Optical Communication Networks

Notes & Tutorials

Andrew Simon Wilson

12^{th} July, 2022

UNIVERSITÀ DEGLI STUDI DI BRESCIA

EMIMEO Programme

Module Coordinator: Prof. Fabio Baronio

Author Details

Andrew Simon Wilson, BEng

Post-graduate Master's Student - EMIMEO Programme

andrew.s.wilson@protonmail.com andrew-simon-wilson AS-Wilson +44 7930 560 383

Contents

Introduction

I wrote this document for the students studying Optical Communication Networks to have a nice set of notes, and correct reference code and graphs for the module. I hope that it is sufficient for this task and it helps all of your studies.
I spent have spent a lot of time developing the template used to make this LaTeX document, I want others to benefit from this work so the source code for this template is available on GitHub [?].

1 Introduction, The NLSE, and Optical Fibre Propagation

1.1 Introduction

The main goal of this class is the investigation of the evolution of the optical pulses which propagate in an optical fibre.

We will consider the propagation of pulses from two perspectives:

1. Theoretically
2. Via numerical simulation, using software such as MATLAB, Python, or C/C++

We will analyse and understand a number of different optical effects and regimes / models. These include:

- Group Velocity Dispersion (GVD, this is a linear effect)
- Self-Phase Modulation (this is a non-linear effect)
- Optical, Self-Trapped Waves (Solitons)
- Abnormal, Extreme Waves
- Optical Shocks

The assessment for the course will consist of a piece of coursework on the numerical dynamics of optical pulses propagating under different linear and non-linear regimes. After this coursework is handed in, a type of oral examination will be performed with each student about their coursework.
Each of you is invited to work in groups of 2-3 persons, each with different jobs.

Finally, should the lectures leave you with any confusion, or you wish to study further, the suggested textbook for this course to aid in study (if this is needed) is Non-linear Fibre Optics by Govind P. Agrawal [?].

1.2 The Non Linear Schrödinger Equation (3+3D)

The first step to considering the propagation of optical pules in a fibre is to know that an optical fibre is a non-linear and dispersive medium and that any propagation with this waveguide is governed by the fundamental and universal modal of optical wave dynamics; the Non Linear Schrödinger Equation (NLSE).

In Optical Communication Components Prof. Constantino De Angelis will analytically study the properties of the NLSE. This course, however, will consider different regimes from theoretical viewpoints, as well as numerical simulation of each of these regimes to understand the linear and non-liner effects and their uses.

So, without further ado, the Non Linear Schrödinger Equation (NLSE) (3+3D) is given by Equation **??**:

$$j\frac{\partial A(r,t)}{\partial z} + \frac{1}{2\beta}\frac{\partial^2 A(r,t)}{\partial x^2} + \frac{1}{2\beta}\frac{\partial^2 A(r,t)}{\partial y^2} - \frac{\beta''}{2}\frac{\partial^2 A(r,t)}{\partial t^2} + \chi^{(3)}|A(r,t)|^2 A(r,t) = 0 \tag{1}$$

Where:

$r = (x, y, z)$, this is the 3-dimensional spatial coordinates

t, is the time coordinate

$A(r, t)$, is the slowly varying (compared to the carrier signal) envelope of the signal

$E(r, t) = Re[A(r, t)e^{i(\omega_0 t + \beta_0 z)}]$, is the electrical field of the pulse and the carrier signal

$e^{i(\omega_0 t + \beta_0 z)}$, is the optical carrier signal at the angular frequency ω_0 and 'wavenumber' β_0

1.2.1 A Quick Detour - The Gaussian Pulse

The type of communication signal model that we will deal with most often initially in the course is that of the Pulsed Gaussian or Gaussian Pulse. This signal is the combination of a lower frequency Gaussian (this is where the information is really communicated) and a higher frequency carrier sine/cosine component, due to how high the frequency of the carrier typically is (10THz) it cannot be easily sampled, so we instead send information via the slowly varying envelope. To give an idea of what the this model looks like these individual signals are shown separately in Figure **??**.

Please note that we will give the Gaussian as a function most simply described by Equation **??**:

$$f(t) = Ie^{-\frac{t^2}{2t_0^2}} \tag{2}$$

Also note t_0 is expressed here as the half-waist duration at an intensity of $\frac{1}{e}$, there are other definitions which are useful for other purposes

If one has a linear system (or one that can be approximated as such a type of system) these two signals can be super-positioned to create a Gaussian pulse, which is shown in Figure **??**

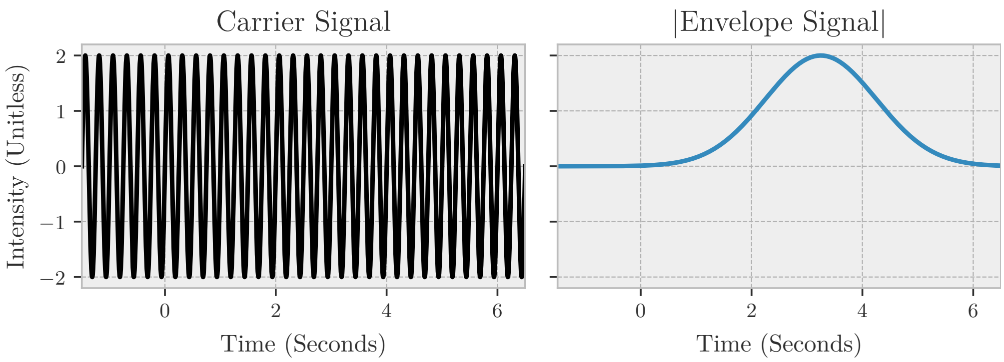

Figure 1: An example graph demonstrating the de-constructed elements of a Gaussian pulse, the carrier signal is on the left and the positive portion of Gaussian on the right

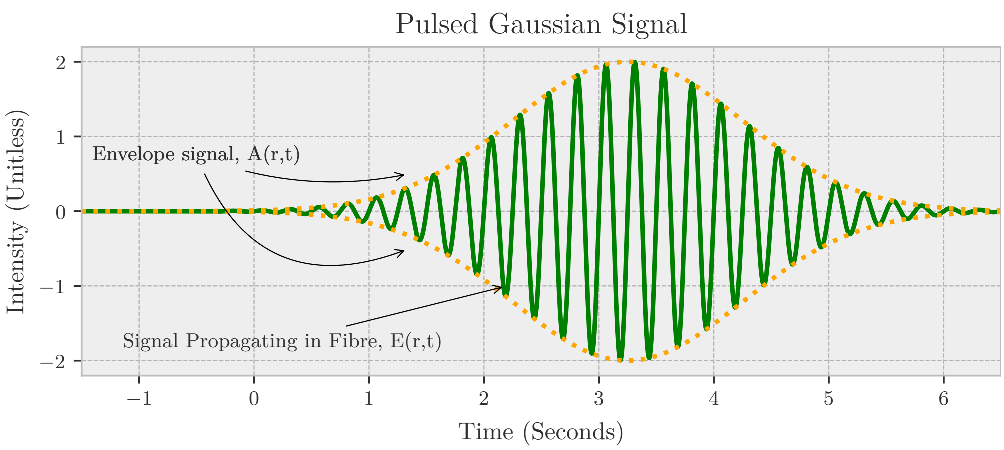

Figure 2: An example graph of the Gaussian Pulse, the envelope signal is shown as a dotted line, the signal propagated in the medium is the solid line.

It is also important to note here that the envelope signal is itself comprised of two components, a positive and a negative portion, this is easily displayed in the frequency domain, observe the spectrum (Figure **??**) of the propagating signal (displayed before in Figure **??**), there are a positive and negative frequency components:

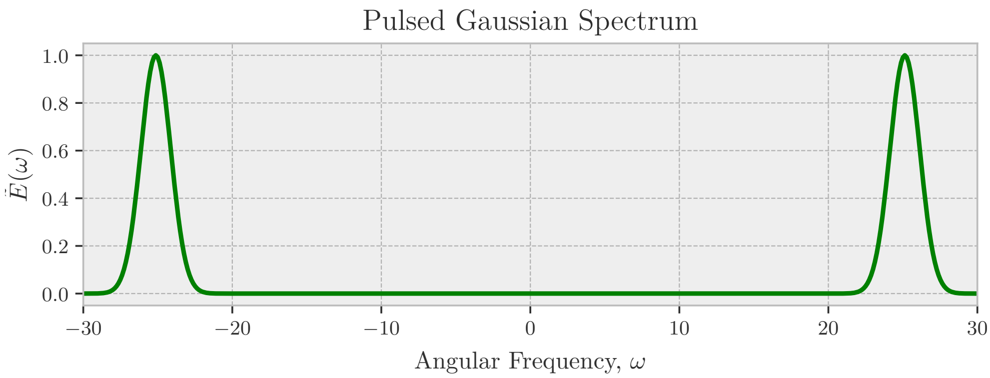

Figure 3: The Gaussian Pulse spectrum

Note that the two Gaussians are centred at the carrier frequency and that the waist frequency interval at 1/e intensity ($\Delta\omega$) must be significantly smaller than the carrier frequency of the signal (ω_0), i.e. $\Delta\omega \ll \omega_0$. This is shown in the normalised frequency spectrum graph, Figure **??**:

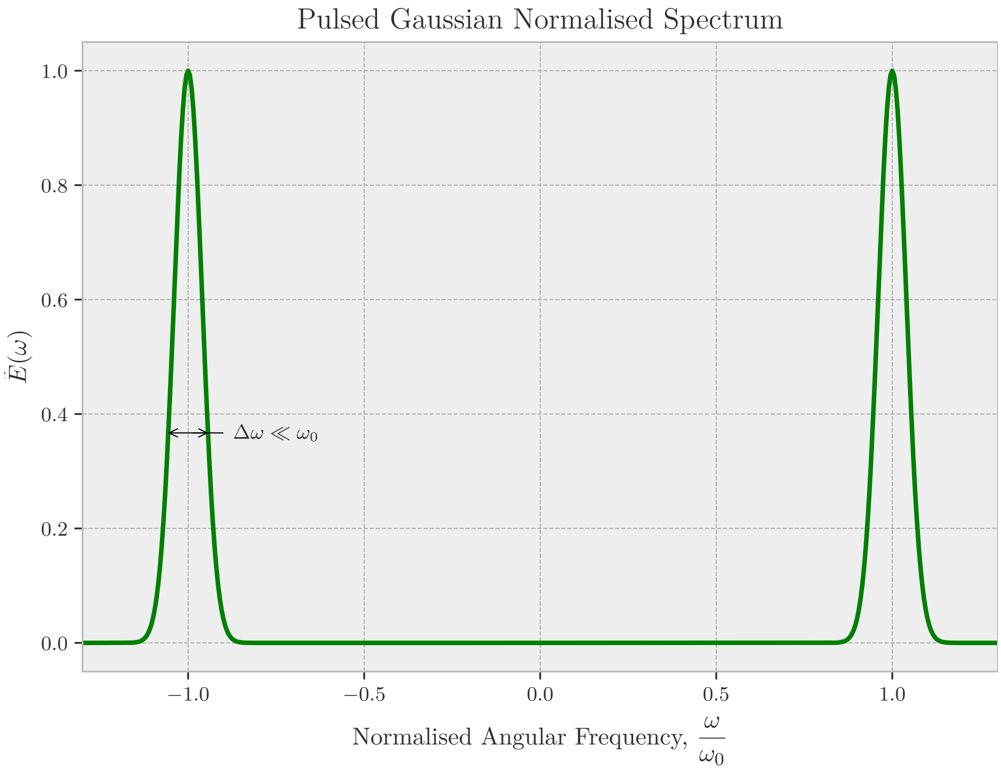

Figure 4: The Gaussian Pulse on the normalised frequency spectrum

1.2.2 The Components of the NLSE

The various components of the NLSE shown before (in Equation **??** and repeated below for completeness), can be summarised as follows:

$$j\frac{\partial A(r,t)}{\partial z}+\frac{1}{2\beta}\frac{\partial^2 A(r,t)}{\partial x^2}+\frac{1}{2\beta}\frac{\partial^2 A(r,t)}{\partial y^2}-\frac{\beta''}{2}\frac{\partial^2 A(r,t)}{\partial t^2}+\chi^{(3)}|A(r,t)|^2A(r,t)=0$$

$j\frac{\partial A(r,t)}{\partial z}$	This takes into account the propagation along the z-axis
$\frac{1}{2\beta}\frac{\partial^2 A(r,t)}{\partial x^2}$	Is the dispersion term along the x-axis
$\frac{1}{2\beta}\frac{\partial^2 A(r,t)}{\partial y^2}$	Is the dispersion term along the y-axis
$-\frac{\beta''}{2}\frac{\partial^2 A(r,t)}{\partial t^2}$	Is the dispersion term along the t-axis
$\chi^{(3)}\|A(r,t)\|^2A(r,t)$	Is the non-linear term
$\beta=\frac{\omega}{c}n(\omega)$	Is the propagation constant
β''	Is the group velocity dispersion
$\chi^{(3)} \quad n=\sqrt{1+\chi^{(\omega)}}$	Accounts for non-linear cubic response of the medium

In general, the NLSE is a (3+1) Dimension (z,y,z,t) model ((3+1)D model) which describes the evolution of an optical light pulse in space and time. In optical fibres, the envelope of the pulse, $A(r,t)$ can be expressed (using separation of variables) as shown below in Equation **??**:

$$A(r,t)=A(x,y,z,t)=F(z,t)\cdot M(x,y)\cdot e^{i\delta\beta z} \tag{3}$$

Where:

$M(z,y)$, defines the mode profiles

$\partial\beta$, is the correction of the propagation constant $\implies \beta=\beta_0+\partial\beta$

$\therefore M(z,y)\cdot e^{i\delta\beta z}$, describes the modal distribution in the x-y plane at a given z, and;

$F(z,t)$, describes the slowly varying (compared to the carrier) pulse envelope in the z-t plane

1.3 Spatio-Temporal Regime, the NLSE (1+1D)

The first regime we will focus on is the spatio-temporal evolution of $F(z,t)$, ignoring other sections of our pulse. From the NLSE (3+1D) one can derive a model for $F(z,t)$ to obtain the NLSE (1+1D), given by Equation **??**:

$$j\frac{\partial F(z,t)}{\partial z} - \frac{\beta''}{2}\frac{\partial^2 F(z,t)}{\partial t^2} + \gamma|F(z,t)|^2 F(z,t) = 0 \tag{4}$$

Where:

z, is the propagation or evolution coordinate

t, is the temporal coordinate

β'', is the Group Velocity Dispersion (GVD) inside the fibre

γ, is the effective non-linear term, related to $\chi^{(3)}$

It should be noted that both the (3+1D) and (1+1D) NLSEs are non-linear, partial differential equations that do not trivially lend themselves to analytical solutions, except in some very specific cases. It is more likely that one could successfully perform an analytical study of the (1+1D) NLSE as compared to the (3+1D) NLSE for obvious reasons, however it is even more practical to take a numerical approach to each of these equations to understand their linear and non-linear effects in optical fibres.

A large number of specific numerical methods may be performed, but they all fall into two general families:

- Finite-Difference methods
- Pseudo-Spectral methods (these are generally faster by an order of magnitude whilst achieving the same accuracy as the above methods)

One pseudo-spectral method which has been used extensively to solve pulse propagation in non-linear, dispersive media is the **split-step Fourier method** (also known as the Beam Propagation Method - BPM), the relative speed of this method when compared with finite-difference schemes can be attributed to the Fast Fourier Transform (FFT) algorithm.

2 Numerical Techniques

As mentioned in the previous section one of the most practical and efficient methods to solve pulse propagation in optical fibres, and the one we will heavily focus on in the course, is the split-step Fourier method. To understand the methodology of, and use, the split-step Fourier method one must write the NLSE in the form shown below in Equation **??**:

$$\frac{\partial F(z,t)}{\partial z} = (\tilde{D} + \tilde{N})F \tag{5}$$

$\tilde{D}$ is a differential operator that accounts for dispersion in a linear medium (however it can account for absorption and/or higher order dispersions) and is given by Equation **??**:

$$\tilde{D} = -j\frac{\beta''}{2}\frac{\partial^2}{\partial t^2} \tag{6}$$

$\tilde{N}$ is a non-linear operator that governs the effect of fibre non-linearity on pulse propagation, e.g. Kerr non-linearity (but it can account for other non-linear phenomena like Raman effects). $\tilde{N}$ is given by Equation **??**:

$$\tilde{N} = -j\gamma|F|^2 \tag{7}$$

In general the dispersion, $\tilde{D}$, and non-linearity, $\tilde{N}$, act together along the fibre, but the Split-Step Fourier method obtains an **approximate solution** by assuming that by propagating the envelope over a small distance, h, the dispersive and non-linear effects can be considered to act independently.
More specifically, the propagation from a point, z, to another, $z+h$, can be described via two steps of calculation (the astute reader will note this is why the method is called split-step):

1. Calculating the propagation accounting only for non-linear effects, $\tilde{D} = 0$
2. Calculating the propagation accounting only for dispersive effects, $\tilde{N} = 0$

Mathematically this process is described by Equation **??**:

$$F(z+h,t) \approx e^{h\tilde{D}} \cdot e^{h\tilde{N}} \cdot F(z,t) \tag{8}$$

Later we will show that the term $e^{h\tilde{D}}$ can be evaluated in the Fourier domain using Equation **??**:

$$e^{h\tilde{D}} \cdot F(z,t) = \mathcal{FT}^{-1}[e^{h\tilde{D}(j\omega)}] \quad \mathcal{FT}[F(z,t)] \tag{9}$$

For full clarity (to avoid any confusion with notation used) $\mathcal{FT}$ denotes the Fourier Transform, and $\tilde{D}(j\omega)$ is obtained by replacing the differential operators in time domain; $\frac{\partial}{\partial t} \rightarrow j\omega$ and $\frac{\partial^2}{\partial^2 t} \rightarrow (j\omega)^2 = -\omega^2$. As $\tilde{D}(j\omega)$ is a number in Fourier space, the evaluation of the term mathematically, and numerically with the FFT, is fast and simple.

To estimate the accuracy of the split-step method, one must observe the formal solution of the NLSE compared to the approximate solution we have formed, the formal solution is given by Equation **??** (assuming $\tilde{N}$ is z-independent):

$$F(z+h,t) = e^{h(\tilde{D}+\tilde{N})}F(z,t) \tag{10}$$

The comparison of this equation and the approximate solution leads us to a topic of mathematics you may not be familiar with, the Baker-Campbell-Hausdorff proof;

2.1 The Baker-Campbell-Hausdorff Formula

Now is a good time to recall the Baker-Campbell-Hausdorff formula [?] which is a solution for an equation such as that shown below in Equation **??**:

$$e^{\tilde{a}}e^{\tilde{b}} = e^{\tilde{c}} \tag{11}$$

The solution to which is given by Equation **??**:

$$e^{\tilde{c}} = e^{\tilde{a}+\tilde{b}+\frac{1}{2}[\tilde{a},\tilde{b}]+\frac{1}{12}[\tilde{a},[\tilde{a},\tilde{b}]]-\frac{1}{12}[\tilde{b},[\tilde{a},\tilde{b}]]+\dots} \tag{12}$$

Where:

$$[\tilde{a},\tilde{b}] = \tilde{a}\tilde{b} - \tilde{b}\tilde{a}$$

And:

If $[\tilde{a},\tilde{b}] = 0$ the operators commutate

If $[\tilde{a},\tilde{b}] \neq 0$ the operators do not commutate

Comparing Equations **??** and **??** exactly indicates that the Split-Step method ignores the non-commutating nature of the $\tilde{D}$ and $\tilde{N}$ operators. However, if we cheat a bit and insert $\tilde{a} = h\tilde{D}$ and $\tilde{b} = h\tilde{N}$ into Equation **??** and compare it to Equation **??** the third exponent term will be $\frac{1}{2}[\tilde{a},\tilde{b}] = \frac{1}{2}h^2[\tilde{D},\tilde{N}]$. Thus, the Split-Step Fourier method is accurate to the second order when using the step size of h (i.e. the equations match until the third exponential term and thus can be said to be approximate to a certain degree), one must emphasise what this signifies: **as long as h is small the errors of calculation will also be small (insignificant).**

2.2 Numeric Implementation

To summarise the previous section, the general implementation of the Split-Step Fourier method is given by Equation **??**:

$$F(z+h,t) = e^{h(\tilde{D}+\tilde{N})}F(z,t) \approx e^{h\tilde{D}} \cdot e^{h\tilde{N}} \cdot F(z,t) \tag{13}$$

And thus, to simulate the propagation from z to $z+h$ we will follow the previously set out steps, repeated below for completeness (this could be completed in reverse order just as effectively):

1. Calculating the propagation accounting only for non-linear effects, $\tilde{D} = 0$
2. Calculating the propagation accounting only for dispersive effects, $\tilde{N} = 0$

2.2.1 Considerations for Numeric Implementation

For the envelope $F(z,t)$, we must take into account that it's profile, for numerical applications, will be recorded in a matrix (or vector), via appropriate sampling of the signal, F, e.g. Shannon-Nyquist Sampling Theorem [**?**] [**?**]. This is displayed below in Figure **??**, the dotted line displays the discrete sampling of the signal and the solid line the continuous signal that exists in reality:

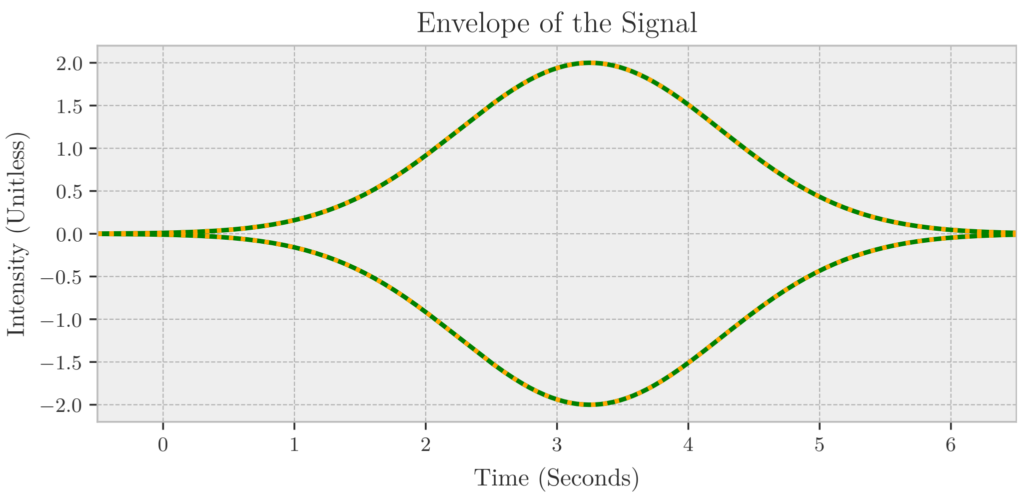

Figure 5: The difference between the continuous signal and discretely measured data-points of the signal

TODO: something about having to use multiple matrices

2.2.2 Dispersive Step

To solve for the dispersive step of the equation, we have to solve Equation ??:

$$\frac{\partial F(z,t)}{\partial z} = -j\frac{\beta''}{2}\frac{\partial^2}{\partial t^2} \tag{14}$$

To do this we must consider the Fourier transform of $F(z,t)$, given by Equation ??:

$$\tilde{F}(z,\omega) = \int_{-\infty}^{+\infty} F(z,t)e^{-j\omega}dt \tag{15}$$

We are interested in change with propagation, so computing this integral produces and differentiating gives:

$$\frac{\partial \tilde{F}(z,\omega)}{\partial z} = (+j\frac{\omega^2}{2}\beta'')\tilde{F}(z,\omega)$$

And finally to form a general solution to calculate the new signal value after each dispersive portion of the step, h, the necessary calculations before completing the non-linear portion are shown in frequency and time domain as Equations ?? and ??:

$$\tilde{F}(z+h,\omega) = \tilde{F}(z,\omega)e^{+j\frac{\omega^2}{2}\beta''\cdot h} \tag{16}$$

$$F(z+h,t) = \mathcal{FT}^{-1}\left[\tilde{F}(z+h,\omega)\right] \tag{17}$$

2.2.3 Non-Linear Step

For the non-linear portion of our regime we begin with Equation ??:

$$\frac{\partial F(z,t)}{\partial z} = j\gamma|F|^2F \tag{18}$$

Since one can demonstrate that $|F|^2$ is z-independent, the final solution is given by Equation ??:

$$F(z+h,t) = F(z,t)e^{j\gamma|F(z,t)|^2\cdot h} \tag{19}$$

Recall that h will be our step size in the z-axis. Also note this is no th

2.3 Summary of the Scheme for Numeric Algorithm

The summary of the steps we will take in order to complete the numeric implementation of everything we have covered in this section is as follows:

1. Initialisation (of variables, matrices, and any necessary functions or classes)
2. Definition of:
   - Temporal and Spatial Co-Ordinates
   - Frequency Domain Co-Ordinates
   - Quantities relevant to our signal (intensities, envelope definition, etc.)
3. Perform the Split-Step Fourier Method:
   - The Dispersive Step
   - The Non-Linear Step
   - Save the data for each z step at each time index
4. Display the results appropriately

2.4 Numeric Implementation in Python

The code can be found below in Section **??**, it has been purposefully made more difficult to copy and paste so that you will, at the very least, type the code out manually and perhaps absorb some of the essential information regarding the implementation of the Split-Step Fourier Method.

This is only introductory code, so we aren't actually going to look at any propagation that involves dispersion or non-linearity, but the two relevant terms, β'' and γ, can be changed to observe this (in the next sections this is what will be done in order to produce the graphs to observe the various effects).

The first graph shown below, Figure **??**, displays the input envelope and final, output envelope in the time domain. In this case it is a simple Gaussian signal with a half-waist of 1 second and an absolute intensity of 1.

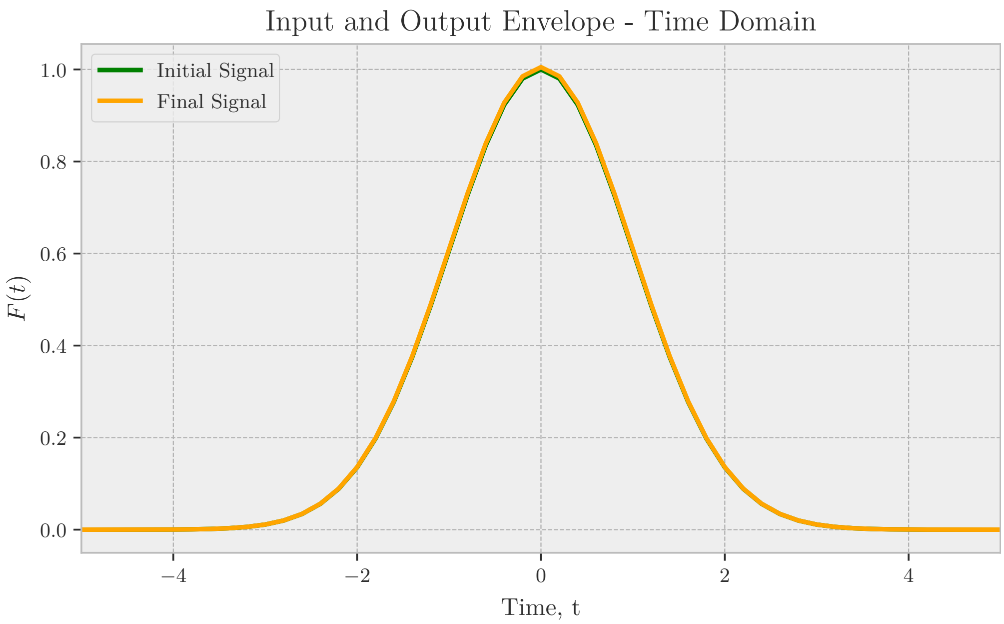

Figure 6: Split-Step Fourier Method Introduction - 2D Graph of the Initial and Final Envelope in the Time Domain

The next important graph, Figure **??**, is the input and output envelopes in the frequency domain.

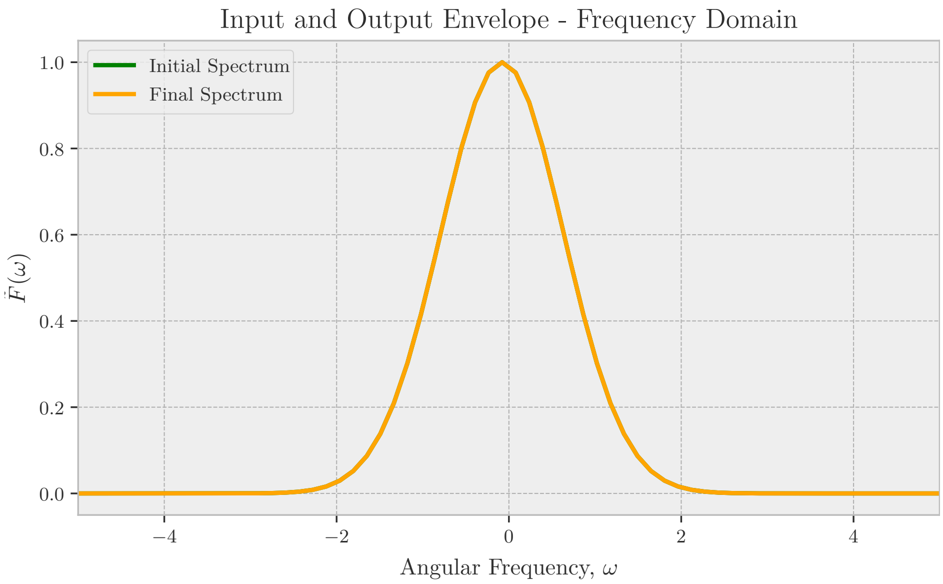

Figure 7: Split-Step Fourier Method Introduction - 2D Graph of the Initial and Final Envelope in the Frequency Domain

Then finally we have the two 3D plots of the propagation in both the time domain and frequency domain, Figures **??** and **??**:

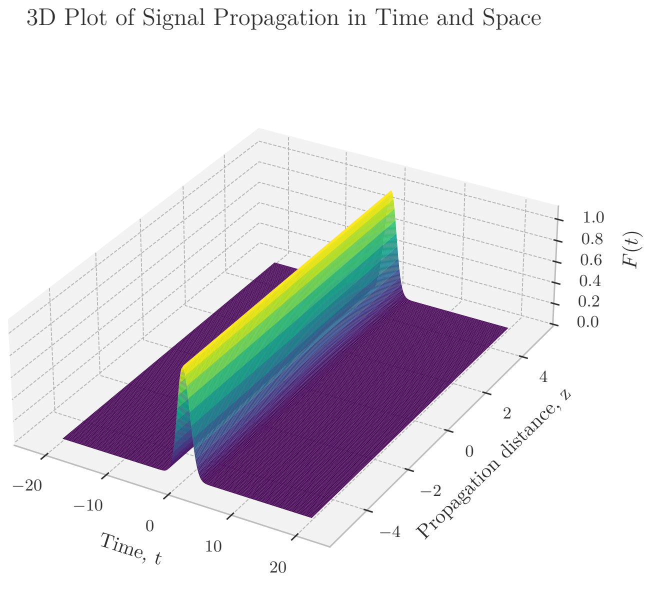

Figure 8: Split-Step Fourier Method Introduction - 3D Propagation Graph Envelope in the Time Domain

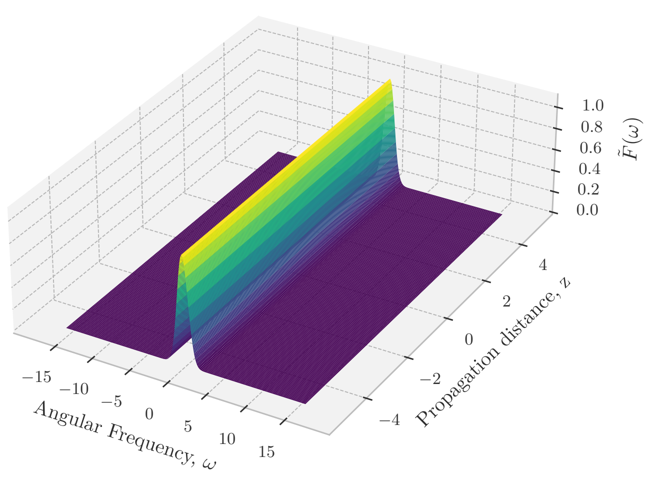

Figure 9: Split-Step Fourier Method Introduction - 3D Propagation Graph Envelope in the Frequency Domain

2.5 Split-Step Fourier Method in Python - Introduction

```
###########################################################################################################################
##### THIS IS NECESSARY, THIS GIVES US ALL THE FUNCTIONS WE NEED FOR NUMERICAL IMPLEMENTATION #####

import numpy as np
import matplotlib.pyplot as plt
from scipy.fftpack import fft, fftshift, ifft, ifftshift
from mpl_toolkits.mplot3d import Axes3D


###########################################################################################################################
##### IGNORE THIS IF YOU WISH, THIS IS SIMPLY TO MAKE PRETTY GRAPHS #####

##### matplotlib graph settings #####
# Produce all graphs as PGFs so that they can be imported natively to latex and still be edited if needed
plt.rcParams["pgf.texsystem"] = "pdflatex"
# Setting fonts for all the graphs etc.
plt.rcParams["text.usetex"] = True
plt.rcParams["font.family"] = "serif"
plt.rcParams["font.serif"] = "ptserif"
plt.rcParams["font.size"] = 9

# Match the colouring if my Latex docs
COLOUR = '#2E2E2E'
plt.rcParams['text.color'] = COLOUR
plt.rcParams['axes.labelcolor'] = COLOUR
plt.rcParams['xtick.color'] = COLOUR
plt.rcParams['ytick.color'] = COLOUR
plt.rcParams['pgf.preamble'] = r"\usepackage[T1]{fontenc} \usepackage{mathpazo}"

# Set style to bmh and place the ticks for the axes on the outside to be readable
plt.style.use('bmh')
plt.rcParams['xtick.direction'] = 'out'
plt.rcParams['ytick.direction'] = 'out'




###########################################################################################################################
##### THE CODE TO IMPLEMENT THE SPLIT-STEP FOURIER METHOD #####

## The Fibre Material Properties
beta_2 = 0 #
gamma = 0 #


## Temporal Co-Ordinate Creation
```

```
# The higher the number of samples for a given signal duration, the better the curved gaussian shape will be after the
#   FFT:
no_of_samples = 201
signal_duration = 40

# Create an array of time values with X number of equally-spaced values, a larger array size for a given signal
#   duration makes for a better frequency domain transform
time = np.linspace(-signal_duration/2, signal_duration/2, no_of_samples)

# This is the period between each discrete sample of the signal in the time domain
sample_period = signal_duration / (no_of_samples - 1)



## Spatial (z-axis) Co-Ordinate Creation
signal_propagation_distance = 10

# Create an array of z-axis values with X number of equally-spaced values, a larger array size for a given signal
#   duration makes for a better frequency domain transform
z_axis = np.linspace(-signal_propagation_distance/2, signal_propagation_distance/2, no_of_samples)

# This is the distance between each discrete sample of the signal on the z-axis
z_step_distance = signal_propagation_distance / (no_of_samples - 1)



## Frequency Domain Co-Ordinate Creation
# Angular frequency
# This is complicated but the angular frequency interval (distance between each measured discrete frequency of
#   the FFT) is given by this calculation
ang_freq_interval = (2.0 * np.pi) / signal_duration
# From the above we can then create a frequency axis with defined frequency points based on the number of
#   samples, this will be explained in the notes
ang_freq_axis = np.arange(-no_of_samples/2, no_of_samples/2) * ang_freq_interval



## Input Envelope Definition
intensity = 1 # Signal intensity
init_waist = 1 # Used to define the Gaussian signal, this is the half-waist size at 1/e intensity

# The Gaussian (envelope) signal
envelope = intensity * np.exp(-((time ** 2) / (2 * (init_waist ** 2))))
envelope = np.array(envelope)


## Input Spectrum Calculations
# First the raw intensity FFT transform, that will be arranged in order from -ve frequency to +ve frequency

# The FFT algorithm in python is missing a component (sample_period/sqrt(2*pi)) see:
#   https://cvarin.github.io/CSci-Survival-Guide/fft.html
envelope_raw_spectrum = fftshift(fft(envelope)) * sample_period/np.sqrt(2*np.pi)
envelope_raw_spectrum = np.array(envelope_raw_spectrum)

# Then one calculates the absolute intensity (|E(omega)|^2) for the input spectrum
envelope_abs_spectrum = abs(envelope_raw_spectrum)**2



## Split-Step Fourier Method
# Factors for calculating the evolution of the signal
dispersive_terms = [np.exp(complex(0, (((ang_freq_axis[0] ** 2) / 2) * beta_2 * z_step_distance)))]

for i in range(1, no_of_samples):
    dispersive_terms.append(np.exp(complex(0, (((ang_freq_axis[i] ** 2) / 2) * beta_2 * z_step_distance))))


 # Since gamma is 0 in this code, this will be an array with no value, but the code is still useful
non_linear_term = gamma * z_step_distance



envelope_complex = [complex(envelope[0], 0)]

for i in range(1, no_of_samples):
    envelope_complex.append(complex(envelope[i], 0))

pulse_output_time = np.r_[envelope_complex]
pulse_output_spectrum = np.r_[envelope_raw_spectrum]


# Create vectors to manipulate when carrying out our steps
current_step_spectrum = envelope_raw_spectrum

```

```python
# The actual Split-Step Fourier Method Calculations
for x in range(1, no_of_samples):

    for j in range(no_of_samples):
        # Dispersive (Linear) Step
        current_step_spectrum[j] = current_step_spectrum[j] * dispersive_terms[j]

    # Prep for Non-Linear Step
    current_step_time = ifft(ifftshift(current_step_spectrum)) * ang_freq_interval / np.sqrt(2.0*np.pi) * no_of_samples

    for j in range(no_of_samples):
        # Non-Linear Step
        current_step_time[j] = current_step_time[j] * np.exp(complex(0, (non_linear_term *
                                                                    (abs(current_step_time[j])**2))))

    pulse_output_time = np.row_stack((pulse_output_time, current_step_time))
    pulse_output_spectrum = np.row_stack((pulse_output_spectrum, current_step_spectrum))




##########################################################################################################################
########## GRAPHING ##########

## ENVELOPES IN TIME DOMAIN
plt.figure()

# Plot the input envelope vs time
plt.plot(time, envelope, color='green', label='Initial Signal')

# Plot the output envelope vs time
plt.plot(time, abs(pulse_output_time[-1, :]), color='orange', label='Final Signal')

# Titles and labels
plt.title(r'Input and Output Envelope - Time Domain')
plt.ylabel(r'$F(t)$')
plt.xlabel(r'Time, t')
plt.legend(loc="upper left")

# Set limits
plt.xlim(-5, 5)

# Aspect Ratio
axes = plt.gca()
axes.set_aspect(5)

plt.tight_layout(pad=1.2) # Place everything slightly closer together

# Save graph in this location
plt.savefig('../Graphs/split-step-intro-2d-time.pgf', bbox_inches = 'tight', pad_inches = 0)



## ENVELOPES IN FREQUENCY DOMAIN
plt.figure()

# Plot the input envelope intensity vs angular frequency
plt.plot(ang_freq_axis, envelope_abs_spectrum, color='green', label='Initial Spectrum')

# Plot the output envelope vs time
plt.plot(ang_freq_axis, abs(pulse_output_spectrum[-1, :])**2, color='orange', label='Final Spectrum')

# Titles and labels
plt.title(r'Input and Output Envelope - Frequency Domain')
plt.ylabel(r'$\tilde{F}(\omega)$')
plt.xlabel(r'Angular Frequency, $\omega$')
plt.legend(loc="upper left")


# Set limits
plt.xlim(-5, 5)

# Aspect Ratio
axes = plt.gca()
axes.set_aspect(5)

plt.tight_layout(pad=1.2) # Place everything slightly closer together

# Save graph in this location
plt.savefig('../Graphs/split-step-intro-2d-freq.pgf', bbox_inches = 'tight', pad_inches = 0)
```

```
## 3D TIME DOMAIN GRAPH
# Create a mesh matrix to plot signal values against
T, Z = np.meshgrid(time, z_axis)

# Create figure and axis objects
fig = plt.figure()
ax = plt.axes(projection='3d')

# Plot the signal as it propagates
ax.plot_surface(T, Z, abs(pulse_output_time), rstride=1, cstride=1, cmap='viridis', edgecolor='none')

# Colouring the background of the graph
fig.patch.set_facecolor('white')
ax.set_facecolor('white')
ax.w_xaxis.set_pane_color((0.95, 0.95, 0.95, 0.95))
ax.w_yaxis.set_pane_color((0.95, 0.95, 0.95, 0.95))
ax.w_zaxis.set_pane_color((0.95, 0.95, 0.95, 0.95))

# Title and axis labels
ax.set_title(r'3D Plot of Signal Propagation in Time and Space')
ax.set_xlabel(r'Time, t')
ax.set_ylabel(r'Propagation distance, z')
ax.set_zlabel(r'$F(t)$')


# Aspect ratio settings
ax.auto_scale_xyz([-22, 22], [-5, 5], [0, 1.1])
ax.set_box_aspect((1.5, 2.0, 0.6))

plt.tight_layout(pad=0.8) # Place everything slightly closer together

# Save graph in this location
plt.savefig('../Graphs/split-step-intro-3d-time.pgf', bbox_inches = 'tight', pad_inches = 0)



## 3D FREQUENCY DOMAIN GRAPH
# Create a mesh matrix to plot signal values against
S, Zs = np.meshgrid(ang_freq_axis, z_axis)

# Create figure and axis objects
fig = plt.figure()
ax = plt.axes(projection='3d')

# Plot the signal as it propagates
ax.plot_surface(S, Zs, abs(pulse_output_spectrum)**2, rstride=1, cstride=1, cmap='viridis', edgecolor='none')

# Colouring the background of the graph
fig.patch.set_facecolor('white')
ax.set_facecolor('white')
ax.w_xaxis.set_pane_color((0.95, 0.95, 0.95, 0.95))
ax.w_yaxis.set_pane_color((0.95, 0.95, 0.95, 0.95))
ax.w_zaxis.set_pane_color((0.95, 0.95, 0.95, 0.95))

# Title and axis labels
ax.set_title(r'3D Plot of Signal Propagation in Freq. and Space')
ax.set_xlabel(r'Angular Frequency, $\omega$')
ax.set_ylabel(r'Propagation distance, z')
ax.set_zlabel(r'$\tilde{F}(\omega)$')

# Aspect ratio settings
ax.auto_scale_xyz([-18, 18], [-5, 5], [0, 1.1])
ax.set_box_aspect((1.5, 2.0, 0.6))

plt.tight_layout(pad=0.8) # Place everything slightly closer together

# Save graph in this location
plt.savefig('../Graphs/split-step-intro-3d-freq.pgf', bbox_inches = 'tight', pad_inches = 0)

# Display a window of each graph now that we're finished
plt.show()
```

3 Group Velocity Dispersion, GVD

Also known as Dispersion Induced Pulse Broadening, the effects of GVD on optical pulses propagating in an optical fibre are studied by setting the non-linear term to zero, $\gamma = 0$, $F(z,t)$ then satisfies Equation **??**:

$$j\frac{\partial F(z,t)}{\partial z} \approx \frac{\beta_2}{2}\frac{\partial^2 F}{\partial t^2} \tag{20}$$

Be careful, $\beta_2 \neq \beta''$

Equation **??** is trivially solved using the Fourier Transform, if $\tilde{F}(z,\omega$ is the signal $F(z,t)$ in frequency domain, such that it satisfies Equation **??**, then it also satisfies Equation **??**:

$$F(z,t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \tilde{F}(z,\omega)e^{-j\omega t}d\omega \tag{21}$$

$$j\frac{\partial F(z,t)}{\partial z} = \frac{1}{2}\beta_2\omega^2\tilde{F} \tag{22}$$

The solution to this ODE is given by Equation **??**, it is clear from this solution that each spectral component of the signal will have it's phase changed differently for a given propagation distance, z, and frequency, ω.

$$\tilde{F}(z,\omega) = \tilde{F}(z=0,\omega)e^{j\frac{1}{2}\beta_2\omega^2 z} \tag{23}$$

If one substitutes Equation **??** into Equation **??** the general solution to Equation **??** is obtained, shown below in Equation **??**:

$$F(z,t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\tilde{0},\omega)e^{-j\frac{1}{2}\beta_2\omega^2 z - j\omega t}d\omega \tag{24}$$

Where $F(\tilde{0},\omega)$ is the Fourier Transform of the input signal.

This Equation (**??**) can be used for input functions of almost any shape, making this equation rather powerful.

3.1 GVD with a Gaussian Pulse Input - Analytical Solution

We will first analytically consider the propagation of a Gaussian Pulse with GVD, the Gaussian equation was given in Section **??** as Equation **??**, repeated again the Gaussian is defined here as:

$$f(t) = Ie^{-\frac{t^2}{2t_0^2}}$$

Where t_0 is the half-width at $\frac{1}{e}$ intensity, we could also use the Full Width at Half Maximum (FWHM), we can define for the FWHM:

$$T_{FWHM} = 1.665t_0 = 2\sqrt{ln(2)}t_0$$

Now, if we combine Equations ?? and ?? then perform the integration, we will obtain Equation ??:

$$F(z,t) = \frac{t_0}{(t_0^2 - j\beta_2 z)^{\frac{1}{2}}} e^{-\frac{t^2}{2(t_0^2 - j\beta_2 z)}} \tag{25}$$

This equation displays that a Gaussian signal will clearly maintain it's shape upon propagation, however it's intensity will lessen, and it's width in time will increase, we can define the equation for the time domain change after a given propagation distance as Equation ??:

$$T_1(z) = t_0(1 + (\frac{z}{L_D})^2)^{\frac{1}{2}} \tag{26}$$

With $L_D = \frac{T_0^2}{|\beta_2|}$ and it is usually called the "Dispersion Length". This value defines how, for a given fibre length, shorter pulses will broaden more than longer ones as the dispersion length will be larger. In other words the dispersion length governs the extent of broadening. At a propagation distance equal to the dispersion length, $z = L_D$, a Gaussian will broaden by $\sqrt{2}$.

Comparing our Gaussian input, and calculated output after propagation (Equations ?? and ??) indicates that even though our input is un-chirped (with no phase modulation), we have a chirped output, this is probably easier to observe if we define our propagation in different terms, shown it Equations ?? and ??:

$$F(z,t) = |F(z,t)|e^{j\phi(z,t)} \tag{27}$$

$$\phi(z,t) = \frac{-sin(\beta_2)(\frac{z}{L_D})}{1 + (\frac{z}{L_D})^2} \frac{t^2}{t_0^2} + \frac{1}{2} tan^{-2}(\frac{z}{L_D}) \tag{28}$$

As the phase changes with respect to time, the change in the frequency domain of the signal due to dispersion after propagation will be different for a given frequency. This difference can be obtained by differentiating with respect to time, given in Equation ??, $\partial\omega$ is the chirp and clearly depends on the sign and value of β_2:

$$\partial\omega(t) = -\frac{\partial\phi(z,t)}{\partial t} = \frac{-sin(\beta_2)(\frac{2z}{L_D})t}{1 + (\frac{z}{L_D})^2 t_0^2} \tag{29}$$

Dispersion-Induced pulse broadening is an important phenomena to understand, there is also some unmentioned (thus far) terminology attached to it; when the dispersive term is greater than zero, $\beta_2 > 0$, long wavelengths (e.g. red light) travel faster than short wavelengths (e.g. blue light) and we term this type of dispersion normal, whilst the opposite ($\beta_2 < 0$) is termed anomalous dispersion.

3.2 GVD with a Gaussian Pulse Input - Numerical Simulation

3.3 GVD with a Chirped Gaussian Pulse Input - Analytical Solution

3.4 GVD with a Chirped Gaussian Pulse Input - Numerical Simulation

3.5 GVD with Hyperbolic-Secant Pulse Input - Analytical Solution

3.6 GVD with Hyperbolic-Secant Pulse Input - Numerical Simulation

3.7 GVD Induced Limitations - Dispersion Management

4 Self Phase Modulation, SPM

4.1 Changes in Pulse Spectra due to SPM

5 Competition of SPM and GVD