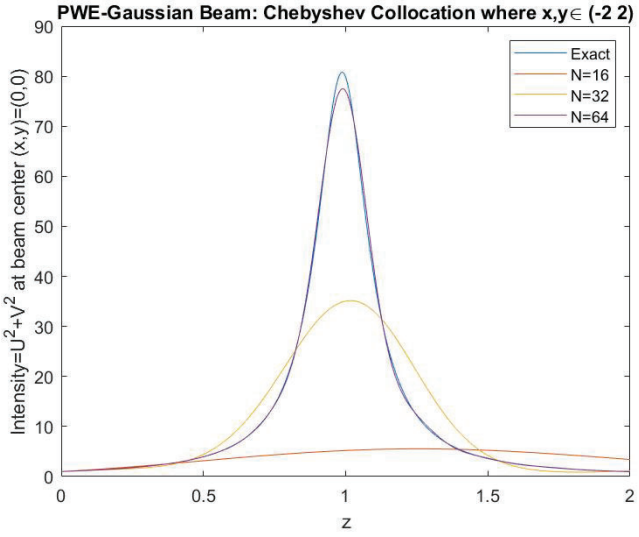

Figure 14: Intensity at the beam center (x,y)=(0,0) varying along the z-axis. PWE solved on the grid [-2, 2] × [-2,2].

The implementation of our Chebyshev collocation method strains MATLAB's ODE solvers as we refine the spatial discretization by increasing N.

MATLAB ODE45 fails to produce a solution for $N = 128$ on the domain $[-2, 2] \times [-2, 2]$. Time integration issues are left for future studies.

The table below gives approximate run times for sampled N values. We can see the Collocation method runs much faster with higher accuracy than the Galerkin method in the previous section.

$N = 16$	$N = 32$	$N = 64$
< 1 seconds	< 1 seconds	12 seconds

In contrast with the Galerkin method, the Collocation method's run time results from solving the resulting ODEs. A large coupled system of ODEs are formed as a result of the implementation of the method of lines on the PWE. A numerical instability occurs when the Chebyshev differentiation matrix is applied to the PWE for high N's. This instability does not occur for the Fourier Collocation method.

6.1.3 Fourier Collocation Method Solution

Out of the three numerical schemes, the Fourier Collocation method combined with Matlab's ODE45 runs the most smoothly and efficiently. Applying the Fourier differentiation matrix, we simulate a Gaussian beam in the domain [0, 2π] with periodic boundary conditions.

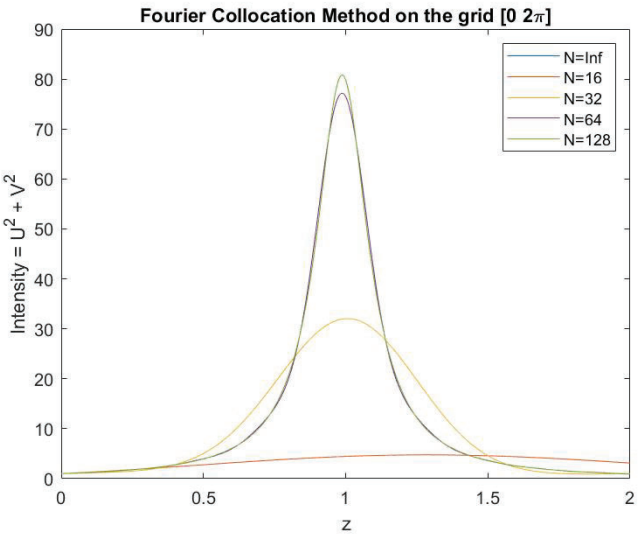

Figure 15: Intensity at the beam center varying along the z-axis. PWE solved on the grid [0, 2π]$\times$ [0, 2π].

The Fourier method solution converged to the exact unbounded solution at the beam center as shown in Figure 15. The error in the beam intensity at the

beam center continues to decrease at the beam center with high N as shown in Figure 16.

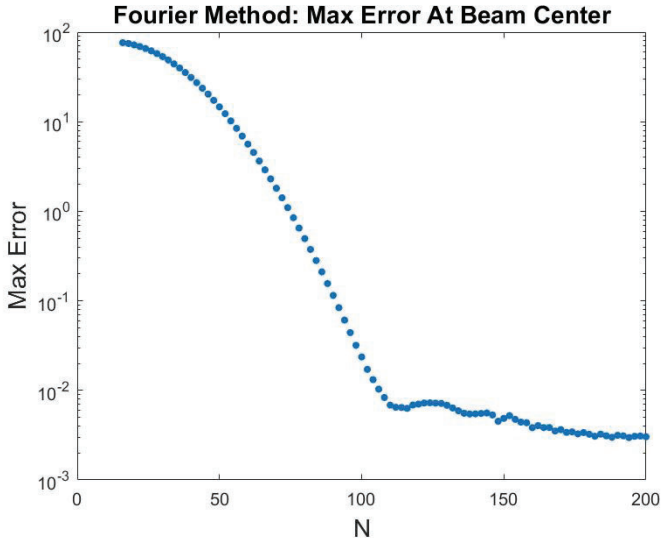

Figure 16: Max error for the Intensities of the Fourier Method Solution and Exact Solution at the Beam Center for N between 16 and 200.

The Gaussian beam is completely resolved for $N = 128$ with an absolute error in the magnitude of 10^{-4} and an efficient run time of about 10 seconds.

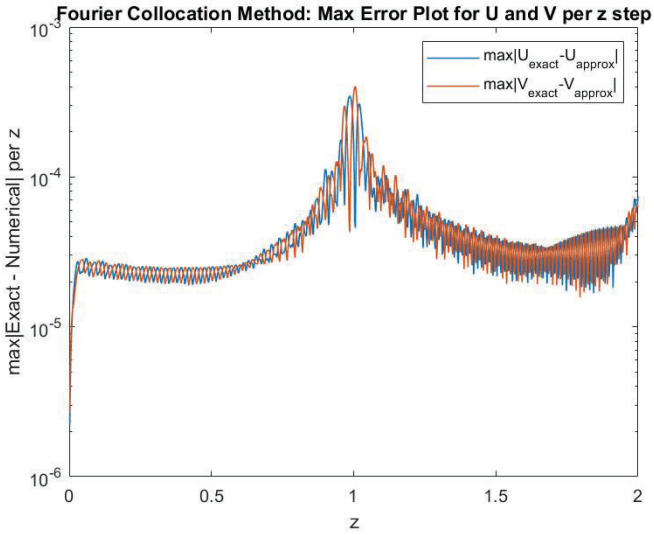

Figure 17: Max error of the Fourier Collocation method solution and exact solution for U, V per point along the propagation axis, z for N=128.

Given in the table below are approximate run times for the Fourier Collocation method for sampled N values. The table shows that the Fourier Collocation method is the most efficient out of the three numerical schemes introduced.

$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
< 1 seconds	< 1 seconds	3 seconds	10 seconds	90 seconds

As a result of the efficiency and accuracy, the Fourier Collocation method will be implemented for the nonhomogeneous propagation problem with $\beta \neq 0$.

6.2 Method of Manufactured Solutions

To further verify our code, we apply the method of manufactured solution. We consider a prototype Initial Boundary Value problem for some constant α and some functions $\beta(x, y, z)$, $f(x, y, z)$ and $g(x, y, z)$ for code verification,

$$\begin{cases} \Delta U - \alpha \frac{\partial V}{\partial z} + \beta U = f, \\ \Delta V + \alpha \frac{\partial U}{\partial z} + \beta V = g, \end{cases} \tag{98}$$

with homogeneous Dirichlet boundary conditions on a square domain,

$$\Omega = (-1, 1) \times (-1, 1), \qquad U|_{\partial\Omega} = V|_{\partial\Omega} = 0. \tag{99}$$

We let $\alpha = 1$ and $\beta = \cos(10xyz)$. To verify our code, we set f and g such that these functions satisfy our PDE given a set of manufactured solutions,

$$U_e = x(1-x^2)y(1-y^2)\cos(z), \qquad V_e = \sin(4\pi x(1-x^2)y(1-y^2))\cos(z). \tag{100}$$

MATLAB's symbolic math toolbox allows symbolic variable manipulation and analytical differentiation. The lines below compute f and g.

```
syms x y z
Alpha =1; %Alpha not equal to 0
Beta (x,y,z)= cos(10*x*y*z); %/(x*y*z)*1;
uexact(x,y,z)=x*(1-x^2)*y*(1-y^2)*cos(z);
vexact(x,y,z)=sin(4*pi*x*(1-x^2)*y*(1-y^2))*cos(z);

g=(diff(diff(uexact,x),x)+diff(diff(uexact,y),y))-Alpha*diff(vexact,z)+Beta*uexact;
f=(diff(diff(vexact,x),x)+diff(diff(vexact,y),y))+Alpha*diff(uexact,z)+Beta*vexact;

f=matlabFunction(f); g=matlabFunction(g);
uexact=matlabFunction(uexact); vexact=matlabFunction(vexact);
Beta=matlabFunction(Beta);
clear x y z
```

Floating point calculations are used after determining f and g. Hence, we convert the symbolic expression into MATLAB function handles using the *matlabFunction* command.

6.2.1 Chebyshev Method

Presented is an error plot for sampled N values for our Chebyshev collocation method.

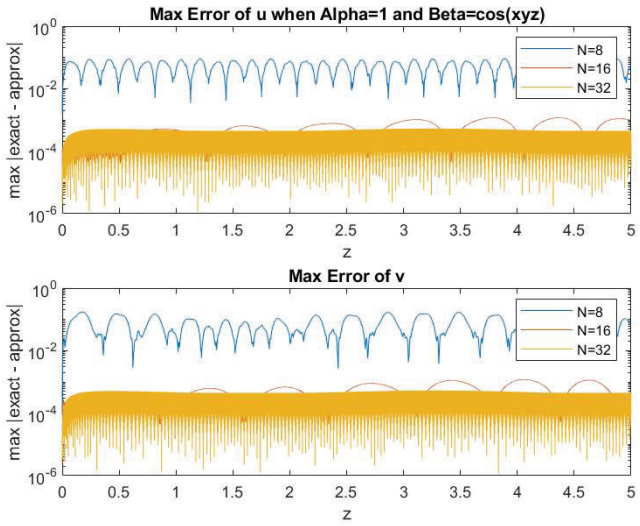

Figure 18: Max error per step z between computed solutions and manufactured solutions.

The Chebyshev collocation method resolves the manufactured solutions well enough. However, our free space results show that it may not be conducive for our laser problems. Therefore, we proceed with the Gaussian beam problem using the Fourier collocation method.

6.2.2 Fourier Method Verification

We verify our code for the Fourier Collocation method using the Gaussian beam functions in equation (82) as manufactured solutions for sampled α parameters. The following figures shows the numerical method is accurate for a constructed variable coefficient term $\beta = 1000\cos(10xyz)$ when the parameter α is small.

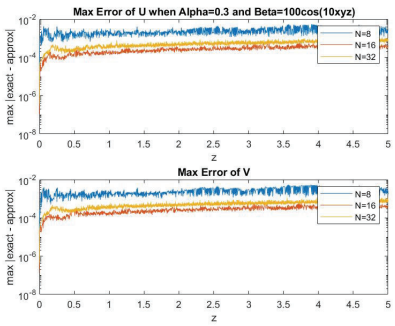

Figure 19: $\alpha = 0.3$

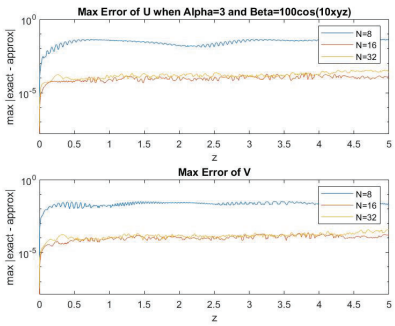

Figure 20: $\alpha = 3$

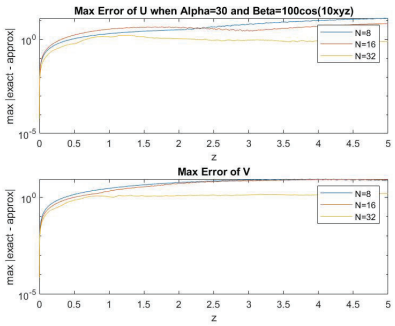

Figure 21: $\alpha = 30$

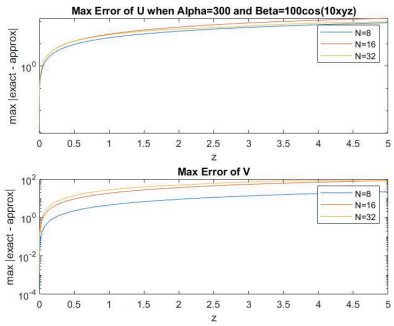

Figure 22: $\alpha = 300$

Figure 23: The figures gives the max error plots per time step for $\alpha = 0.3$ (top left), $\alpha = 3$ (top right), $\alpha = 30$ (bottom left), and $\alpha = 300$ (bottom right).

For the fixed $N = 32$, we see that the Fourier Collocation method is accurate to the order of 10^{-5} for $\alpha = 0.3$ and $\alpha = 3$. However, as the parameter α increases, the beam error increases as well for fixed N. A more refined grid is needed as the magnitude of the parameter α increases.

Figure (24) shows that we need $N = 128$ to resolve the beam at the accuracy of 10^{-5} when $\alpha = 30$.

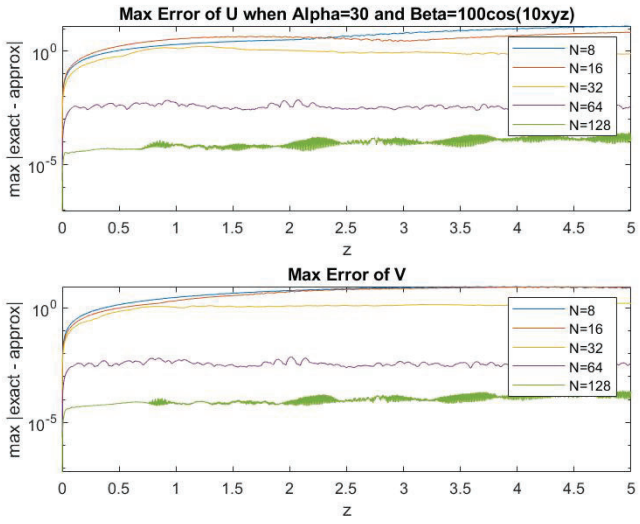

Figure 24: Method of Manufactured Solutions error plot for U and V where $\alpha = 30$ and $\beta = 100\cos(10xyz)$.

We showed with the Method of Manufactured solutions that $N = 128$ is accurate for our Gaussian beam problem with $w_0 = 0.03$, $F_0 = 500$, and $\lambda = 633 \times 10^{-9}$, where $\alpha = 35.7388$.

7 Lasers Beams in Nonhomogeneous Conditions

We proceed with the Fourier Collocation method to simulate laser beam propagation in nonhomogeneous medium. We solve a prototype problem for an arbitrary function to simulate as a precursor for future works on laser beams in nonhomogeneous media.

7.0.1 Prototype Problem

A prototype problem is considered to solve the nonhomogeneous PWE in equation (12) using the Fourier collocation method with an arbitrarily constructed refractive index term $\beta = 1000\cos(10xyz)$. The Figures below show the distortions of the beam as it propagates through space.

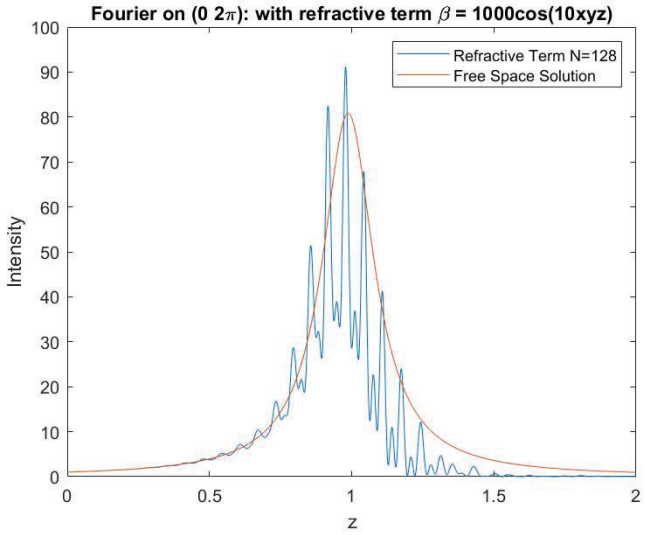

Figure 25: Intensity at the beam center varying along the z-axis. PWE with a varying refractive index $\beta = 1000\cos(10xyz)$ solved on the grid $[0,\ 2\pi]\times\ [0,\ 2\pi]$.

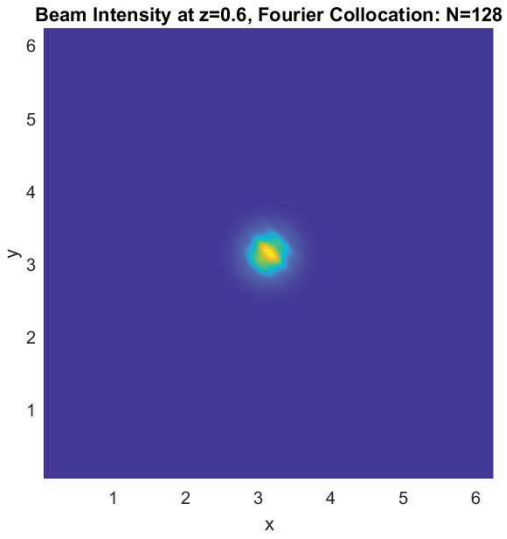

Figure 26: Intensity at z=0.6. PWE with a varying index $\beta = 1000\cos(10xyz)$ solved on the grid $[0,\ 2\pi]\times\ [0,\ 2\pi]$.

8 Conclusion

The goal of this study is to apply Spectral Methods to obtain approximate solutions to the Paraxial Wave Equation. We have successfully implemented a Galerkin method with a sine series expansion and two Collocation methods with a complex Fourier series expansion and a Chebyshev polynomial expansion. These methods were applied to find approximate solutions to an initial boundary value problem motivated by our desire to model how laser beams propagate in a turbulent medium. We first approached the homogeneous propagation problem where exact solutions in half-space are available. To test our numerical method, we designed our IBVP with initial conditions and boundary conditions on a large enough domain to sufficiently mimic the half space geometry for which we have exact solutions.

We worked with a non-dimensional version of the PWE where we rescaled the parameters intended to model laser beam propagation.The parameter w_0 is used to rescale the variables x and y in the cross section of the beam, while the radius of curvature of the initial phase F_0 is used to rescale z, which is the direction of beam propagation.The non-dimensional parameter α plays a key role in all of the numerical simulations and error analysis in our work, as described in the previous sections. The parameter α affects the shape of the functions that model the laser beams. Therefore, the series expansion approximations are affected.

To compare numerical methods, we standardized the Gaussian beam problem for propagation in a homogeneous medium ($\beta = 0$). We let the initial radius of curvature $F_0 = 500$ meters, the initial beam radius $w_0 = 0.03$ meters, and the laser's wavelength $\lambda = 633 \times 10^{-9}$ meters. These parameters results in the following non-dimensional constant: $\alpha = 35.7338$. Our numerical study found that to approximate the solution functions U and V to the accuracy of 10^{-4}, we need $N = 128$ modes for the series expansion in all three methods.

Our implementation of the Galerkin method is more computationally intensive than the Collocation methods. The extra computation expenditure results from resolving the initial expansion coefficients using numerical integration. MATLAB's built in numerical integration function *integral2* struggled with resolving these integrals for our specific Gaussian beam problem. The introduction of the linear refractive index term βu where $\beta \neq 0$ requires even more integrals to be resolved, which adds to the computational expenditure. In contrast, the collocation methods resolve these quadrature schemes analytically in the derivation of differentiation matrices. The resulting scheme runs more efficiently and quickly in MATLAB.

We first explored a Chebyshev polynomial expansion in our implementation of the Collocation method. The initial attempt used a square domain on $[-1, 1] \times [-1, 1]$. The beam on this domain resulted in significant boundary effects. The expansion of the domain to $[-2, 2] \times [-2, 2]$ resulted in more accurate results. However, MATLAB's ODE45 solvers struggled to solve the resulting ODEs for high N values.

Unlike the Chebyshev Collocation method, MATLAB's ODE solvers com-

puted the resulting ODEs for the Fourier Collocation method without a problem. The Fourier Collocation proved to be the most computationally robust out of the three Spectral methods. We proceeded with the Fourier collocation method to approximate solutions to the nonhomogeneous problem.

Because of the nature of the collocation methods, these methods easily apply to PDEs with non-constant coefficients as well to nonlinear PDEs. One of the areas of future work will be the task of implementing the two spectral methods to the PWE that governs thermal blooming, where a term proportional to $|u|^2u$ will replace the refractive index term we have encountered in our work. A second, and equally important, future work is to go beyond the range of Gaussian initial conditions and consider initial profiles that result in hybrid beams such as Gaussian-Hermite and Gaussian-Laguerre, and in general, beams that exhibit orbital angular momentum (OAM). One of the important challenges is to understand the impact of numerical discretization on the phase profiles of an OAM beam as a function of the non-dimensional parameter α described above.

8.1 ODE Solvers

More analysis and computation needs to be done to assess Spectral Methods for solving the PWE compared to conventional beam propagation methods. Within this study, we looked at spatial discretization and approximations to the Laplacian operators. The two parts of our numerical scheme are 1) the Spectral approximation in the x, y domain and 2) the time-integration method for propagating the beam along the z-axis, both of which induces their own approximation errors. This study looked at Spectral Method approximations using a Fourier Galerkin, Fourier Collocation, and Chebyshev Collocation approximation. MATLAB's ODE solvers were relied upon to propagate the beam forward.

Future research can focus on time integration methods and search for more optimal schemes to solve laser beam problems using Spectral methods. With more in depth study of time integration methods, more extensive error analysis can be performed, especially pertaining to the relationship between the spatial and temporal approximations. Additionally, boundary effects such as the one found in Figure (13) can be more accurately studied with a deeper understanding of the spatial-temporal errors.

9 Appendix A: Maxwell's Equations to Paraxial Wave Equation

The derivation of the Paraxial Wave Equation begins with Maxwell's equations, the fundamental model for classical electromagnetic theory. Then, Maxwell's equations are reduced to a form of the wave equation, which characterizes one component of the electromagnetic field. The wave equation reduces to the Helmholtz equation and from there, the Paraxial Wave Equation is derived.

Laser beams are a special form of monochromatic light. Like all light, laser beams fall into the broad spectrum of electromagnetic waves, which consists of an electric field component $\mathbf{E}(x, y, z, t)$ and a magnetic flux component $\mathbf{B}(x, y, z, t)$. These two time-dependent vector components form sinusoidal waves and travel transversely to each other while remaining spatially and temporally in phase. Electromagnetic waves propagate in the direction of the cross product $\mathbf{E} \times \mathbf{B}$, meaning they always travels perpendicular to each field component. The term Transverse Electronic Mode (TEM) within optics is often used to describe the phase front patterns of laser beams at points along the propagation axis [1].

Maxwell's equations provide a model for the movement of electromagnetic waves in terms of each individual component: $\mathbf{E}$ and $\mathbf{B}$. Maxwell's equations consist of Gauss's law for electricity, Gauss's law for magnetism, Faraday's law of induction, and Ampere's law. These equations are the starting point for deriving the model for laser beam propagation, the Paraxial Wave Equation. The differential form of simplified Maxwell's equations for a medium free of charge and absent of external current is given by

$$\nabla \cdot \mathbf{D} = 0 \tag{101}$$

$$\nabla \cdot \mathbf{B} \ = 0, \tag{102}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \tag{103}$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t}, \tag{104}$$

where $\mathbf{H} = \frac{1}{\mu}\mathbf{B}$ is the strength of the magnetic field and $\mathbf{D} = \varepsilon \boldsymbol{E}$ is the electric displacement. The parameter μ is the permeability and ε is the permittivity of the given medium. These parameters relate to the electric and magnetic fields respectively. These two parameters provide information on the speed v for electromagnetic waves in a given medium as given in the following relationship,

$$v^2 = \frac{1}{\mu\varepsilon}. \tag{105}$$

The refractive index n gives the ratio between the speed of a propagating electromagnetic wave v in a given medium and the speed of light c in a vacuum,

$$n = \frac{c}{v}. \tag{106}$$

The speed of light in a vacuum $c = \frac{1}{\sqrt{\mu_0 \varepsilon_0}}$ includes the vacuum constants for permittivity μ_0 and permeability ε_0. For electromagnetic waves in a non-vacuum environment, it is customary to let μ be constant and assume ε varies spatially and temporally through the propagating medium [cite Strohbehn].

To collapse Maxwell's coupled equation to a singular wave equation characterizing only the electric field component, the curl of Faraday's law of induction in (103) is taken,

$$\nabla \times \nabla \times \mathbf{E} = -\nabla \times (\frac{\partial \mathbf{B}}{\partial t}), \tag{107}$$

The vector identity is applied,

$$\nabla \times \nabla \times \mathbf{E} = \nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E}. \tag{108}$$

Ampere's law in (104) is differentiated with respect to time and the assumption that μ is constant is applied to find the relation,

$$\nabla \times (\frac{\partial \mathbf{B}}{\partial t}) = \mu \frac{\partial^2}{\partial t^2}(\varepsilon \mathbf{E}). \tag{109}$$

Applying (108) and (109) to (107), we get an equation in terms of only the electric field,

$$\nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E} = -\mu \frac{\partial^2}{\partial t^2}(\varepsilon \mathbf{E}). \tag{110}$$

From Gauss's law for electricity in (101) in a media free of charge, we find the relationship,

$$\mathbf{E} \cdot (\nabla \varepsilon) + \varepsilon(\nabla \cdot \mathbf{E}) = 0 \implies (\nabla \cdot \mathbf{E}) = -\frac{\mathbf{E} \cdot (\nabla \varepsilon)}{\varepsilon} = -\mathbf{E} \cdot \nabla \ln(\varepsilon). \tag{111}$$

Using the chain rule, we have $\nabla \ln(\varepsilon) = \frac{1}{\varepsilon} \nabla \epsilon$. Substituting $\nabla \cdot \mathbf{E}$ from equation (111) into equation (110) gives

$$\nabla^2 \mathbf{E} - \mu \frac{\partial^2}{\partial t^2}(\varepsilon \mathbf{E}) = -\nabla[\mathbf{E} \cdot \nabla(\ln \varepsilon)]. \tag{112}$$

If ε is a constant, (112) reduces to the standard wave equation,

$$\nabla^2 \mathbf{E} - \frac{1}{v^2} \frac{\partial^2}{\partial t^2}(\mathbf{E}) = 0 \tag{113}$$

However, the permittivity varies for an inhomogeneous medium, so a varying refractive index can be rewritten in terms of permittivity and permeability,

$$n^2(x, y, z, t) = \frac{\mu \varepsilon(x, y, z, t)}{\mu_0 \varepsilon_0}. \tag{114}$$

We assume $\mu = \mu_0$, which leaves the relationship,

$$\varepsilon(x, y, z, t) = \varepsilon_0 n^2(x, y, z, t). \tag{115}$$

Substituting (115) converts equation (112) in terms of the speed of light and a varying refractive index,

$$\nabla^2\mathbf{E} - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}(n^2\mathbf{E}) = -2\nabla(\mathbf{E}\cdot\nabla(\ln n)). \tag{116}$$

It is customary in the field of optics [19] to rewrite (116) as,

$$\nabla^2\mathbf{E} - \frac{n^2}{c^2}\frac{\partial^2}{\partial t^2}(\mathbf{E}) = 0. \tag{117}$$

The simplification of moving n^2 outside of the partial time derivative results from differences in the order of magnitude of the temporal variations in the refractive index and the period of the electric field. Significant variations in the refractive index occur within the order of 10^{-5} seconds, which is large compared to the magnitude of the oscillation period of light: 10^{-14} seconds or faster [19].

The second approximation by dropping the right hand side of (116) takes into account the scales of refractive index changes within a medium [15]. The following approximation can be made:

$$-2\nabla(\mathbf{E}\cdot\nabla(\ln n)) = \nabla(\frac{1}{n^2}\nabla n^2) \approx 0. \tag{118}$$

If we assume the electric field is polarized in the x-direction, we can isolate the E_x component of the field. The vector equation reduces to a scalar wave equation. For the case of laser beams, we assume the light is monochromatic, which is characterized by a single wavelength λ_w, resulting in a constant angular frequency $\omega = \frac{kc}{n_0}$ where $k = \frac{2\pi n_0}{\lambda_w}$ is the wave number. Therefore, time dependence can be separated in form,

$$E_x(x,y,z,t) = \psi(x,y,z)e^{-i\omega t}. \tag{119}$$

From (117), we get

$$\Delta\psi + \frac{n^2\omega^2}{c^2}\psi = 0 \tag{120}$$

The rapid oscillations of the wave in the z direction is separated to produce a slowly varying wave u,

$$\psi(x,y,z) = u(x,y,z)e^{ikz}. \tag{121}$$

The resulting equation from (120) becomes

$$\Delta u + 2ik\frac{\partial u}{\partial z} - k^2u + \frac{n^2\omega^2}{c^2}u = 0. \tag{122}$$

Finally, we get the PWE by converting parameters and applying the Paraxial Approximation, $|\frac{\partial^2 u}{\partial z^2}| << |k\frac{\partial u}{\partial z}|$,

$$\Delta_\perp u + 2ik\frac{\partial u}{\partial z} + k^2(\frac{n_r^2}{n_0^2} - 1)u = 0. \tag{123}$$

The typical magnitude of k for visible light is 10^7. The paraxial approximation is discussed more depth in [11].

10 Appendix B: MATLAB Script for 1D Galerkin Method

ODESys function for ODE45.

```
function dadt=ODESys(t,a,B,c,L,N,lambda)
    %Equation compiling: *note that a_m=a(2m-1), a'(2m-1)=a(2m), and a_m''=a'(2m) dadt=z
    for i=1:N
        dadt(2*i-1)=a(2*i);
        dadt(2*i)=-(c^2*lambda(i)*a(2*i-1)+sum(B(i,1:1:N)*a(1:2:2*N)));
    end
end
```

MATLAB Script

```
clear all;
tic
c=1;L=1;N=10;
%Inner Product Coefficient B_{m,n}=<f(x)*phi_n,phi_m>
syms x
f(x)=sin(x) - 5*sin(30*x) + 10*sin(40*x); % defines f(x)
lambda=(1:N).^2*pi^2/L^2; % eigenvalue lambda_m
%Eigenfunction phi_m(x,m)=2/L*m^2*pi^2/L^2
B=zeros(N);
for i=1:N
    for j=1:N
        B(i,j)=4/L^2*int(f(x)*sin(i*pi*x/L)*sin(j*pi*x/L),x,0,L);
    end
end
%%
%initial conditions
u0(x)=exp(-200*(x-0.5)^2); %u(x,0)=u0
u1(x)=diff(u0(x));  %u_t(x,0)=u1
aic=2/L*int(u0(x)*sin(pi*(1:N)*x/L),x,0,L); %a(0)
aicp=2/L*int(u1(x)*sin(pi*(1:N)*x/L),x,0,L); %a'(0)
for i=1:1:N
    a0(2*i-1)=aic(i);
    a0(2*i)=aicp(i);
end
a0=double(a0);
tspan=[0 4]; %time interval
[t,a]=ode45(@(t,a) ODESys(t,a,B,c,L,N,lambda), tspan,a0);%ODE solver

%%
%Solution u=phi_m(x)*a_m(t)
x=linspace(0,L,100);
for m=1:N;
```

```
phi(m,:)=sin(m*pi*x/L);
end
tbegin=1; tend=length(t); dt=10;
u=zeros(N,length(x));
tp=tbegin:dt:tend;i=1;
for ti=tbegin:dt:tend
    for k=1:N
        u(k,:)=phi(k,:)*a(ti,2*k-1);
    end
    u=sum(u);
    up(i,:)=u;
    i=i+1;
    %hold on
    %plot(x,u);
end
waterfall(x,t(tp),up)
view([10,80]); title('1D Wave Equation Problem')
ylabel('0<t<'+string(t(end))), zlabel('u'),xlabel('0<x<'+string(L));
```

11 Appendix C: MATLAB Script for the Chebyshev Collocation Method for the PWE

```
clear all
tic
%PDE Parameters
w0=0.03; F0=500; wavelength=633E-9; k=2.*pi/wavelength;
Alpha=2*k*w0^2/F0; %Constant
Beta= @(x,y,z) 0;  %Refractive index function

a1=-2; b1=2; %Set domain of chebyshev grid [a1 b1] x [a1 b1]
shiftx=0; shifty=0; %Set Coordinates of Beam Center

%Initial Conditions
Uinitial=@(x,y) exp(-((x-shiftx).^2+(y-shifty).^2))...
    .*cos(Alpha/4*((x-shiftx).^2+(y-shifty).^2));
Vinitial=@(x,y) -exp(-((x-shiftx).^2+(y-shifty).^2))...
    .*sin(Alpha/4*((x-shiftx).^2+(y-shifty).^2));

%%
%Differentiation Matrix
N=64;
zsteps=linspace(0,2,1E4+1); %Timestep Evaluations
zi=1:length(zsteps); %Timestepping index

[D,x] = cheb2(N,a1,b1); y=x;
D2=D^2;D2interior=D2(2:end-1,2:end-1);
[X,Y]=meshgrid(x(2:end-1),y(2:end-1));
Xi=X(:); Yi=Y(:);  NN=N-1;
u0=Uinitial(X,Y); v0=Vinitial(X,Y);
a0=[u0(:); v0(:)];

%Solve System of ODEs
[z,a]=ode45(@(z,a) Spectral2DODESysGRI(a,N,D2interior,X,Y,z,Alpha,Beta),zsteps,a0);
U=a(:,1:NN^2); V=a(:,NN^2+1:end); %Reshape Solution

TIME=toc

%%

%Exact Solution in Free Space
w0s=w0; F0s=500; alpha0=2/(k*w0s^2)+i/F0s;
ue=@(x,y,z) 1./(1+i*alpha0*(z*F0s))...
    .*exp(-1/2*(alpha0*k./(1+i*alpha0.*(z*F0s))).*((x*w0s).^2+(y*w0s).^2));
```

```
%Plot 2D Intensity Figures
p=1;
for ti=[1 3E3+1 5E3+1 7E3+1] %Propagation Path Indices
   ur=reshape(U(ti,:),NN,NN);
   vr=reshape(V(ti,:),NN,NN);
   I=ur.^2+vr.^2;
   Ie=abs(ue(X,Y,z(ti))).^2;

%    ErrorI(p)= max(max(abs(I-Ie))); %Max I error Along Propagation Axis
%    ErrorU(p)= max(max(abs(ur-real(ue(X,Y,z(ti)))))); %Max U Error
%    ErrorV(p)= max(max(abs(vr-imag(ue(X,Y,z(ti)))))); %Max V Error

   %Plot
   figure; surf(X,Y,I)
   view(0,90); shading interp
   xlim([a1 b1]); ylim([a1 b1])
   xlabel('x'); ylabel('y'); zlabel('Intensity');
   title('Beam Intensity at z=' +string(z(ti))+...
                   ', Chebyshev Collocation: N='+string(N))

   p=p+1;
end




%%
%Plot intensity at the beam center along propagation axis
Iaxis=U(1:end,(end+1)/2).^2+V(1:end,(end+1)/2).^2;
Ieaxis=abs(ue(0,0,z(1:length(Iaxis)))).^2;
figure
plot(z,Iaxis)
hold on
plot(z,Ieaxis)
legend('Approximation N=' +string(N), 'Free Space Solution')
title('Chebyshev Collocation Method on ['+string(a1)+' '+string(b1)+ ']')
ylabel('Intensity'); xlabel('z')



%%
%FUNCTIONS

%Laplacian Approximation and ODEs
    function dadt=Spectral2DODESysGRI(a,N,D2interior,X,Y,z,Alpha,Beta);
```

```
   NN=N-1;
   u=a(1:NN^2);v=a(NN^2+1:end);
   umatrix=reshape(u,NN,NN);vmatrix=reshape(v,NN,NN);
   Lu=D2interior*umatrix+umatrix*D2interior';
   Lv=D2interior*vmatrix+vmatrix*D2interior';

   dudz=(-Lv-Beta(X,Y,z).*vmatrix)./Alpha;
   dvdz=(Lu+Beta(X,Y,z).*umatrix)./Alpha;
   dadt=[dudz(:); dvdz(:)];
   end


%Cheb Function from Trefethen, modified for arbitrary square [a,b]x[a,b]
   function [D,x] = cheb2(N,a,b)
     if N==0, D=0; x=1; return, end
     index=(0:N)';
     x=((b-a)/2)*cos((2*index+1)*pi/(2*(N+1)))+(a+b)/2;
     c = [2; ones(N-1,1); 2].*(-1).^(0:N)';
     X = repmat(x,1,N+1);
     dX = X-X';
     D  = (c*(1./c)')./(dX+(eye(N+1)));      % off-diagonal entries
     D  = D - diag(sum(D'));                 % diagonal entries
   end
```

12 Appendix D: MATLAB Script for the Fourier Collocation Method for the PWE

```
    clear all
tic
%PDE Parameters
w0=0.03; F0=500; wavelength=633E-9; k=2.*pi/wavelength;
Alpha=2*k*w0^2/F0; %Constant
Beta= @(x,y,z) 0;  %Refractive index function

shiftx=pi; shifty=pi; %Set Coordinates of Beam Center

%Initial Conditions
Uinitial=@(x,y) exp(-((x-shiftx).^2+(y-shifty).^2))...
    .*cos(Alpha/4*((x-shiftx).^2+(y-shifty).^2));
Vinitial=@(x,y) -exp(-((x-shiftx).^2+(y-shifty).^2))...
    .*sin(Alpha/4*((x-shiftx).^2+(y-shifty).^2));

%%
%Differentiation Matrix
N=64;
zsteps=linspace(0,2,1E4+1); %Timestep Evaluations
zi=1:length(zsteps); %Timestepping index
x=linspace(0,2*pi,N+1); y=x;
j=1:length(x)-1; h=2*pi/(N);
r1=-pi^2/(3*h^2)-1/6;
r2=-(-1).^(j)./(2*sin(j*h/2).^2);
D2=toeplitz([r1 r2]);

D2interior=D2(2:end-1,2:end-1);
[X,Y]=meshgrid(x(2:end-1),y(2:end-1));
Xi=X(:); Yi=Y(:);  NN=N-1;
u0=Uinitial(X,Y); v0=Vinitial(X,Y);
a0=[u0(:); v0(:)];


%Solve System of ODEs
[z,a]=ode45(@(z,a) Spectral2DODESysGRI(a,N,D2interior,X,Y,z,Alpha,Beta),zsteps,a0);
U=a(:,1:NN^2); V=a(:,NN^2+1:end); %Reshape Solution

TIME=toc

%%

%Exact Solution in Free Space
```

```
w0s=w0; F0s=500; alpha0=2/(k*w0s^2)+i/F0s;
ue=@(x,y,z) 1./(1+i*alpha0*(z*F0s))...
    .*exp(-1/2*(alpha0*k./(1+i*alpha0.*(z*F0s))).*((x*w0s).^2+(y*w0s).^2));


%Plot 2D Intensity Figures
p=1; Xs=X-shiftx; Ys=Y-shifty;
for ti=[1 3E3+1 5E3+1 7E3+1] %Propagation Path Indices
   ur=reshape(U(ti,:),NN,NN);
   vr=reshape(V(ti,:),NN,NN);
   I=ur.^2+vr.^2;
   Ie=abs(ue(Xs,Ys,z(ti))).^2;

%    ErrorI(p)= max(max(abs(I-Ie))); %Max I error Along Propagation Axis
%    ErrorU(p)= max(max(abs(ur-real(ue(X,Y,z(ti)))))); %Max U Error
%    ErrorV(p)= max(max(abs(vr-imag(ue(X,Y,z(ti)))))); %Max V Error

   %Plot
   figure; surf(Xs,Ys,I)
   view(0,90); shading interp
   xlim([a1 b1]); ylim([a1 b1])
   xlabel('x'); ylabel('y'); zlabel('Intensity');
   title('Beam Intensity at z=' +string(z(ti))+...
                   ', Chebyshev Collocation: N='+string(N))

   p=p+1;
end


%%
%Plot intensity at the beam center along propagation axis
Iaxis=U(1:end,(end+1)/2).^2+V(1:end,(end+1)/2).^2;
Ieaxis=abs(ue(0,0,z(1:length(Iaxis)))).^2;
figure
plot(z,Iaxis)
hold on
plot(z,Ieaxis)
legend('Approximation N=' +string(N), 'Free Space Solution')
title('Fourier Collocation Method on [0 2\pi]')
ylabel('Intensity'); xlabel('z')



%%
%FUNCTION
%Laplacian Approximation and ODEs
```

```
function dadt=Spectral2DODESysGRI(a,N,D2interior,X,Y,z,Alpha,Beta);
NN=N-1;
u=a(1:NN^2);v=a(NN^2+1:end);
umatrix=reshape(u,NN,NN);vmatrix=reshape(v,NN,NN);
Lu=D2interior*umatrix+umatrix*D2interior';
Lv=D2interior*vmatrix+vmatrix*D2interior';

dudz=(-Lv-Beta(X,Y,z).*vmatrix)./Alpha;
dvdz=(Lu+Beta(X,Y,z).*umatrix)./Alpha;
dadt=[dudz(:); dvdz(:)];
end
```

13 Appendix E: MATLAB Script for the Fourier Galerkin Method for the PWE

```
    clear all;
tic
%parameters
L=4; N=32;
w0=0.03; F0=500; wavelength=633E-9; k=2.*pi/wavelength;
alpha=2*k*w0^2/F0; %Constant

%%
%eigenvalue lambda_{m,n} indexed as lambda(m,n)
lambda=((1:N).^2)*pi^2/L^2.*ones(N)+((1:N).^2)'*pi^2/L^2.*ones(N);
%%
%Initial Conditions
shiftx=L/2; shifty=L/2; %Set Coordinates of Beam Center
Uinitial=@(x,y)  exp(-((x-shiftx).^2+(y-shifty).^2))...
    .*cos(alpha/4*((x-shiftx).^2+(y-shifty).^2));
Vinitial=@(x,y) -exp(-((x-shiftx).^2+(y-shifty).^2))...
    .*sin(alpha/4*((x-shiftx).^2+(y-shifty).^2));

%Array Value is not available for integral2 function.
%for loop is used instead
%2*N^2 number of integrals computed (can be computed in parallel)
%Integrals are the most computationally intensive
IC=zeros(2*N^2,1);
phi=@(x,y,m,n) sin(n.*x.*pi/L).*sin(m.*pi.*y/L); %Basis Functions
for m=1:N
    for n=1:N
        IC(2*N*(m-1)+2*n-1)=integral2(@(x,y) Uinitial(x,y)...
            .*phi(x,y,m,n),0,L,0,L)*4/L^2;
        IC(2*N*(m-1)+2*n)= integral2(@(x,y) Vinitial(x,y)...
            .*phi(x,y,m,n),0,L,0,L)*4/L^2;
    end
end
TIME(1)=toc;
%%
%Time stepper
tspan=linspace(0,2,1E4+1); %timestep
[t,a]=ode45(@(t,a) ODEPWESys(t,a,alpha,L,N,lambda), tspan,IC); %ODE solver
TIME(2)=toc
ac=reshape(a(1:2:end),N,N,length(tspan));
bc=reshape(a(2:2:end),N,N,length(tspan));
```

```
%%
%Plot Solution

[X,Y]=meshgrid(linspace(0,L,100));
tbegin=1; tend= length(t); dt=100;
ki=1;

for ti=[1 3E3+1 5E3+1 7E3+1]%tbegin:dt:tend
V1=0; V2=0; %initialize
    for m=1:N
        for n=1:N
            V1old=V1;
            V1=a(ti,2.*N.*(m-1)+2.*n-1).*sin(n.*pi.*X/L).*sin(m.*pi.*Y/L);
            V1=V1old+V1;
            V2old=V2;
            V2=a(ti,2.*N.*(m-1)+2.*n).*sin(n.*pi.*X/L).*sin(m.*pi.*Y/L);
            V2=V2old+V2;
        end
    end

    I=V1.^2+V2.^2;
    V1center(ki)=V1(50,50);
    V2center(ki)=V2(50,50);
    Icenter(ki)=I(50,50); ki=ki+1;

    while ki<10
    figure; surf(X,Y,I);
    title('Beam Intensity at z=' +string(t(ti))+...
                    ', Fourier Galerkin: N='+string(N))
    xlabel('x'); ylabel('y'); axis square; shading interp; view(2)
    pause(0.01)
    end

end


%%
% %Compare how integrals resolved the IC
% V1IC=max(max((V1-Uinitial(X,Y))))
% V2IC=max(max((V2-Vinitial(X,Y))))

%%
%Exact Solution Gaussian beam in Free Space
w0s=w0; F0s=500; alpha0=2/(k*w0s^2)+i/F0s;
ue=@(x,y,z) 1./(1+i*alpha0*(z*F0s))...
    .*exp(-1/2*(alpha0*k./(1+i*alpha0.*(z*F0s))).*((x*w0s).^2+(y*w0s).^2));
```

```
%%
function dadt=ODEPWESys(t,a,alpha,L,N,lambda)
%Equation compiling: *note that a_{m,n}=(2*N*(m-1)+2n-1) and b_{m,n}=(2*N*(m-1)+2n)
dadt=zeros(2*N^2,1);

%Fourier Coefficients
ac=reshape(a(1:2:end),N,N); %a coefficient
bc=reshape(a(2:2:end),N,N);  %b coefficient

%ODEs
dacdt=lambda.*bc/alpha;
dbcdt=-lambda.*ac/alpha;
dadt(1:2:2*N^2)=dacdt(:);
dadt(2:2:2*N^2)=dbcdt(:);
end
```

14 Appendix F: MATLAB Script for the Exact Solution plot of the Gaussian Beam

The Matlab script for plotting the exact solution of the Gaussian beam is presented below:

```
clear all;
%Beam Parameters
w0=0.03; F0=500; lambda=633E-9; k=2*pi/lambda;
alpha0=2/(k*w0^2)+i/F0;
L1=-2*w0; L2=-L1; N=500;

%Coordinates and Frame
z=1000;
x=linspace(L1,L2,N); y=x;
[X,Y]=meshgrid(x,y);
r=sqrt(X.^2+Y.^2);

%Solution
u=1/(1+i*alpha0*z).*exp(-1/2*(alpha0*k/(1+i*alpha0*z)).*r.^2);
I=abs(u).^2;

%Plot
mesh(X,Y,I);
title('Guassian Beam Intensity: F_0='+string(F0)+', \lambda='...
    +string(lambda)+ ', w_0='+string(w0)+ ', z=' +string(z))
xlabel('x'); ylabel('y'); zlabel('I')
```

```
axis square; view(2); colorbar;
```

References

[1] Andrews, Larry C. and Phillips, Roland L. *Laser Beam Propagation through Random Media,* 2005, SPIE, San Francisco, California.

[2] Bradie, Brian. *A Friendly Introduction to Numerical Analysis*, 2006, Pearson, Upper Saddle River, New Jersey.

[3] Carminati, Remi and Schotland, John. *Principles of Scattering and Transport of Light*. 2021. Cambridge University Press. Cambridge, UK.

[4] Canuto, C, Hussaini, MY, Quarteroni, A, and Zang TA. *Spectral Methods in Fluid Dynamics*, 1988, Springer-Verlag, Heidelberg, Germany.

[5] Department of Defense. *Department of Defense Strategic Spectrum Plan*, 2008, Department of Defense.

[6] Gbur, Gregory, *Singular Optics*, CRC Press, Boca Raton, Florida.

[7] Gottlieb, David and Orszag, Steven. *Numerical Analysis of Spectral Methods: Theory and Applications*, 1977, SIAM, Philadelphia.

[8] Hanna, Ray and Rowland, John. *Fourier Series, Transforms, and Boundary Value Problems: Second Edition*, 1990, Dover Publications, Mineola, New York.

[9] Hesthaven, Jan, Gottlieb, Sigal, Gottlieb, David. *Spectral Methods for Time-Dependent Problems*, 2007, Cambridge University Press, New York.

[10] Judd KP, Avramov-Zamurovic, S, Matt S, Handler, RA, Watnik, AT, Lindle, JR, Esposito, J, Jarret, WA. *Propagation of laser beams carrying orbital angular momentum through simulated optical turbulence in Rayleigh-Benard convection*, 2021, SPIE.

[11] Lax, Melvin, Louisell, William H., and McKnight, William B. *From Maxwell to paraxial wave optics*. 1975. Physical Review A. America Physical Association.

[12] Nielsen, Phillip D. *Effects of Directed Energy Weapons*, 1994, United States Air Force.

[13] Peyret, Roger. *Spectral Methods for Incompressible Viscous Flow*. 2010 Springer-Verlag, New York, New York.

[14] Shampine, Lawrence and Reichelt, Mark. *The Matlab ODE Suite*, SIAM Journal on Scientific Computing, Vol 18, 1997, pp. 1-22.

[15] Strohbehn, J. W. *Laser Beam Propagation in the Atmosphere*, 1978, Springer, Berlin Germany.

[16] Trefethen, Lloyld N. *Spectral Methods in MATLAB,* 2000, SIAM, Philadelphia, PA.

[17] US Navy, *Naval Research Enterprise Addendum to the Naval Research and Development Framework*. 2017. US Navy.

[18] Vasudeva, Harkrishan, *Elements of Hilbert Spaces and Operator Theory*, 2017, Springer, Singapore.

[19] Walsh, J.L. and Ulrich P.B., *Thermal Blooming in the atmosphere,* 1978, Springer, New York.

[20] Wang, Meng, Mani, Ali, and Gardeyev, Stanislav. *Physics and Computations of Aero-Optics*, 2012, Annual Review of Fluid Mechanics.

[21] White, Micheal D. *High-order parabolic beam approximation for aero-optics*, 2010, Ohio Aerospace Institute, Wright Patterson Air Force Base, OH.