

A New Method of Modelling Tunable Lasers with Functional Composition

by

Brady Metherall

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in
The Faculty of Science
Modelling and Computational Science

University of Ontario Institute of Technology

May 2019



© Brady Metherall 2019

Abstract

A new nonlinear model is proposed for tuneable lasers. Using the generalized nonlinear Schrödinger equation as a starting point, expressions for the transformations undergone by the pulse are derived for each component within the laser cavity. These transformations are then composed to give the overall effect of one trip around the cavity. The linear version of this model is solved analytically, and the nonlinear version numerically. A consequence of the nonlinear nature of this model is that it is able to exhibit wave breaking which prior models could not. We highlight the rich structure of the boundary of stability for a particular plane of the parameter space.

Acknowledgements

First and most importantly, I would like to thank my supervisor and mentor, Dr. Sean Bohun, for his assistance, guidance, and inspiration over the past two years. I would also like to thank my fellow modelling and computational science students for helping make the last two years enjoyable.

Lastly, I would like to thank my mom and dad, and grandparents for their endless support and encouragement over the past two years.

Author's Declaration

I declare that the work in this thesis was carried out in accordance with the regulations of the University of Ontario Institute of Technology. The work is original except where indicated by special reference in the text and no part of the dissertation has been submitted for any other degree. Any views expressed in the dissertation are those of the author and in no way represent those of the University of Ontario Institute of Technology. This thesis has not been presented to any other university for examination either in Canada or elsewhere.

Brady Metheraall
Friday 31st May, 2019

Contents

Abstract	ii
Acknowledgements	iii
Author's Declaration	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Tuneable Lasers	3
1.1.1 Optical Coupler and Laser Output	4
1.1.2 Modulator	4
1.1.3 Fibre Bragg Grating	5
1.1.4 Optical Circulator	10
1.1.5 Optical Amplifier and Pump Laser	10
1.2 Mode-Locking	11
2 Previous Modelling Efforts	13
2.1 Generalized Nonlinear Schrödinger Equation	13
2.2 The Master Equation of Mode-Locking	15
2.3 Discrete Functional Models	17
3 A New Model	19
3.1 Components	19
3.1.1 Gain	20
3.1.2 Fibre Nonlinearity	21
3.1.3 Loss	22
3.1.4 Dispersion	22
3.1.5 Modulation	23
3.2 Span of Gaussians in $L^2(\mathbb{R})$	24
3.3 Non-Dimensionalization	28
3.4 Combining the Effects	30

3.5	Solution to the Linear Model	30
3.5.1	Spread Due to Dispersion	32
3.5.2	Equilibrium Shape	34
3.5.3	Asymptotic Expansion of the Variance	37
3.5.4	Equilibrium Energy	41
4	Solution of the Nonlinear Model	44
4.1	Code	44
4.1.1	Validation	45
4.2	Nonlinear Model	47
4.2.1	Energy	48
4.2.2	Convergence	53
4.2.3	Permutation of Components	57
5	Conclusion	60
References		62
A The Lambert W Function		69
B Code		72

List of Figures

1.1	Comparison of pumping, spontaneous emission, and stimulated emission.	2
1.2	Typical cavity of a tuneable laser.	3
1.3	Manufacture methods of fibre Bragg gratings.	6
1.4	Comparison of a chirped and unchirped Gaussian pulse.	7
1.5	Chirped fibre Bragg grating.	9
1.6	Symbol for a four port optical circulator.	10
1.7	Two examples of mode-locking.	12
3.1	Two examples of Gaussian series.	26
3.2	Gaussian series for a particular modulation function.	28
3.3	Envelope, Fourier transform, and chirp of the pulse—linear case.	36
3.4	Asymptotic expansion of the variance as $s \rightarrow 0$.	38
3.5	Asymptotic expansion of the variance as $s \rightarrow \infty$.	41
4.1	Equilibrium variance, chirp, and phase shift of the pulse as a function of s .	45
4.2	Equilibrium energy, and peak power of the pulse as a function of s .	46
4.3	Envelope, Fourier transform, and chirp of the pulse—stable case.	47
4.4	Energy of the pulse at equilibrium.	49
4.5	Stability of the pulse in the s - b plane.	50
4.6	Approximate effective nonlinearity.	50
4.7	Envelope, Fourier transform, and chirp of the pulse—unstable case.	52
4.8	Error of the pulse.	54
4.9	Error for various values of Δ .	55
4.10	Composite error.	56
4.11	Energy of the pulse at equilibrium with the modulation and dispersion blocks switched.	58
A.1	The two branches of the Lambert W function.	70

List of Tables

3.1 Orders of magnitude of various parameters. 29

Introduction

The word laser was originally an acronym for *Light Amplification by Stimulated Emission of Radiation*. The cavity of a standard laser—such as a HeNe laser—is composed of four main components: a highly reflective mirror, a partially reflecting mirror, something to supply the pumping energy, and the gain medium. The two mirrors are at opposite ends of the cavity and serve to contain most of the light to the cavity while allowing a small amount to exit—this is the output of the laser. The pump energy is generally supplied by either an external light source, or an electric field. The gain medium is what amplifies the laser light and consists of gases, such as Helium and Neon, or a crystal, such as Neodymium-doped Yttrium Aluminium garnet (Nd:YAG).

More precisely, the atoms in the gain medium are pumped with energy raising the electrons to an excited state. For stimulated emission an incident photon triggers the emission of a photon from this excited electron [1]. This new photon absorbs the electron’s energy as it returns to its ground state, amplifying the light. Furthermore, the photon is released with the same direction and phase as the incident one, maintaining coherence [1]. These processes are highlighted in Figure 1.1. The amplified light perpetually bounces between the two mirrors becoming more amplified with each interaction with an excited electron, a fraction of this light is able to pass

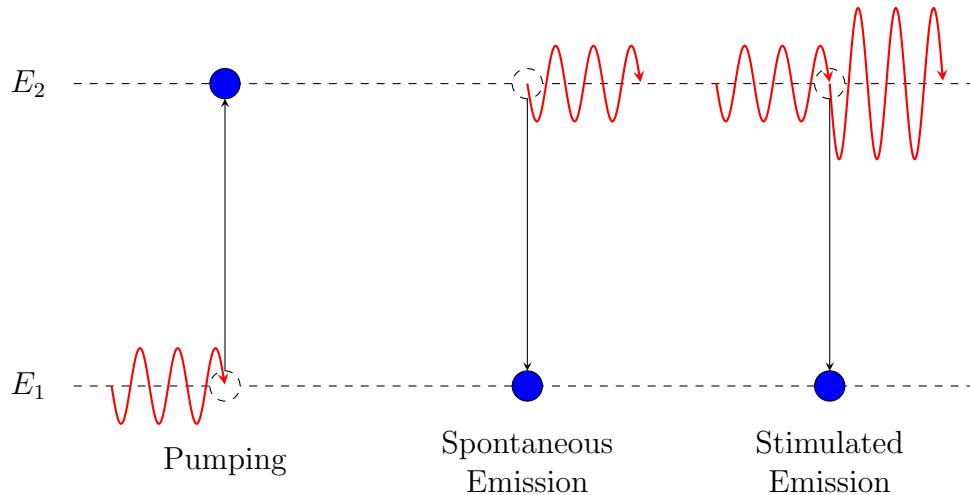


Figure 1.1: In pumping, incident photons are absorbed by electrons which are then excited into a higher energy state (the electrons may also be excited by an electric field). In spontaneous emission, an electron in an excited state emits a photon and returns to the lower energy state. Finally, in stimulated emission, an incident photon interacts with an electron in an excited state, the electron donates its energy to the photon and returns to the lower energy state.

through the partially reflecting mirror to become the output of the laser.

This is how laser light is generated, but, laser light has two fundamental characteristics that regular light does not. The first is that the light is highly monochromatic—the light contains (ideally) a single frequency. This is in contrast with an incandescent light bulb, for example, which emits light at all wavelengths with relative intensities given by Planck's law of black-body radiation. The other key feature is coherence, in which all the peaks and troughs of the light overlap giving very strong constructive interference—this is why laser light can be so intense. Something like a light emitting diode (LED) does not produce laser light because, while monochromatic, the light is not coherent. The monochromatic nature is a result of how the light is generated; the frequency of the light emitted directly corresponds to the energy difference between the excited and ground states of the gain medium (as with LEDs). Note that this correspondence means the frequency the laser operates at is fixed, and cannot be

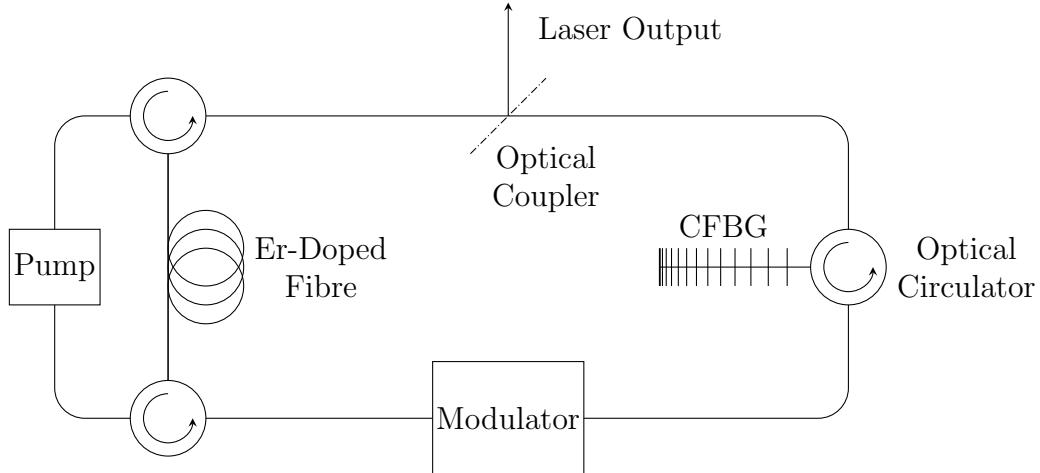


Figure 1.2: Typical cavity of a tuneable laser. The pulses travel clockwise around each loop.

changed. Coherence is achieved in the laser cavity—the cavity is carefully designed so the light resonates with the length of the cavity. This causes all the peaks and troughs to align, and interfere constructively.

1.1 Tuneable Lasers

A tuneable laser, on the other hand, has the ability to vary its wavelength by up to about 100 nanometres [2–4]. Tuneable lasers in fact lase at all of these different frequencies simultaneously. This tuneability is quite useful and has applications in spectroscopy and high resolution imaging such as coherent anti-Stokes Raman spectroscopy and optical coherence tomography [2, 4, 5], as well as communications and diagnostics of ultra fast processes [6]. However, having the capacity to tune the wavelength comes with a cost; tuneable lasers vary considerably from standard lasers in both construction and operation, but, the central ideas are the same. A typical tuneable laser cavity is shown in Figure 1.2. Instead of being perpetually reflected back and forth, the light travels in a circular fashion around the cavity—this is a so called

ring laser. The gain medium is still excited by an external power source, and the output passes through a partially reflecting mirror. However, a tuneable laser also contains a few additional components, each of which will be described in the following subsections.

1.1.1 Optical Coupler and Laser Output

The optical coupler is a device that splits its input into two outputs—one continues through the laser cavity, while the other exits the cavity to become the output of the laser. There are multiple devices that can accomplish this, however, the simplest is a partially reflecting mirror [1]. These mirrors are characterized by their reflection coefficient, R . In the schematic shown in Figure 1.2, the part of the signal that is reflected exits the cavity, whereas the part of the signal transmitted remains within the cavity.

1.1.2 Modulator

The optical pulse in the cavity can be modulated with one of two methods—amplitude modulation, or phase modulation. Mathematically these can be considered equivalent since they are Fourier transforms of each other, but in practice these are accomplished by different apparatuses. For us, we shall assume the pulses are amplitude modulated. Generally, the pulse can be amplitude modulated using one of two techniques. The first is with an acousto-optic modulator which utilizes the acousto-optic effect where the index of refraction of the fibre is varied by a sound wave [7, 8]. The second—and more common—method uses the electro-optic effect and is called an electro-optic modulator. These work the same way, the fibre inside the modulator responds to an applied electric field, modifying the index of refraction [7–10]. The shape of the electrical (sound) pulse can be controlled in such a way to provide the desired effect

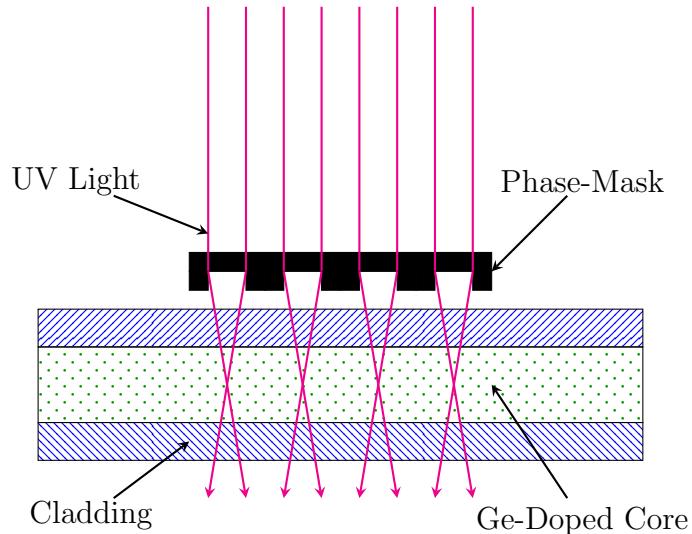
of modulation.

Moreover, a Mach–Zehnder interferometer can be used in tandem with an acousto- or electro-optic modulator to modulate an optical pulse. The incident pulse is separated with a Y-waveguide, one of the two branches is then modulated, and finally the two branches are recombined with another Y-waveguide [1, 7, 8]. In the absence of a modulator on one of the branches the two pulses recombine into the original pulse. However, with modulation the two pulses are no longer identical, and so when the recombine, they interfere. This interference can be used to further modulate the pulse [7–9, 11].

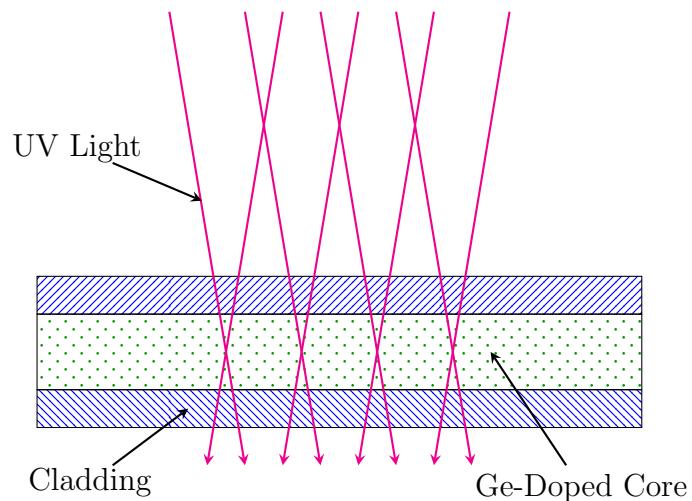
1.1.3 Fibre Bragg Grating

A fibre Bragg grating (FBG) is an optical fibre where the refractive index varies periodically along its length [12]. To achieve this, typically silica fibres are doped with Germanium which when exposed to intense ultraviolet (UV) light alters the refractive index of the core [13, 14]. The photosensitivity of the optical fibres can be increased by more than an order of magnitude by the Germanium doping [12, 13].

FBGs can be manufactured using one of two methods—the phase-mask method [1, 9, 13, 14], or the holographic side exposure method [1, 9, 12–14]—these are shown in Figure 1.3. Both methods cause the periodic nature of the refractive index through interference. In the phase-mask method (Figure 1.3a), a single beam of UV light passes through the phase-mask which acts as a series of lenses, focusing the light onto the core—this causes a sinusoidal interference pattern. Similarly, in the holographic side exposure method (Figure 1.3b), two beams of UV light are instead used to create the interference pattern.



(a) **Phase-Mask:** The phase-mask refracts the UV light onto the core.



(b) **Holographic Side Exposure:** Two incident beams intersect at the core.

Figure 1.3: Depictions of the two most common methods for manufacturing FBGs. The UV light is focused on the core causing periodic constructive and destructive interference. The UV light alters the refractive index of the core, causing a sinusoidal variation along the length.

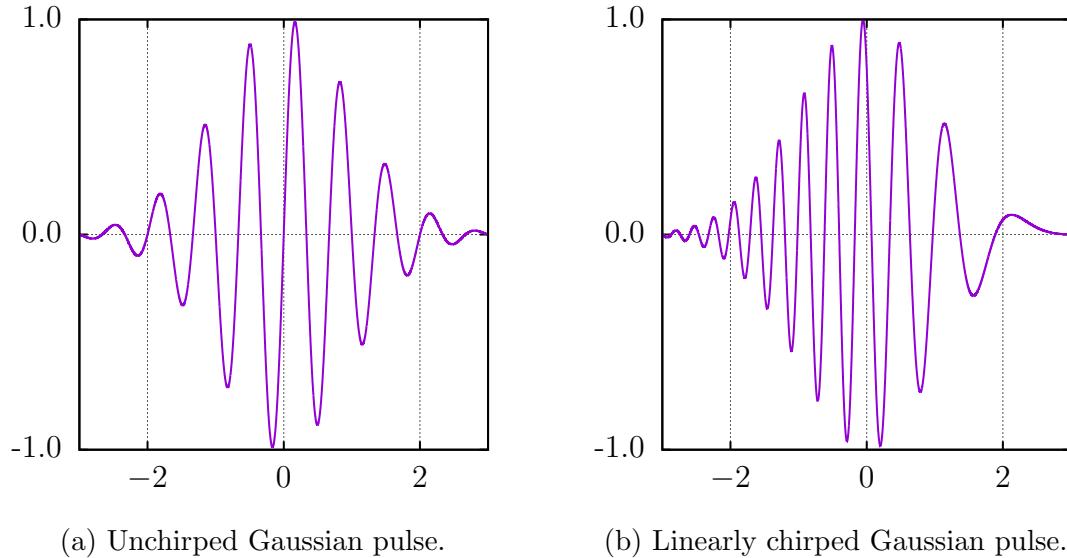


Figure 1.4: In an unchirped pulse the frequency is constant, however, in a chirped pulse the frequency varies along the envelope.

The purpose of FBGs is that they act as reflective filters [1, 9, 12, 14]. Due to the periodicity of the refractive index, light with the corresponding wavelength will be reflected with all others passing through. This wavelength is defined by the Bragg condition [1, 6, 9, 12–14]:

$$\lambda_B = 2\Lambda\bar{n}, \quad (1.1)$$

where λ_B is the Bragg wavelength, Λ is the period of the grating, and \bar{n} is the average index of refraction. Ideally, only light with a wavelength of λ_B would be reflected, however, there is a very narrow neighbourhood around λ_B in which wavelengths are partially reflected too, the characteristic width of this region is known as the stop-band.

Chirped Fibre Bragg Grating

Chirp is simply the term for a signal that has a non-constant frequency across it. Figure 1.4 shows examples of chirped and unchirped Gaussian pulses—the most common type of chirp is linear chirp, where the frequency varies linearly across the pulse. Because of this, for a spatially varying frequency, the oscillations are characterized by $\exp(ik(x)x) = \exp(iCx^2)$, where $k(x) = Cx$ is the linear variation of the wave number, and C is the chirp parameter. By using a chirped phase-mask a chirped fibre Bragg grating (CFBG) can be created. Since the period of the refractive index varies across the CFBG, so does when the Bragg condition, (1.1), is satisfied. This causes most wavelengths to be reflected by a CFBG, but with each wavelength penetrating to a different depth. Note that a monotonic chirping ensures that all the wavelengths are simultaneously supported. A consequence of this is that a time delay is created between wavelengths—this is depicted in Figure 1.5 with the upper portion showing the refractive index as a function of the depth. In this orientation, the red (dashed) wave is unable to penetrate as far as the blue (solid) wave since each wave is reflected where it matches the frequency of the refractive index.

Furthermore, the speed of light in an optical fibre is slightly dependent on the wavelength—this causes light with a longer wavelength to travel faster, and is known as chromatic dispersion. This is a large problem in fibre optic communications, the signal can spread and potentially becomes uninterpretable after vast distances. However, chromatic dispersion can be counteracted using a CFBG [1, 9, 13, 14] (in the opposite orientation of Figure 1.5). By forcing the longer wavelengths to travel farther the dispersion can be reversed, restoring the original signal. In one experiment [15], a signal was successfully transmitted over 109 km at 40 Gb/s by compensating the dispersion with two 40 cm CFBGs. Over this distance, the pulse would have spread to about 55 times its original width, and could only have been transmitted 4 km at that

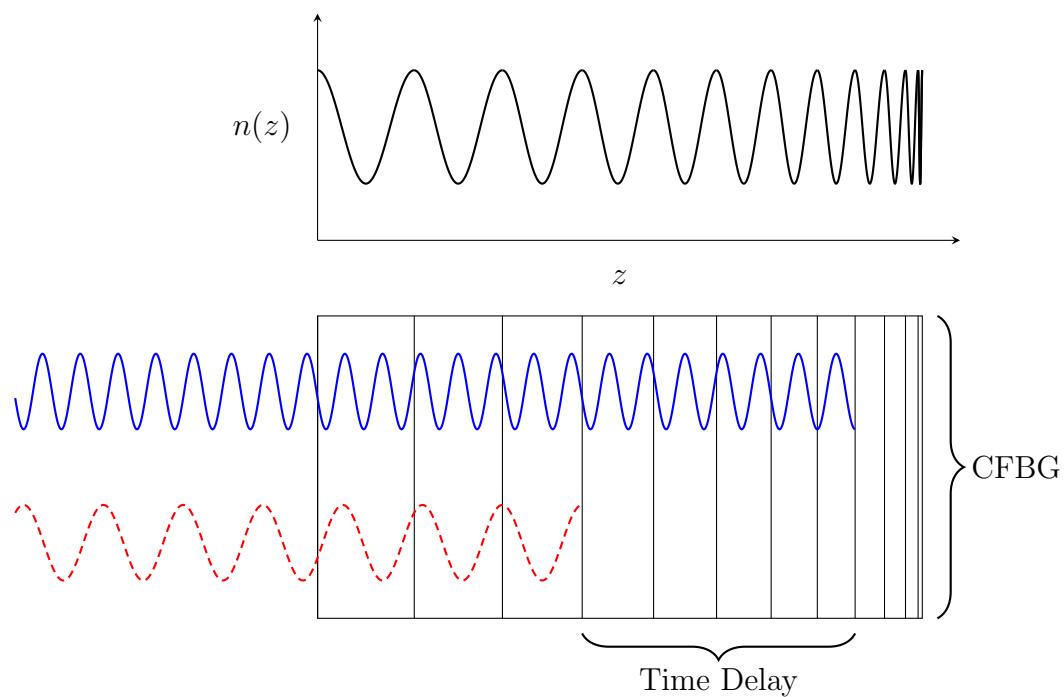


Figure 1.5: Chirped fibre Bragg grating. Light is reflected when the wavelength coincides with the Bragg wavelength. This causes light with different wavelengths to penetrate to different depths. In turn, dispersion is either heightened or compensated—depending on the orientation.

bit rate. At reduced bit rates however, a 10 cm CFBG can compensate the dispersion of 300 km of fibre [9]. However, in our case, we wish to accelerate the dispersion, simulating hundreds of metres of fibre, so the CFBG is used in the orientation shown in Figure 1.5.

1.1.4 Optical Circulator

An optical circulator is a device that routes signals from port to port in a circular fashion [1, 9, 13], the symbol for a four port optical circulator is shown in Figure 1.6. A signal entering from port 1 will be outputted from port 2; a signal entering from port 2 will exit from port 3; and so forth. Typically, optical circulators have three or four ports, with the first port being input only, and the final port being output only [1]. Optical circulators are most commonly used with devices that reflect signals instead of transmit them. For example, a signal may enter through port 1, exit through port 2, be reflected by an FBG, re-enter port 2, and finally exit through port 3.

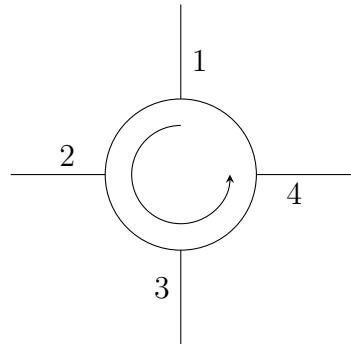


Figure 1.6: Symbol for a four port optical circulator.

1.1.5 Optical Amplifier and Pump Laser

Optical amplifiers are of particular importance in fibre optic communications, they are used to restore the strength of a signal after it has been attenuated over large distances, or when a signal is divided into multiple paths [1, 14]. They are also more efficient, and introduce less noise than an electrical repeater. In this context, the

optical amplifier provides the energy of the laser [1]. Most commonly, optical amplifiers are created by doping a length of fibre (called the gain fibre) with a rare-earth element which receives power from a pump laser [1, 9, 14]. The most common dopant is Erbium, however, Ytterbium and Neodymium are also used—Holmium, Samarium, Thulium, and Tellurium are infrequently used as well [9].

Erbium-doped fibre amplifiers (EDFAs) are used most widely since Erbium has a band gap that corresponds to 1.54–1.57 μm , which is the preferred band for fibre optics since this has the least power loss [1, 9, 14]. The pump laser typically operates at 980 nm or 1.48 μm because these wavelengths are able to transfer the most power (up to 100 mW) into the fibre while introducing minimal noise [1, 9, 13, 14]. The pump power can be applied either forwards (with the laser), backwards (against the laser), or both [1], with each configuration having similar performance [9]. However, backwards pumping has slightly better performance at high powers when the gain begins to saturate¹ [9]. Additionally, with the configuration shown in Figure 1.2, the optical circulators can be used to isolate the pump circuit from the rest of the laser cavity.

1.2 Mode-Locking

One of the fundamental features of a laser is coherence. However, in the context of tuneable lasers it is perhaps less clear what is meant by coherence since the light is no longer monochromatic. Instead, we have a particular peak of all the frequencies aligned, that is, the phase shift between modes is zero. This process is called mode-locking—all of the different modes are locked together at a peak [6, 7, 11]. This of course creates a very intense and short pulse of light—as short as 5 fs [6]. At this particular peak all of the frequencies constructively interfere as with a standard laser,

¹This concept will be discussed in Section 3.1.1.

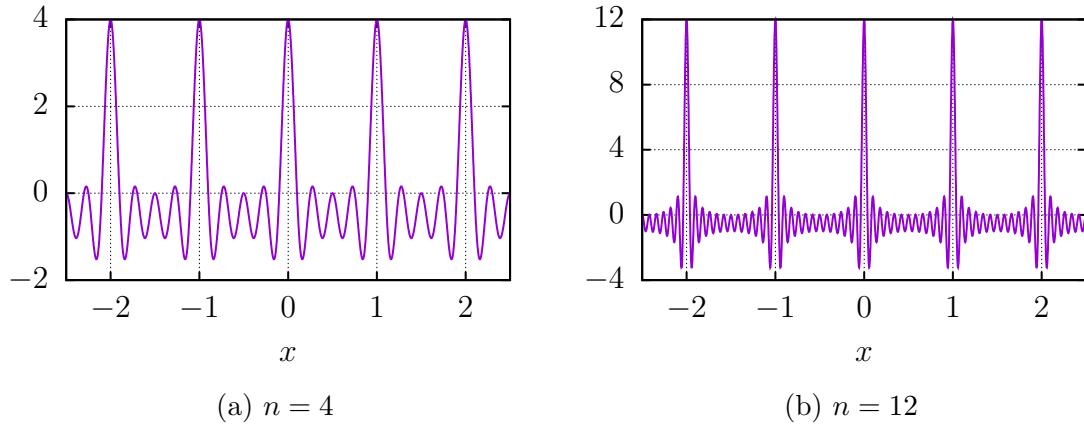


Figure 1.7: Two examples of mode-locking given by $\sum_{i=1}^n \cos(2\pi i x)$.

however, deviations from the peak lead to destructive interference. Two examples of this are shown in Figure 1.7, and as expected, including more modes simultaneously narrows and intensifies the pulse.

2

Chapter

Previous Modelling Efforts

In this chapter we shall explore the previous modelling efforts by first reviewing the classic equations used in nonlinear optics. We will then build upon this to obtain the master equation of mode-locking, where the solutions to this will briefly be reviewed before considering ‘discretized’ functional models.

2.1 Generalized Nonlinear Schrödinger Equation

The standard equation for studying nonlinear optics is the nonlinear Schrödinger equation (NLSE) [11, 12, 16–20]

$$\frac{\partial A}{\partial z} = -i \frac{\beta_2}{2} \frac{\partial^2 A}{\partial T^2} + i\gamma |A|^2 A.$$

Here $A = A(z, T)$ is the complex pulse amplitude, $\beta_2 \in \mathbb{R}$ is the second order—or group delay—dispersion, and $\gamma \in \mathbb{R}$ is the coefficient of nonlinearity or Kerr coefficient. This equation can be derived from the nonlinear wave equation for electric fields, the derivation is presented in detail in [11, 12]. Within the derivation comoving coordinates are used so that the reference frame propagates with the pulse at the

group velocity. This is achieved with the substitution

$$T = t - \frac{z}{v_g}, \quad (2.1)$$

reminiscent of moving to a fixed frame in a free boundary problem [21]. In this way, z defines the distance travelled by the pulse, and T the time difference from the peak of the pulse. Moreover, $T = 0$ is the peak of the pulse regardless of the actual time elapsed, because of this, generally T is not very large. In standard coordinates the nonlinear Schrödinger equation has an additional term:

$$\frac{\partial A}{\partial z} = -i\frac{\beta_2}{2}\frac{\partial^2 A}{\partial t^2} + i\gamma|A|^2A - \frac{1}{v_g}\frac{\partial A}{\partial t}.$$

However, under the transformation (2.1), we have that $\frac{\partial}{\partial t} \mapsto \frac{\partial}{\partial T}$, and $\frac{\partial}{\partial z} \mapsto \frac{\partial}{\partial z} - \frac{1}{v_g}\frac{\partial}{\partial T}$ by chain rule, thus, cancelling out this final term.

In practice, this equation lacks a few key terms. Thus, it is often generalized by adding amplification, loss, and higher order terms. This gives the generalized nonlinear Schrödinger equation (GNLSE) [2, 11, 19, 22–24],

$$\frac{\partial A}{\partial z} = -i\frac{\beta_2}{2}\frac{\partial^2 A}{\partial T^2} + \frac{\beta_3}{6}\frac{\partial^3 A}{\partial T^3} + i\gamma|A|^2A + \frac{1}{2}g(A)A - \alpha A, \quad (2.2)$$

where $\beta_3 \in \mathbb{R}$ is the third order coefficient of dispersion, $g(A)$ is an amplifying term due to the gain, and $\alpha \in \mathbb{R}$ is the loss due to scattering and absorption.

The GNLSE takes the same form as the Schrödinger equation with the inclusion of the cubic nonlinear term, hence its name. For this reason, it is sometimes referred to as the cubic nonlinear Schrödinger equation. For intensities approaching 1 GW/cm^2 , the γ parameter must be replaced by $\gamma_0(1 - b_s|A|^2)$, where b_s is a saturation parameter [11],

this has the addition of a quintic term to incorporate nonlinearities associated with such large powers. Furthermore, the β terms come from a Taylor expansion of the wavenumber [25], that is,

$$\begin{aligned} k(\omega) &= k_0 + \frac{\partial k}{\partial \omega} \Big|_{\omega_0} (\omega - \omega_0) + \frac{1}{2} \frac{\partial^2 k}{\partial \omega^2} \Big|_{\omega_0} (\omega - \omega_0)^2 + \frac{1}{6} \frac{\partial^3 k}{\partial \omega^3} \Big|_{\omega_0} (\omega - \omega_0)^3 + \dots \\ &= k_0 + \frac{1}{v_g} (\omega - \omega_0) + \frac{1}{2} \beta_2 (\omega - \omega_0)^2 + \frac{1}{6} \beta_3 (\omega - \omega_0)^3 + \dots, \end{aligned}$$

where k_0 is the phase shift. Typically, the third order effects must only be considered for ultrashort pulses—pulse widths less than ~ 5 ps—because of their large bandwidth [11].

2.2 The Master Equation of Mode-Locking

The GNLSE has many applications in nonlinear optics and fibre optic communications, however, in the context of lasers we wish to add a modulation term to ensure mode-locking, this yields the master equation of mode-locking, [7, 25–31]

$$\frac{\partial A}{\partial z} = -i \frac{\beta_2}{2} \frac{\partial^2 A}{\partial T^2} + \frac{\beta_3}{6} \frac{\partial^3 A}{\partial T^3} + i\gamma |A|^2 A + \frac{1}{2} g(A) A - \alpha A - M(T). \quad (2.3)$$

The most common form of modulation is the sinusoid $M(T) = \frac{1}{2} \frac{M_s}{\omega_M^2} (1 - \cos(\omega_M T))$ [7, 25, 26, 32], and since T is generally small, by (2.1), we expand via it's Taylor series so that $M(T) = \frac{1}{2} M_s T^2 + \mathcal{O}(T^4)$. This brings us to the most common form of the master equation of mode-locking (neglecting the third order dispersion),

$$\frac{\partial A}{\partial z} = -i \frac{\beta_2}{2} \frac{\partial^2 A}{\partial T^2} + i\gamma |A|^2 A + \frac{1}{2} g(A) A - \alpha A - \frac{1}{2} M_s T^2. \quad (2.4)$$

Commonly, in an attempt to simplify the equation the gain is assumed to be constant with $g(A) = 2g_0 \in \mathbb{R}$. As a whole no analytic solution is known for (2.4), however,

after additional simplifications there are three flavours of solutions.

In the least complicated case, the modulation and nonlinearity are both omitted giving

$$\frac{\partial A}{\partial z} = -i \frac{\beta_2}{2} \frac{\partial^2 A}{\partial T^2} + (g_0 - \alpha)A.$$

This results in a solution in the form of a hyperbolic secant [26–28]. On the other hand, including the nonlinearity yields

$$\frac{\partial A}{\partial z} = -i \frac{\beta_2}{2} \frac{\partial^2 A}{\partial T^2} + i\gamma|A|^2A + (g_0 - \alpha)A,$$

where a similar solution is found, however, it is instead of the form of a chirped hyperbolic secant [31, 33]. Both of these are unsurprising since this reduces to the soliton solution of the NLSE [12]. Finally, by including the modulation term, and excluding the nonlinearity we have

$$\frac{\partial A}{\partial z} = -i \frac{\beta_2}{2} \frac{\partial^2 A}{\partial T^2} + (g_0 - \alpha)A - \frac{1}{2}M_sT^2.$$

This can be solved using separation of variables and one finds the solutions are the Gaussian–Hermite polynomials [5, 7, 25, 26, 29–32]. However, in practice only the Gaussian is stable—the higher modes quickly decay within the laser [7, 26, 29, 32]. For a more comprehensive and exhaustive history see [29].

2.3 Discrete Functional Models

While solutions to the master equation yield reasonable results, it is not necessarily representative of what happens within the laser cavity. The issue with (2.4) is that it assumes each process effects the pulse continuously within the cavity. As highlighted by Figure 1.2, this is a rather poor assumption. Within the cavity each effect is localized to its corresponding component: almost all of the dispersion happens within the CFBG, the pulse is only amplified within the Erbium-doped fibre, etc. Thus, perhaps a better model is one where (2.4) is broken down into the individual components giving the effect of each ‘block’ of the cavity. Each of the blocks can then be composed together functionally to give the effect of one circuit around the cavity. This yields an algebraic equation instead of a differential one.

Such a method was first proposed in 1955 by Cutler [34] while analyzing a microwave regenerative pulse generator. This method was adapted for mode-locked lasers in 1969 by Siegman and Kuizenga [35], which they then greatly added to the following year [36]. Kuizenga and Siegman also had success experimentally validating their model [37,38]. The effects of the nonlinearity would not be considered until Martinez, Fork, and Gordon [39,40] tried to model passive mode-locking—mode-locking without the use of a modulator. In the absence of a modulator the nonlinearity becomes crucial to shaping the pulse. This issue has recently been readdressed by Burgoyne [5] in the literature for tuneable lasers. In each of these models the effect of each block is described by a Gaussian transfer function. These transfer functions are then multiplied together to give the overall effect.

Despite these attempts, several short-comings exist. The clearest is that none of these models have contained every block—either the nonlinearity or the modulation

have been omitted. In the framework of tuneable lasers, each component plays a crucial role and the tuneable laser will not function without the inclusion of all of the components. Another key drawback is that the functional operations of some of the components are somewhat phenomenological. While these functions are chosen based on the observed output, they are not necessarily consistent with their underlying physics. Finally, none of these previous models have been able to exhibit a phenomena called *wave-breaking* in which the self-phase modulation (SPM) of the pulse becomes too strong, distorting and damaging the wave until it ultimately becomes unstable and unsustainable. This notion will be explained in greater detail in Chapter 4.

Chapter 3

A New Model

In this chapter we shall derive our new model and solve it analytically in the linear case. To accomplish this we shall use the ideas of the previous functional models [5, 34–40]. To alleviate some of the pitfalls mentioned in the previous chapter, we shall include all five processes involved in the modification of the pulse within the laser cavity (gain, nonlinearity, loss, dispersion, and modulation). In addition to this, the functional operations associated with each component will be derived from (2.2),

$$\frac{\partial A}{\partial z} = -i\frac{\beta_2}{2}\frac{\partial^2 A}{\partial T^2} + \frac{\beta_3}{6}\frac{\partial^3 A}{\partial T^3} + i\gamma|A|^2A + \frac{1}{2}g(A)A - \alpha A,$$

with the exception of the modulation in which we consider the exact functional form to be determined by the laser operator.

3.1 Components

We shall begin our analysis with the derivation of the functional operators for the five components.

3.1.1 Gain

Within the Er-doped gain fibre, the gain term is dominant, and equation (2.2) reduces to

$$\frac{\partial A}{\partial z} = \frac{1}{2}g(A), \quad (3.1)$$

where $g(A)$ takes the form [2, 5–7, 22–26, 28, 29, 31, 33]

$$g(A) = \frac{g_0}{1 + E/E_{\text{sat}}}A, \quad E = \int_{-\infty}^{\infty} |A|^2 dT, \quad (3.2)$$

where g_0 is a small signal gain, E is the energy of the pulse, and E_{sat} is the energy at which the gain begins to saturate. Multiplying (3.1) by \bar{A} , the complex conjugate of A , yields

$$\bar{A} \frac{\partial A}{\partial z} = \frac{1}{2} \frac{g_0 |A|^2}{1 + E/E_{\text{sat}}},$$

adding this to its complex conjugate and integrating over T gives

$$\frac{dE}{dz} = \frac{g_0 E}{1 + E/E_{\text{sat}}}. \quad (3.3)$$

For $E \ll E_{\text{sat}}$ the energy will grow exponentially, whereas for $E \gg E_{\text{sat}}$ the gain has saturated and so the growth is linear. To obtain a closed form solution, (3.3) is integrated over a gain fibre of length z and assuming the energy increases from E to E_{out} , then

$$g_0 z = \log \frac{E_{\text{out}}}{E} + \frac{E_{\text{out}} - E}{E_{\text{sat}}},$$

and by exponentiating, rearranging, and applying W , the Lambert W function¹,

$$W\left(\frac{E}{E_{\text{sat}}}\text{e}^{E/E_{\text{sat}}}\text{e}^{g_0 z}\right) = W\left(\frac{E_{\text{out}}}{E_{\text{sat}}}\text{e}^{E_{\text{out}}/E_{\text{sat}}}\right) = \frac{E_{\text{out}}}{E_{\text{sat}}},$$

by (A.1). This results in the closed form expression

$$E_{\text{out}}(z) = E_{\text{sat}}W\left(\frac{E}{E_{\text{sat}}}\text{e}^{E/E_{\text{sat}}}\text{e}^{g_0 z}\right)$$

with the desired property that $E_{\text{out}}(0) = E$. Furthermore, since $E \sim |A|^2$, the gain in terms of the amplitude is given by

$$G(A; E) = \left(\frac{E_{\text{out}}(L_g)}{E}\right)^{1/2} A = \left(\frac{E_{\text{sat}}}{E}W\left(\frac{E}{E_{\text{sat}}}\text{e}^{E/E_{\text{sat}}}\text{e}^{g_0 L_g}\right)\right)^{1/2} A,$$

where L_g is the length of the gain fibre.

3.1.2 Fibre Nonlinearity

The nonlinearity of the fibre arises from the parameter γ ; in regions where this effect is dominant expression (2.2) becomes

$$\frac{\partial A}{\partial z} - i\gamma|A|^2 A = 0. \quad (3.4)$$

This expression can be manipulated in a similar manner to the gain to show $\frac{\partial}{\partial z}|A|^2 = 0$, suggesting that $A(z, T) = A_0(T)e^{i\varphi(T, z)}$. Substituting this representation into (3.4) and setting $\varphi(T, 0) = 0$ gives $\varphi(T, z) = \gamma|A|^2 z$. For a fibre of length L_f the effect of the nonlinearity is thus

$$F(A) = A\text{e}^{i\gamma|A|^2 L_f},$$

¹See Appendix A.

this is also frequently called the Kerr nonlinearity or Kerr effect after John Kerr who discovered the effect in 1875. As we shall see in Section 4.2.1, this is what is responsible for self-phase modulation and ultimately the degradation of the pulse [18, 30, 39, 41].

3.1.3 Loss

Two sources of loss exist within the laser circuit: the loss due to the output coupler and the optical loss due to absorption and scattering. The first case is simply a scalar multiplication depending on the reflectivity of the output coupler. The loss due to absorption and scattering can be derived from (2.2),

$$\frac{\partial A}{\partial z} = -\alpha A.$$

Combining these two effects give a loss that takes the form

$$L(A) = (1 - R)e^{-\alpha L} A,$$

where R is the reflectivity of the output coupler, and L is the total length of the laser circuit.

3.1.4 Dispersion

Within the laser cavity, the dispersion is dominated by the CFBG. In comparison, the dispersion due to the fibre is negligible. The dispersive terms of (2.2) give

$$\frac{\partial A}{\partial z} = -i \frac{\beta_2}{2} \frac{\partial^2 A}{\partial T^2} + \frac{\beta_3}{6} \frac{\partial^3 A}{\partial T^3}, \quad (3.5)$$

and since dispersion acts in the frequency domain it is convenient to take the Fourier transform of (3.5) [42, 43], giving

$$\frac{\partial}{\partial z} \mathcal{F}\{A\} = i \frac{\omega^2}{2} \left(\beta_2 - \frac{\beta_3}{3} \omega \right) \mathcal{F}\{A\}.$$

The effect of dispersion is then

$$D(A) = \mathcal{F}^{-1} \left\{ e^{i\omega^2 L_D (\beta_2 - \beta_3 \omega / 3) / 2} \mathcal{F}\{A\} \right\}.$$

For a highly dispersive media the third order effects may need to be considered [11, 44]. However, for simplicity and because of the nature of the grating, the third order effects will be neglected [11, 12]—we set $\beta_3 = 0$ for the subsequent analysis.

3.1.5 Modulation

In the master equation of mode-locking, the amount of modulation is characterized by the parameter M_s through the term $\frac{1}{2} M_s T^2 A$. In this new model, the modulation is considered to be applied externally through its action on the spectrum and for simplicity the representation is taken as the Gaussian

$$M(A) = e^{-T^2 / 2T_M^2} A,$$

where T_M is a characteristic width of the modulation.

Despite assuming a Gaussian modulation the following solution can be generalized to any modulation function in $L^2(\mathbb{R})$ since Gaussians in fact span $L^2(\mathbb{R})$ as will be shown shortly.

3.2 Span of Gaussians in $L^2(\mathbb{R})$

In order to show that Gaussians span $L^2(\mathbb{R})$, we shall start our analysis with the span of the Hermite polynomials in $L^2(\mathbb{R})$. Typically the Hermite polynomials are recursively defined as [45–47]

$$H_n(x) := (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2},$$

with the inner product²

$$\langle f, g \rangle = \int_{\mathbb{R}} f(x)g(x)e^{-x^2} dx,$$

where e^{-x^2} is the weighting function. This is so that

$$\langle H_m, H_n \rangle = \sqrt{\pi} 2^n n! \delta_{mn},$$

and therefore, the Hermite polynomials form an orthogonal set [46–50]. Consider instead, the Gaussian–Hermite polynomials

$$\tilde{H}_n(x) := e^{-x^2/2} H_n(x),$$

with the inner product

$$\langle f, g \rangle = \int_{\mathbb{R}} f(x)g(x) dx,$$

notice that the weighting function has been absorbed into the Hermite polynomials. We shall now show that the set of Gaussian–Hermite polynomials spans $L^2(\mathbb{R})$.

²The complex conjugate of g is omitted since the functions dealt with are real.

Theorem 3.1. *The set of Gaussian–Hermite polynomials, $\mathcal{G}(\mathbb{R}) = \{\tilde{H}_n(x) : x \in \mathbb{R}, n \in \mathbb{N}\}$, spans $L^2(\mathbb{R})$.*

Proof. The Gaussian–Hermite polynomials are an orthogonalization of the set $\Xi = \{x^n e^{-x^2/2} : x \in \mathbb{R}, n \in \mathbb{N}\}$, and therefore, have the same span [49, 50]. Thus, it is sufficient to show that Ξ spans $L^2(\mathbb{R})$. Suppose there exists an $f \in L^2(\mathbb{R})$ such that $\langle f, \tilde{H}_n \rangle = 0$ for all n , or equivalently, that $\langle f, g_n \rangle = 0$ for $g_n(x) = x^n e^{-x^2/2}$ for all n so that it is not in the span of $\mathcal{G}(\mathbb{R})$.

Let us now consider the function [46, 47, 50]

$$F(z) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(x) e^{zx} e^{-x^2/2} dx,$$

noting that $f(x) e^{zx} e^{-x^2/2} \in L^2(\mathbb{R})$ by the Cauchy–Schwarz inequality [51, 52], so F converges, and that F is holomorphic. The first exponential can be expanded into its Maclaurin series, then with Fubini’s theorem we obtain

$$F(z) = \frac{1}{\sqrt{2\pi}} \sum_{n=0}^{\infty} \frac{z^n}{n!} \int_{\mathbb{R}} f(x) x^n e^{-x^2/2} dx.$$

Recall, that by assumption

$$\int_{\mathbb{R}} f(x) x^n e^{-x^2/2} dx = \langle f, g_n \rangle = 0,$$

for all n and so $F(z) \equiv 0$. We now notice that

$$0 = F(z) = F(i\omega) = \mathcal{F}\left\{f e^{-x^2/2}\right\},$$

and now it is clear that $f(x) e^{-x^2/2} = 0$ almost everywhere, therefore, $f(x) = 0$ almost everywhere.

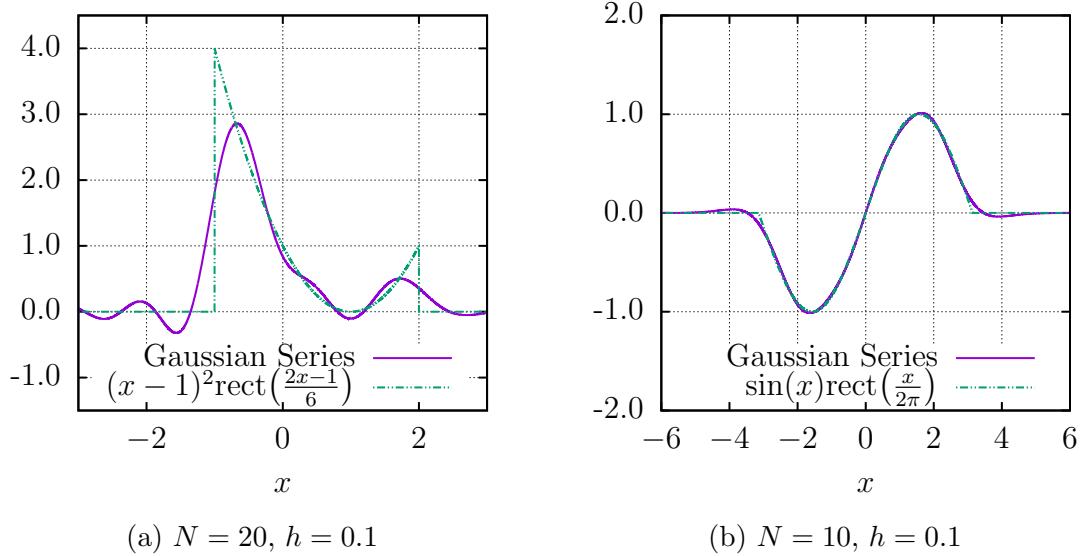


Figure 3.1: Two examples of Gaussian series.

□

Alternatively, without much difficulty it can be shown that the Gaussian–Hermite polynomials satisfy

$$\frac{d^2 \tilde{H}_n(x)}{dx^2} - (x^2 - 2n - 1) \tilde{H}_n(x) = 0. \quad (3.6)$$

This has the form of the parabolic cylinder functions [53], and indeed the Gaussian–Hermite polynomials can be expressed in terms of parabolic cylinder functions [48,53]. Moreover, (3.6) can also be written in Sturm–Liouville form as

$$\frac{d}{dx} \left(1 \cdot \frac{d \tilde{H}_n(x)}{dx} \right) + (1 - x^2) \tilde{H}_n(x) = -2n \tilde{H}_n(x).$$

By the spectral theorem [49,51,54], this suggests that the Gaussian–Hermite polynomials form an orthogonal set with the weighting function $w(x) = 1$, and are complete in $L^2(\mathbb{R})$. Using this result, we can now show that Gaussian functions also span $L^2(\mathbb{R})$ with the following theorem.

Theorem 3.2. *Gaussians of a single variance span $L^2(\mathbb{R})$.*

Proof. Theorem 3.1 showed that any square integrable function can be expressed as

$$f(x) = \sum_{n=0}^{\infty} a_n \tilde{H}_n(x).$$

Using a similar idea to [55, 56], we shall now expand the Gaussian–Hermite polynomials using their definition

$$f(x) = \sum_{n=0}^{\infty} a_n (-1)^n e^{x^2/2} \frac{d^n}{dx^n} e^{-x^2}.$$

Finally, we can rewrite the derivatives using central differences [57] so that

$$f(x) = \sum_{n=0}^{\infty} a_n (-1)^n e^{x^2/2} \left[\lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{i=0}^n (-1)^i \binom{n}{i} \exp\left(-\left(x + h\left(\frac{n}{2} - i\right)\right)^2\right) \right]. \quad (3.7)$$

□

Two examples of Gaussian series from (3.7) are shown in Figure 3.1. These examples are taken from [55, 56], however, we achieve similar or better approximations with either fewer terms, or with an h value an order of magnitude larger. There are two reasons for this, first, we use central difference as opposed to backwards difference yielding a convergence of $\mathcal{O}(h^2)$ instead of $\mathcal{O}(h)$. Furthermore, because [55, 56] used backwards differences, the means of the Gaussians are all non-negative, whereas with central differences, our Gaussians’ means are both positive and negative—this leads to smaller coefficients, and better numerical stability. Moreover, Figure 3.2 shows the Gaussian series for the modulation function in [2, 58].

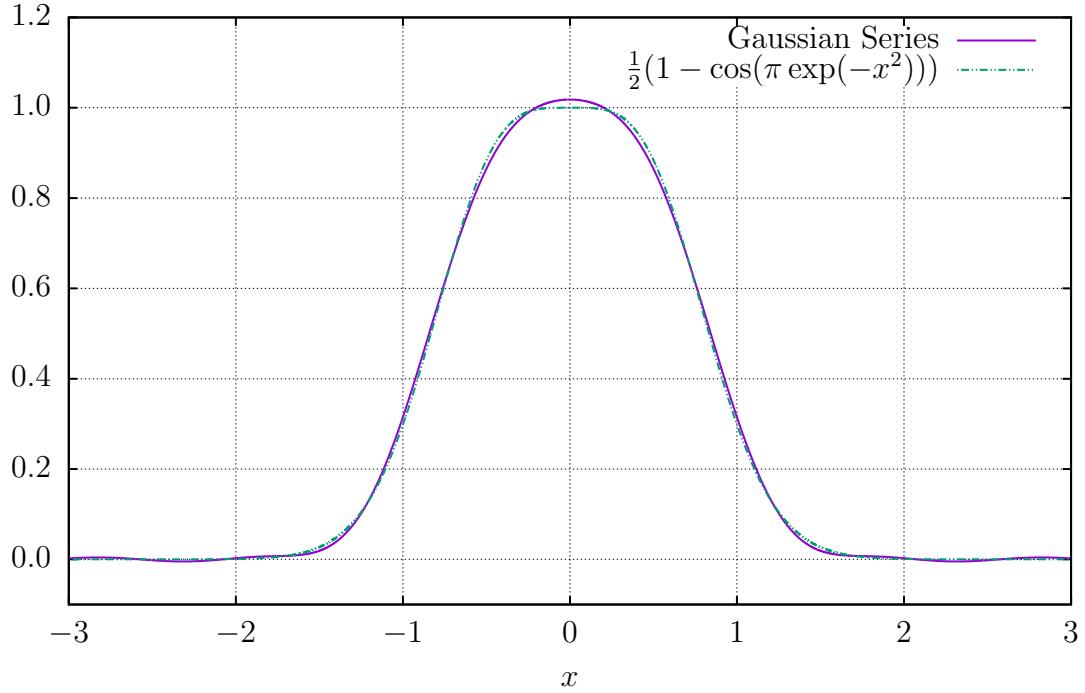


Figure 3.2: Gaussian series with $N = 20$, $h = 0.1$.

3.3 Non-Dimensionalization

The structure of each process of the laser can be better understood by re-scaling the time, energy, and amplitude. Specifically, the time shall be scaled by the characteristic modulation time which is related to the pulse duration, the energy by the saturation energy, and the amplitude will be scaled so that it is consistent with the energy:

$$T = T_M \tilde{T}, \quad E = E_{\text{sat}} \tilde{E}, \quad A = \left(\frac{E_{\text{sat}}}{T_M} \right)^{1/2} \tilde{A}.$$

Parameter	Symbol	Value	Sources
Absorption of Fibre	α	0.01–0.3 m ⁻¹	[23, 24, 31, 58]
Fibre Dispersion	β_2^f	-50–50 ps ² /km	[5, 9, 11, 22, 24, 44]
Fibre Nonlinearity	γ	0.001–0.01 W ⁻¹ m ⁻¹	[11, 24, 31]
Grating Dispersion	$\beta_2^g L_D$	10–2000 ps ²	[5, 9, 11, 59]
Length of Cavity	L	10–100 m	[22, 58]
Length of Fibre	L_f	0.15–1 m	[58]
Length of Gain Fibre	L_g	2–3 m	[5, 22–24]
Modulation Time	T_M	15–150 ps	[2, 5, 58]
Reflectivity of Optical Coupler	R	0.1–0.9	[22, 58]
Saturation Energy	E_{sat}	10 ³ –10 ⁴ pJ	[24, 31, 58]
Small Signal Gain	g_0	1–10 m ⁻¹	[24, 58]

Table 3.1: Orders of magnitude of various parameters.

The new process maps, after dropping the tildes, become

$$G(A) = (E^{-1}W(aEe^E))^{1/2}A, \quad (3.8a)$$

$$F(A) = Ae^{ib|A|^2}, \quad (3.8b)$$

$$L(A) = hA, \quad (3.8c)$$

$$D(A) = \mathcal{F}^{-1}\left\{e^{is^2\omega^2}\mathcal{F}\{A\}\right\}, \quad (3.8d)$$

$$M(A) = e^{-T^2/2}A, \quad (3.8e)$$

with the four dimensionless parameters (see Table 3.1)

$$a = e^{g_0 L_g} \sim 8 \times 10^3, \quad s = \sqrt{\frac{\beta_2 L_D}{2T_M^2}} \sim 0.2,$$

$$b = \gamma L_f \frac{E_{\text{sat}}}{T_M} \sim 1, \quad h = (1 - R)e^{-\alpha L} \sim 0.04,$$

which characterize the behaviour of the laser.

3.4 Combining the Effects

In this model the pulse is iteratively passed through each process, the order of which must now be considered. We are most interested in the output of the laser cavity, and so we shall start with the loss component. Next the pulse is passed through the CFBG, as well as the modulator. Finally, the pulse travels through the gain fibre to be amplified, and then we consider the effect of the nonlinearity since this is the region where the power is maximized. Note that in general the functional operators of the components do not commute, and therefore the order is indeed important—in contrast to the previous models. This is especially the case of dispersion due to the Fourier transform. The pulse after one complete circuit of the laser cavity is then passed back in to restart the process. Functionally this can be denoted as

$$\mathcal{L}(A) = F(G(M(D(L(A))))),$$

where \mathcal{L} is one loop of the laser. A solution to this model is one in which the envelope and chirp are unchanged after traversing every component in the cavity, that is, such that $\mathcal{L}(A) = Ae^{i\phi}$ —for some $\phi \in \mathbb{R}$.

3.5 Solution to the Linear Model

In the case of $b = 0$ —which shall be referred to as the linear model—a solution can be found analytically. It is expected the solution will take the form of a Gaussian. There are a few reasons for this; the solution to the previous models were Gaussian [34–36, 39, 40] (see Section 2.2), the equilibrium shape will be highly correlated to the shape of the modulation function, and since a Gaussian is a fixed point of the Fourier transform [43] dispersion will not alter the envelope.

Consider the initial pulse

$$A_0 = \sqrt{P} \exp\left(-(1 + iC) \frac{T^2}{2\sigma^2}\right) e^{i\phi_0},$$

where P is the peak power, C is the chirp, σ^2 is the variance, and ϕ_0 is the initial phase.

After passing through the optical coupler the pulse will simply decay to $A_1 = hA_0$.

The pulse then enters the CFBG, where it will maintain its Gaussian shape, however, it will spread [6, 11, 12]. This can be written as

$$A_2 = \sqrt{Ph\zeta} \exp\left(-(1 + i\tilde{C}) \frac{T^2}{2\tilde{\sigma}^2}\right) e^{i(\phi_0 + \phi)},$$

where $\tilde{\sigma}^2$ denotes the resulting variance, \tilde{C} denotes the resulting chirp, and ζ is the reduction of the amplitude caused by the spread. Next, the pulse is modulated:

$$A_3 = \sqrt{Ph\zeta} \exp\left(-(1 + i\tilde{C}) \frac{T^2}{2\tilde{\sigma}^2} - \frac{T^2}{2}\right) e^{i(\phi_0 + \phi)}.$$

Finally, the pulse travels through the gain fibre where it is amplified to

$$A_4 = \sqrt{P} \left(\frac{W(aEe^E)}{E} \right)^{1/2} h\zeta \exp\left(-(1 + i\tilde{C}) \frac{T^2}{2\tilde{\sigma}^2} - \frac{T^2}{2}\right) e^{i(\phi_0 + \phi)},$$

with E the energy of the pulse as it enters the gain fibre.

In equilibrium it must be that $A_0 = A_4 e^{-i\phi}$. More explicitly, this gives three conditions:

$$1 = \left(\frac{W(aEe^E)}{E} \right)^{1/2} h\zeta, \quad (3.9a)$$

$$\frac{1}{\sigma^2} = \frac{1}{\tilde{\sigma}^2} + 1, \quad (3.9b)$$

$$\frac{C}{\sigma^2} = \frac{\tilde{C}}{\tilde{\sigma}^2}. \quad (3.9c)$$

3.5.1 Spread Due to Dispersion

Each of these processes has a relatively straight forward effect, with the exception of dispersion. The effect of dispersion can be computed analytically for input pulses using (3.8d). In the case of a Gaussian pulse, we make use of the transforms [42, 43]

$$\mathcal{F}\left\{e^{-\eta T^2}\right\} = (2\eta)^{-1/2} e^{-\omega^2/4\eta}, \quad \mathcal{F}^{-1}\left\{e^{-\eta\omega^2}\right\} = (2\eta)^{-1/2} e^{-T^2/4\eta}.$$

From (3.8d) we have that

$$\begin{aligned} D\left(e^{-\eta T^2}\right) &= \mathcal{F}^{-1}\left\{e^{is^2\omega^2}\mathcal{F}\left\{e^{-\eta T^2}\right\}\right\}, \\ &= (2\eta)^{-1/2}\mathcal{F}^{-1}\left\{\exp\left(-\omega^2\left(\frac{1}{4\eta} - is^2\right)\right)\right\}, \\ &= (1 - 4i\eta s^2)^{-1/2}\exp\left(-T^2\frac{a}{1 - 4is^2\eta}\right). \end{aligned}$$

For us, $\eta = \frac{1}{2}\frac{1+iC}{\sigma^2}$; making this substitution yields

$$D(A_1) = \left(1 + \frac{2Cs^2}{\sigma^2} - \frac{2s^2}{\sigma^2}i\right)^{-1/2} \exp\left(-T^2\frac{1+iC}{2\sigma^2 - 4is^2(1+iC)}\right).$$

This can be greatly simplified by first rationalizing the denominators to give

$$D(A_1) = \left(\frac{1 + \frac{2Cs^2}{\sigma^2} + \frac{2s^2}{\sigma^2}i}{\left(1 + \frac{2Cs^2}{\sigma^2}\right)^2 + \left(\frac{2s^2}{\sigma^2}\right)^2}\right)^{1/2} \exp\left(-T^2\frac{(1+iC)(2\sigma^2 + 4Cs^2 + 4is^2)}{(2\sigma^2 + 4Cs^2)^2 + 16s^4}\right),$$

and then by writing in polar coordinates:

$$D(A_1) = \left(\left(1 + \frac{2Cs^2}{\sigma^2} \right)^2 + \left(\frac{2s^2}{\sigma^2} \right)^2 \right)^{-1/4} \exp \left(\frac{1}{2} i \arctan \left(\frac{\frac{2s^2}{\sigma^2}}{1 + \frac{2Cs^2}{\sigma^2}} \right) \right) \\ \times \exp \left(-T^2 \frac{\sigma^2 \left[1 + i \left(C + (1 + C^2) \frac{2s^2}{\sigma^2} \right) \right]}{2[(\sigma^2 + 2Cs^2)^2 + 4s^4]} \right).$$

Finally, this can be simplified further to

$$D(A_1) = \sigma \left((\sigma^2 + 2Cs^2)^2 + 4s^4 \right)^{-1/4} \exp \left(\frac{1}{2} i \arctan \left(\frac{2s^2}{\sigma^2 + 2Cs^2} \right) \right) \\ \times \exp \left(-T^2 \frac{\sigma^2 \left[1 + i \left(C + (1 + C^2) \frac{2s^2}{\sigma^2} \right) \right]}{2[(\sigma^2 + 2Cs^2)^2 + 4s^4]} \right).$$

From this expression it is clear that at equilibrium

$$\tilde{\sigma}^2 \sigma^2 = (\sigma^2 + 2Cs^2)^2 + 4s^4, \quad (3.10a)$$

$$\tilde{C} = C + (1 + C^2) \frac{2s^2}{\sigma^2}, \quad (3.10b)$$

$$\phi = \frac{1}{2} \arctan \left(\frac{2s^2}{\sigma^2 + 2Cs^2} \right), \quad (3.10c)$$

$$\zeta = \left(\frac{\sigma}{\tilde{\sigma}} \right)^{1/2}. \quad (3.10d)$$

The expressions for the chirp and variance can be verified with the well known relations [6, 11, 12, 16]

$$\left(\frac{T_1}{T_0} \right)^2 = \left(1 + \frac{C\beta_2 z}{T_0^2} \right)^2 + \left(\frac{\beta_2 z}{T_0^2} \right)^2, \quad \tilde{C} = C + (1 + C^2) \frac{\beta_2 z}{T_0^2}.$$

Where, within our non-dimensionalization, $T_0 = \sigma T_M$, $T_1 = \tilde{\sigma} T_M$, $z = L_D$, and $\beta_2 z T_0^{-2} = 2s^2 \sigma^{-2}$. In addition to this, the expression for ζ can be validated using

conservation of energy. Within the CFBG energy is conserved, therefore,

$$\int_{-\infty}^{\infty} |A_1|^2 dT = \int_{-\infty}^{\infty} |A_2|^2 dT.$$

Which, after substituting the expressions, reduces to

$$h^2 P \int_{-\infty}^{\infty} e^{-T^2/\sigma^2} dT = h^2 P \zeta \int_{-\infty}^{\infty} e^{-T^2/\tilde{\sigma}^2} dT.$$

These expressions are easily integrated to show that $\sqrt{\pi\sigma^2} = \sqrt{\pi\tilde{\sigma}^2}\zeta^2$, and finally that

$$\zeta = \left(\frac{\sigma}{\tilde{\sigma}}\right)^{1/2}.$$

3.5.2 Equilibrium Shape

Now that the effect of dispersion is known analytically, the system (3.9) can be solved. The out-going variance and chirp can be eliminated from this system of equations using (3.9b) and (3.9c):

$$\tilde{\sigma}^2 = \frac{\sigma^2}{1 - \sigma^2}, \quad \tilde{C} = C \frac{1}{1 - \sigma^2}.$$

Combining this first expression with (3.10a) and expanding, we have

$$\frac{\sigma^4}{1 - \sigma^2} = \sigma^4 + 4C^2 s^4 + 4Cs^2\sigma^2 + 4s^4,$$

or written as a polynomial in σ ,

$$0 = \sigma^6 + 4Cs^2\sigma^4 + (4s^4(C^2 + 1) - 4Cs^2)\sigma^2 - 4s^4(C^2 + 1).$$

The chirp can now be eliminated using (3.10b), the $1 + C^2$ can be reduced in order by noticing that

$$\begin{aligned} C \frac{1}{1 - \sigma^2} &= C + (1 + C^2) \frac{2s^2}{\sigma^2}, \\ 1 + C^2 &= \frac{\sigma^4}{2s^2(1 - \sigma^2)} C. \end{aligned}$$

Furthermore, the chirp can be completely eliminated since

$$C = \frac{\sigma^4}{2s^2(1 - \sigma^2)} \pm \sqrt{\frac{\sigma^8}{16s^4(1 - \sigma^2)^2} - 1} = \frac{\sigma^4 \pm \sqrt{\sigma^8 - 16s^4(1 - \sigma^2)^2}}{4s^2(1 - \sigma^2)}. \quad (3.11)$$

After simplifying algebraically, we arrive at

$$\begin{aligned} 0 &= \sigma^6 \pm \sqrt{\sigma^8 - 16s^4(1 - \sigma^2)^2}(2 - \sigma^2), \\ \frac{\sigma^6}{2 - \sigma^2} &= \mp \sqrt{\sigma^8 - 16s^4(1 - \sigma^2)^2}. \end{aligned}$$

As we shall see, σ is strictly less than 1 at equilibrium, and so, notice that the left hand side of this expression is strictly positive. Therefore, only the negative root of (3.11) will yield a solution. Moreover, Martinez, Fork, and Gordon [39] found two solutions analytically, but showed that one was unstable—consistent with experiments. After squaring each side of this expression we obtain

$$\sigma^{12} = \sigma^8(2 - \sigma^2)^2 - 16s^4(1 - \sigma^2)^2(2 - \sigma^2)^2,$$

which once fully simplified, yields the biquartic equation

$$(\sigma^2)^4 + 4s^4(\sigma^2)^3 - 20s^4(\sigma^2)^2 + 32s^4(\sigma^2) - 16s^4 = 0. \quad (3.12)$$

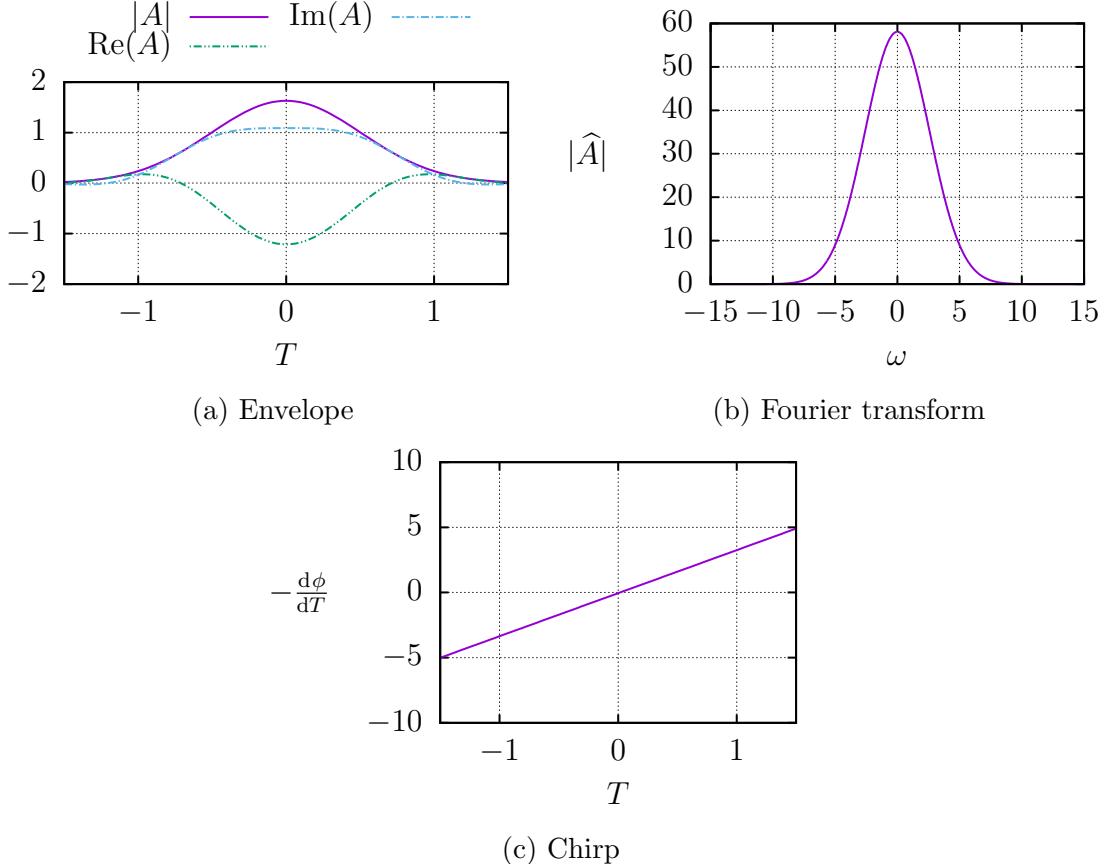


Figure 3.3: Equilibrium state for the parameters $s = 0.15$, $b = 0$, $a = 8000$, and $h = 0.04$.

Since this is a quartic in σ^2 this can be solved analytically; the (positive, real) solution is

$$\sigma^2 = \sqrt{2}s \left(s^6 + 3s^2 + \sqrt{4+s^4} (1+s^4) \right)^{1/2} - s^4 - s^2 \sqrt{4+s^4}. \quad (3.13)$$

A sample solution for the linear model is shown in Figure 3.3. To reiterate, the envelope at equilibrium is Gaussian, and therefore, so too is the Fourier transform. Also, the pulse is linearly chirped.

3.5.3 Asymptotic Expansion of the Variance

While (3.13) is useful, so too is the asymptotic behaviour. We shall investigate the nature of the solution to (3.12) both when $s \rightarrow 0$ (a very short CFBG), as well as when $s \rightarrow \infty$ (a very long CFBG). For ease of notation, (3.12) is rewritten as

$$\frac{1}{4}x^4 = \varepsilon(-x^3 + 5x^2 - 8x + 4), \quad (3.14)$$

where $x = \sigma^2$, $\varepsilon = s^4$, and $\varepsilon \rightarrow 0$. Additionally, suppose x can be expanded as a power series in ε :

$$x = x_0 + x_1\varepsilon^\alpha + x_2\varepsilon^\beta + \dots \quad (3.15)$$

with $0 < \alpha < \beta$. Then, expanding (3.14) using (3.15) and collecting terms, at $\mathcal{O}(1)$,

$$\frac{1}{4}x_0^4 = 0,$$

and so, $x_0 = 0$. Knowing this, we consider how to balance

$$\frac{1}{4}x_1^4\varepsilon^{4\alpha} = \varepsilon(-x_1^3\varepsilon^{3\alpha} + 5x_1^2\varepsilon^{2\alpha} - 8x_1\varepsilon^\alpha + 4).$$

In order to have dominant balance it must be that the left hand side balances with the final term of the right hand side, that is, $4\alpha = 1$, or $\alpha = \frac{1}{4}$. Therefore, at $\mathcal{O}(\varepsilon)$ $x_1^4 = 16$, or

$$(x_1 - 2)(x_1 + 2)(x_1^2 + 4) = 0,$$

and so, $x_1 = \pm 2, \pm 2i$. However, we wish x to be positive and real—recall that $x = \sigma^2$ —thus, we take $x_1 = 2$. Now, the next lowest order of the left hand side must

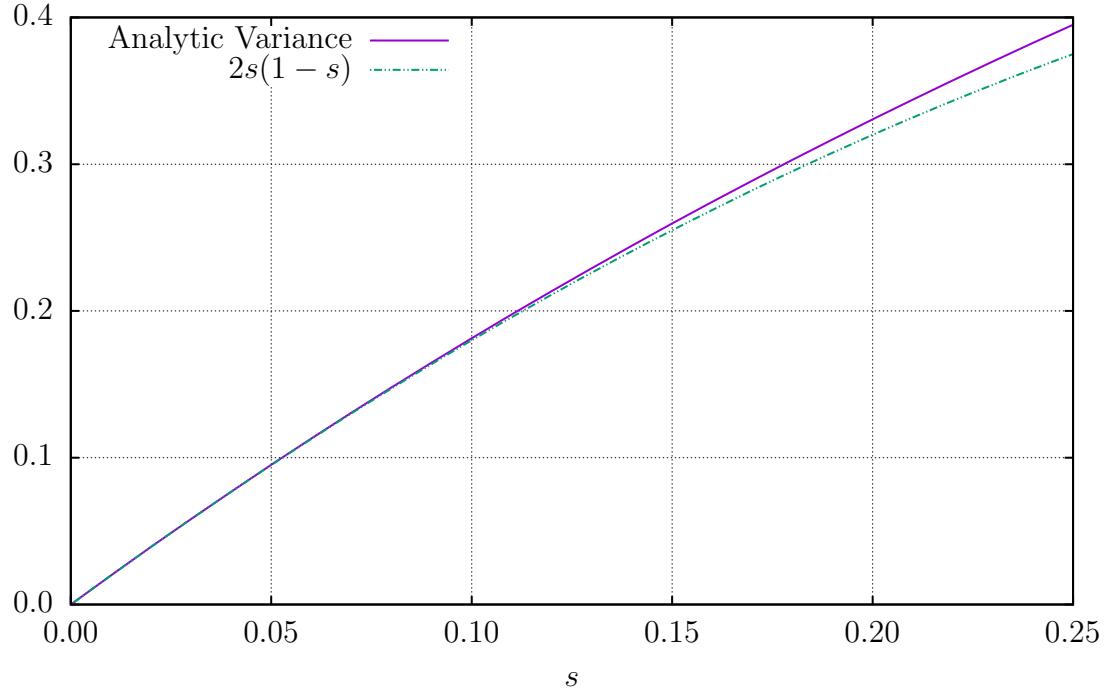


Figure 3.4: Asymptotic expansion of the variance as $s \rightarrow 0$.

be $\varepsilon^{3\alpha+\beta} = \varepsilon^{3/4+\beta}$. As with the previous iteration, this must balance with the lowest order term of the $8x$. Hence, $\frac{3}{4} + \beta = 1 + \alpha = \frac{5}{4}$, thus, $\beta = \frac{1}{2}$. Now, at $\mathcal{O}(\varepsilon^{5/4})$,

$$\frac{1}{4}4x_1^3x_2 = -8x_1,$$

and with the choice of $x_1, x_2 = 2$. Combining these results gives $x = 2\varepsilon^{1/4} - 2\varepsilon^{1/2} + \mathcal{O}(\varepsilon^{3/4})$, or $\sigma^2 = 2s(1 - s) + \mathcal{O}(s^3)$, this is shown in Figure 3.4.

We shall now consider the other limit, when $s \rightarrow \infty$. Using a similar substitution we instead write (3.12) as

$$\frac{1}{4}\varepsilon x^4 = -x^3 + 5x^2 - 8x + 4, \quad (3.16)$$

where $x = \sigma^2$, and $\varepsilon = s^{-4}$ instead so we still have $\varepsilon \rightarrow 0$. Furthermore, we must

make a correction to the series expansion:

$$x = \varepsilon^{-\xi} (x_0 + x_1 \varepsilon^\alpha + x_2 \varepsilon^\beta + \dots). \quad (3.17)$$

The reason for this is because the equation is now singular—when $\varepsilon = 0$ the equation transforms from a quartic to a cubic, losing a root. As with before, we obtain dominant balance when the quartic term balances with the cubic term which occurs when $1 - 4\xi = -3\xi$, or $\xi = 1$. Then to $\mathcal{O}(\varepsilon^{-3})$, expression (3.16) gives

$$x_0^3(x_0 + 4) = 0,$$

we shall choose $x_0 = 0$ —since $x \geq 0$. Continuing, we obtain a dominant balance when $\alpha = 1$. So the next condition occurs at $\mathcal{O}(1)$ where

$$0 = -x_1^3 + 5x_1^2 - 8x_1 + 4 = -(x_1 - 1)(x_1 - 2)^2,$$

so either $x_1 = 1$ or $x_1 = 2$, we shall see in the next step that we can eliminate one of these. The next term obtains dominant balance when the left hand side balances with the linear term of the right hand side, that is, when $\beta = 2$. Now at $\mathcal{O}(\varepsilon)$,

$$\frac{1}{4}x_1^4 = -3x_1^2x_2 + 5 \cdot 2x_1x_2 - 8x_2 = -x_2(3x_1 - 4)(x_1 - 2),$$

where again, the coefficients come from the binomial expansions. From this it is clear that x_2 is unique provided $x_1 \neq 2$ and

$$x_2 = \frac{-x_1^4}{4(3x_1 - 4)(x_1 - 2)}.$$

Thus, $x_1 = 1$, $x_2 = -\frac{1}{4}$, and

$$x = 1 - \frac{1}{4}\varepsilon + x_2\varepsilon^{\gamma-1} + \dots$$

Expanding one final term we find $\gamma = 3$, and to $\mathcal{O}(\varepsilon^2)$,

$$\frac{1}{4}4x_2x_1^3 = -(3x_3x_1^2 + 3x_2^2x_1) + 5(2x_3x_1 + x_2^2) - 8x_3.$$

Knowing the values of x_1 , and x_2 this is simply an arithmetical calculation yielding $x_3 = \frac{3}{8}$. Finally,

$$x = 1 - \frac{1}{4}\varepsilon + \frac{3}{8}\varepsilon^2 + \mathcal{O}(\varepsilon^3),$$

or,

$$\sigma^2 = 1 - \frac{1}{4s^4} + \frac{3}{8s^8} + \mathcal{O}(s^{-12})$$

this approximation is shown in Figure 3.5.

In conclusion, the asymptotic expansions of the observables are

$$\begin{aligned} \sigma^2 &= \begin{cases} 2s(1-s) + \mathcal{O}(s^3) & s \rightarrow 0 \\ 1 - \frac{1}{4s^4} + \frac{3}{8s^8} + \mathcal{O}(s^{-12}) & s \rightarrow \infty \end{cases} & C &= \begin{cases} 1 - s + \frac{1}{2}s^2 + \mathcal{O}(s^3) & s \rightarrow 0 \\ \frac{1}{2s^2} - \frac{3}{8s^6} + \mathcal{O}(s^{-10}) & s \rightarrow \infty \end{cases} \\ \frac{C}{\sigma^2} &= \begin{cases} \frac{1}{2s} - \frac{1}{8}s + \mathcal{O}(s^3) & s \rightarrow 0 \\ \frac{1}{2s^2} - \frac{1}{4s^6} + \mathcal{O}(s^{-10}) & s \rightarrow \infty \end{cases} & \zeta &= \begin{cases} 1 - \frac{1}{2}s + \frac{1}{8}s^2 + \mathcal{O}(s^3) & s \rightarrow 0 \\ \frac{1}{4^{1/4}s} - \frac{3 \cdot 4^{3/4}}{32s^5} + \mathcal{O}(s^{-9}) & s \rightarrow \infty. \end{cases} \end{aligned}$$

The general form of these is as expected. When $s \rightarrow 0$, we effectively remove the CFBG from the cavity. This causes the pulse to be perpetually modulated, and so, the pulse approaches a δ function, and therefore, $\sigma^2 \rightarrow 0$. Moreover, since ζ is the

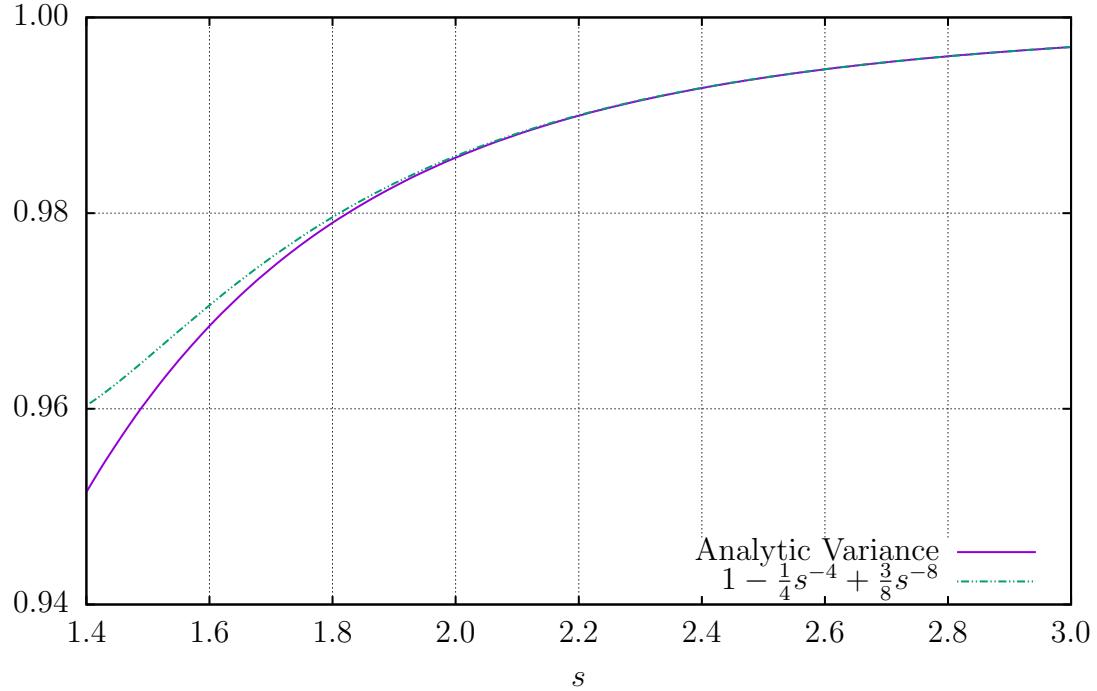


Figure 3.5: Asymptotic expansion of the variance as $s \rightarrow \infty$.

ratio of the amplitude of the pulse before and after dispersion, $\zeta \rightarrow 1$. In the opposite extreme, $s \rightarrow \infty$, we have an incredibly long / powerful CFBG—this causes the pulse to broaden greatly, while conserving its energy. Because of this, the variance approaches unity because the shape is entirely determined by the modulation function. Additionally, $\zeta \rightarrow 0$ because the pulse becomes infinitely wide, and thus, the amplitude becomes infinitesimally small. Finally, because of the expansive nature, the chirp approaches zero because any chirp will be inflated away.

3.5.4 Equilibrium Energy

From (3.9a) the equilibrium energy can be found, as well as the equilibrium peak power. This relation can be simplified by squaring both sides and rearranging to give

$$\frac{1}{h^2\zeta^2}E = W(aEe^E).$$

Then, by taking the exponential of each side, and multiplying by this expression, we obtain³

$$\frac{1}{h^2\zeta^2}E \exp\left(\frac{1}{h^2\zeta^2}E\right) = W(aEe^E) \exp(W(aEe^E)) = aEe^E.$$

Now, this can be written as

$$\begin{aligned} ah^2\zeta^2 &= \exp\left(\frac{1}{h^2\zeta^2}E - E\right), \\ \log(ah^2\zeta^2) &= E\left(\frac{1}{h^2\zeta^2} - 1\right). \end{aligned}$$

The energy of the pulse entering the gain fibre at equilibrium is thus

$$E = \frac{h^2\zeta^2}{1-h^2\zeta^2} \log(ah^2\zeta^2).$$

This expression allows us to determine a restriction on the parameters for a solution to exist—in order for this energy to be positive, $ah^2\zeta^2 > 1$. The energy of the pulses as it enters the optical coupler can now be found, recall from (3.1) that the energy is defined as

$$E = \int_{-\infty}^{\infty} |A|^2 dT;$$

the energy entering the optical coupler is then

$$E_* = \int_{-\infty}^{\infty} |G(A)|^2 dT = \frac{W(aEe^E)}{E} \int_{-\infty}^{\infty} |A|^2 dT = W(aEe^E). \quad (3.18)$$

We can now find the amplitude of the pulse as well, as we have previously found

³By (A.1).

the equilibrium shape. Again, from (3.1), it must be that $E_* = \sqrt{\pi}P\sigma$, or,

$$P = \frac{W(aEe^E)}{\sqrt{\pi}\sigma}. \quad (3.19)$$

The asymptotic expansions when $s \rightarrow 0$ of the energy, and peak power are

$$E_* = \Lambda - \Theta \frac{\Lambda}{1 + \Lambda} s + \mathcal{O}(s^2), \quad P = \frac{\Lambda}{\sqrt{2\pi}s} + \left(\frac{\Lambda}{2} - \Theta \frac{\Lambda}{1 + \Lambda} \right) \sqrt{\frac{s}{2\pi}} + \mathcal{O}(s^{3/2}),$$

respectively, where

$$\Lambda = W\left(\frac{ah^2 \ln(ah^2)}{1 - h^2} \exp\left(\frac{h^2 \ln(ah^2)}{1 - h^2}\right)\right),$$

and

$$\Theta = \frac{(h^2 \ln(ah^2) + 1 - h^2)(\ln(ah^2) + 1 - h^2)}{(1 - h^2)^2 \ln(ah^2)}.$$

Chapter 4

Solution of the Nonlinear Model

We shall now consider the nonlinear case—when $b > 0$. In this case it becomes much too difficult to obtain an analytic result. Recall from (3.8b) that the nonlinearity takes the form

$$F(A) = Ae^{ib|A|^2}.$$

This is a highly nonlinear operator, and attempting to take the Fourier transform of a pulse that has undergone this transformation quickly becomes futile. Instead we must resort to a numerical solution.

4.1 Code

Finding the solution numerically will be done in a similar manner as with the analytic linear solution. Using Python a function is written for each component of the laser cavity given by (3.8), an initial pulse is iteratively passed from function to function in the hopes that a ‘fixed point’ is found. The full Python code can be found in Appendix B.

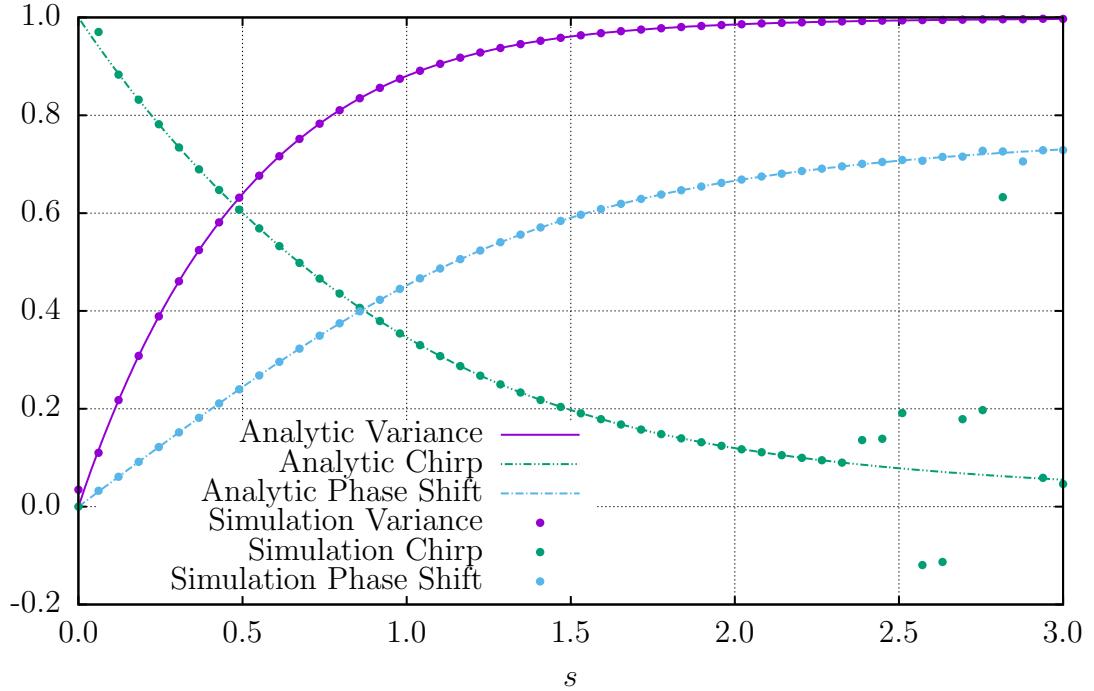


Figure 4.1: Simulation and analytic equilibrium variance, chirp, and phase shift as a function of s . The analytic expressions are given by (3.13), (3.11), (3.10c), respectively.

4.1.1 Validation

Before delving into the nonlinear behaviour of the model, we wish to validate that the code is working as expected—and as a sanity test for the linear solution—by comparing the results of the simulations with $b = 0$ to the results of the linear model in Chapter 3. In the case of the numerical solution, our initial conditions become somewhat important. In all of the following analysis the initial waveform is $\Gamma \operatorname{sech}(2T)e^{i\pi/4}$ normalized so that the initial energy is $E_0 = 0.1$; additionally, $a = 8000$, and $h = 0.04$.

In the first experiment the pulse is allowed to equilibrate for 40 loops of the circuit, we then compare the variance, chirp, and phase shift of the two methods. The results

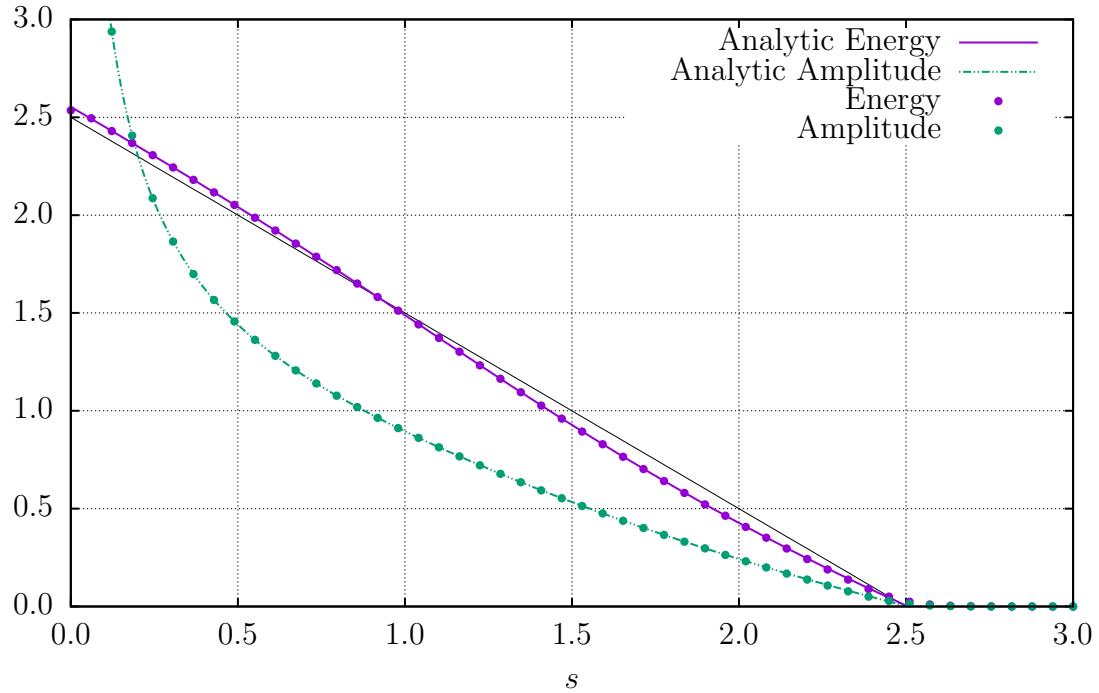


Figure 4.2: Equilibrium energy, and peak power of the pulse as a function of s . The thin black line is to highlight that the energy is *not* linearly dependent on s .

of this are shown in Figure 4.1. For the most part, we see exceptional agreement between the analytic solution and numerical solution for the variance, and phase shift. However, for $s > 2.4$ the chirp from the simulations seems to erratically vary from the analytic solution. The reason for this is quite a simple one.

To investigate this behaviour, we instead turn our attention to the energy, and amplitude of the pulse at equilibrium as shown in Figure 4.2. As with Figure 4.1, there is very good agreement between the two solutions. It is also now clear that at approximately $s = 2.5$ there is too much dispersion—the pulse is no longer sustainable. This is of course a consequence of the condition (3.9a). Since the pulse effectively vanishes after this point, the chirp calculation becomes numerically unstable leading to the wild oscillations.

With the numerical solution yielding the expected results, we are now ready to

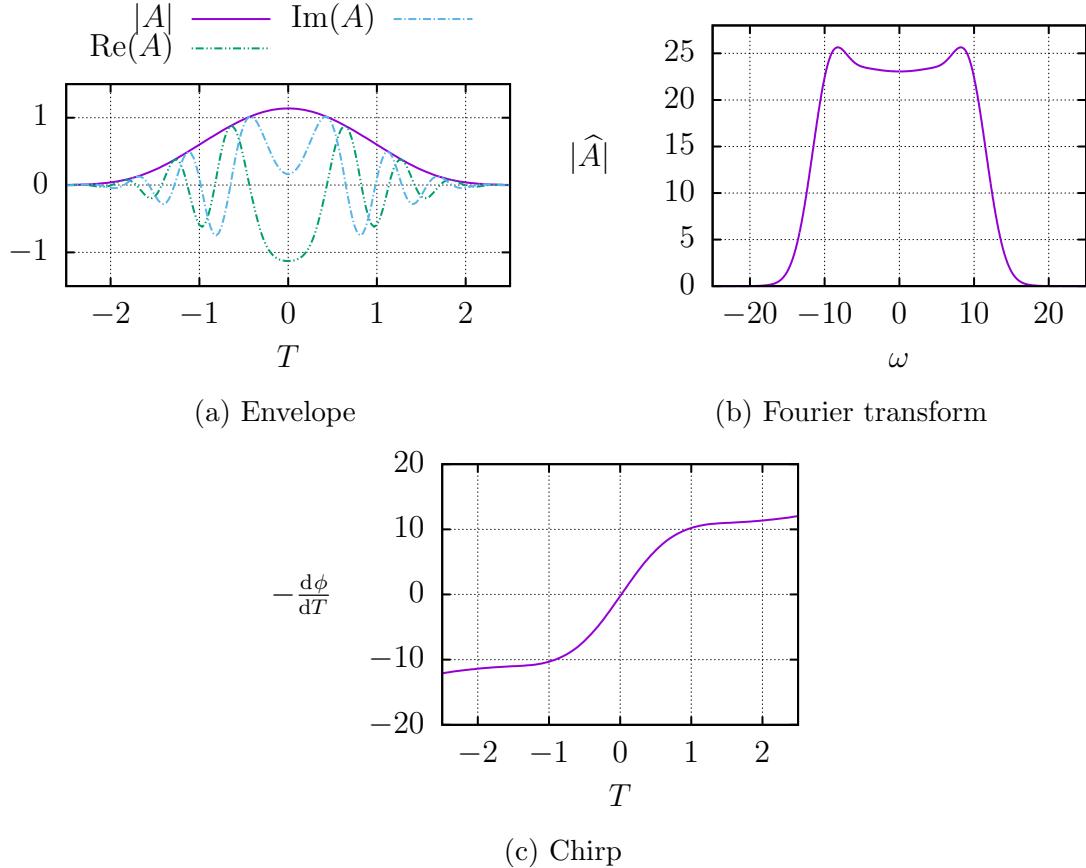


Figure 4.3: Simulation with $s = 0.15$, and $b = 2.1$ after 15 circuits.

dive into the rich structure the nonlinearity adds.

4.2 Nonlinear Model

With the inclusion of the nonlinearity we generally find a similar solution to the linear case. An example of this is shown in Figure 4.3. The envelope of the pulse is unsurprisingly Gaussian-esque, however, it is *not* precisely Gaussian and more closely resembles a generalized Gaussian¹. The fact that the pulse envelope is not Gaussian is further emphasized while examining the Fourier transform of the pulse. If the pulse were a Gaussian, we would expect the Fourier transform to also be a Gaussian [42, 43]. Instead, the magnitude of the Fourier transform has a unique Batman-like shape.

¹A generalized Gaussian has the form $\exp(-t^\alpha)$, with $\alpha > 2$.

This deviation suggests the nonlinearity implants higher frequency oscillations into the pulse—this will be a key observation in the coming subsections. Finally, we shall examine the derivative of the phase—essentially the chirp. Recall in Chapter 3 the chirp was defined as coefficient of $-\frac{1}{2}iT^2$ in the exponential, by taking the negative of the derivative we would expect a linear function with a slope equal to the chirp. In the nonlinear case, this is what we find for moderate values of T . On the other hand, for $|T| > 1$ the relation begins to level off. This is in agreement with experimental results [20, 60, 61]. While the nonlinear model yields similar results as the linear model (Figure 3.3), very interesting variations arise from the nonlinearity.

We shall now take a look at perhaps the most important feature of the pulse—the energy.

4.2.1 Energy

The energy of the pulse is directly related to the output power of the laser, and since this is not as controllable as in a regular laser, it is of great interest. The energy of the pulse at equilibrium² is shown in Figure 4.4. In Figure 4.4a we generally find that the energy is a smooth surface that slowly decays as s , and b increase. In the upper left hand region of the plot this is clearly not the case. The contours show that the energy is very noisy and discontinuous. Perhaps more surprising, as shown in Figure 4.4b, is that this boundary appears to become periodic as b increases around $s = 0.3$. These features are further highlighted in Figure 4.5 by disregarding the energy and only examining whether the pulse is stable or not.

This periodic strip can be explained from the functional nature of the nonlinearity.

²The pulse is passed through each component of the laser 40 times before the energy is computed.

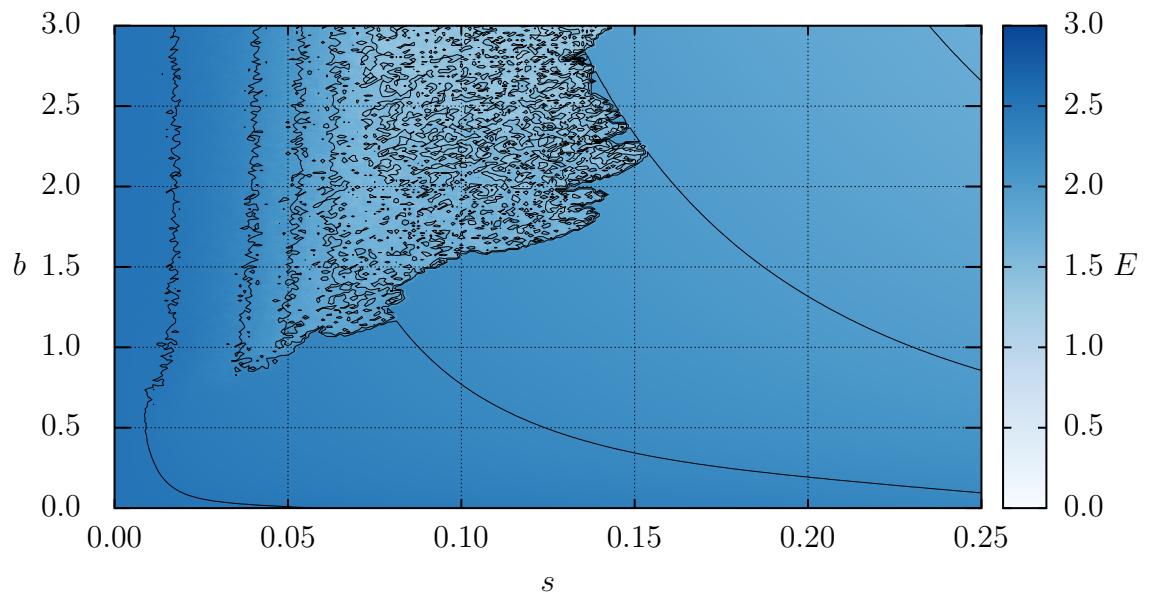
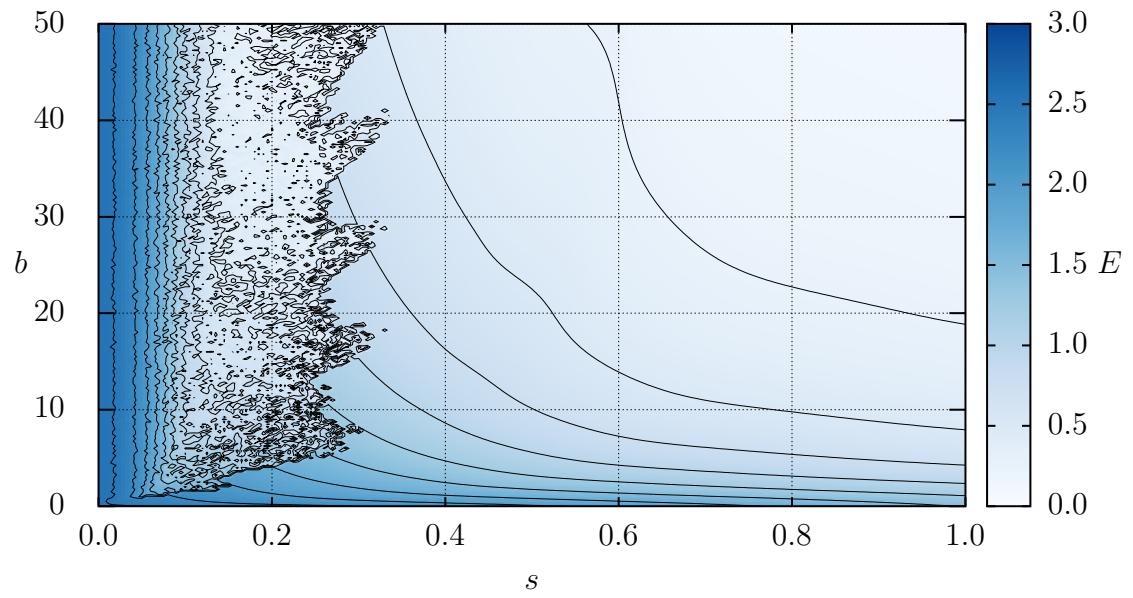
(a) Centralized for typical s , and b values.(b) Energy of the pulse within the s - b plane.

Figure 4.4: Energy of the pulse at equilibrium. The energy is constant along the black lines.

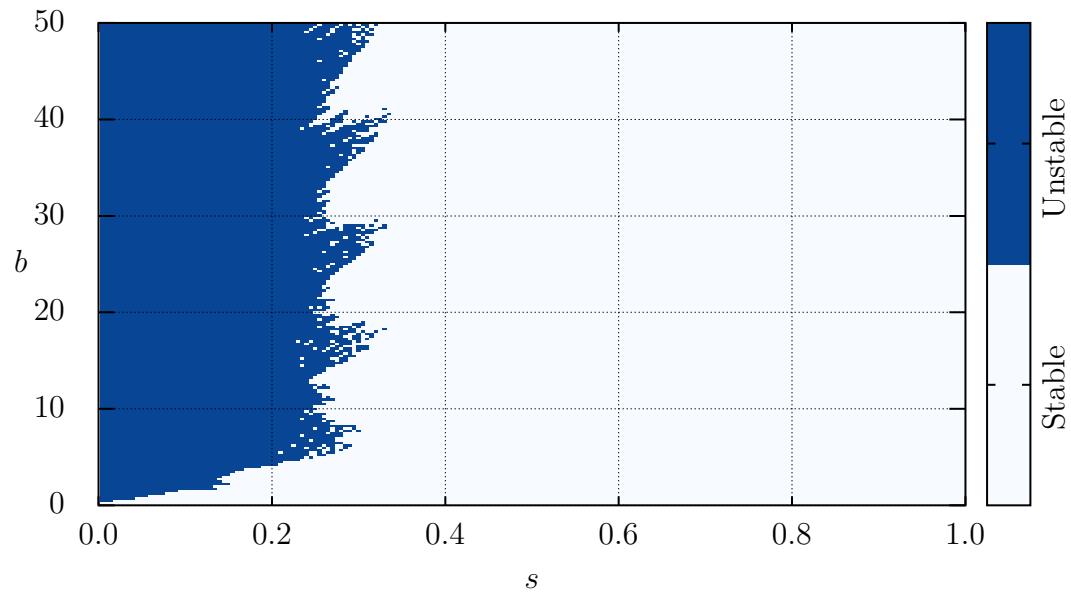
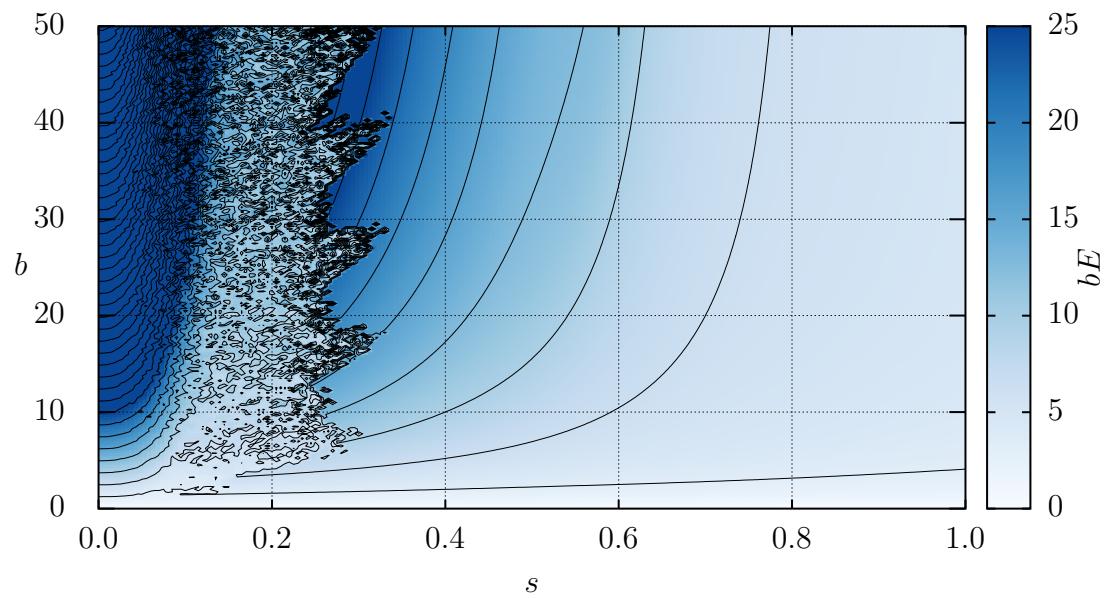
Figure 4.5: Stability of the pulse in the s - b plane.

Figure 4.6: Approximate effective nonlinearity.

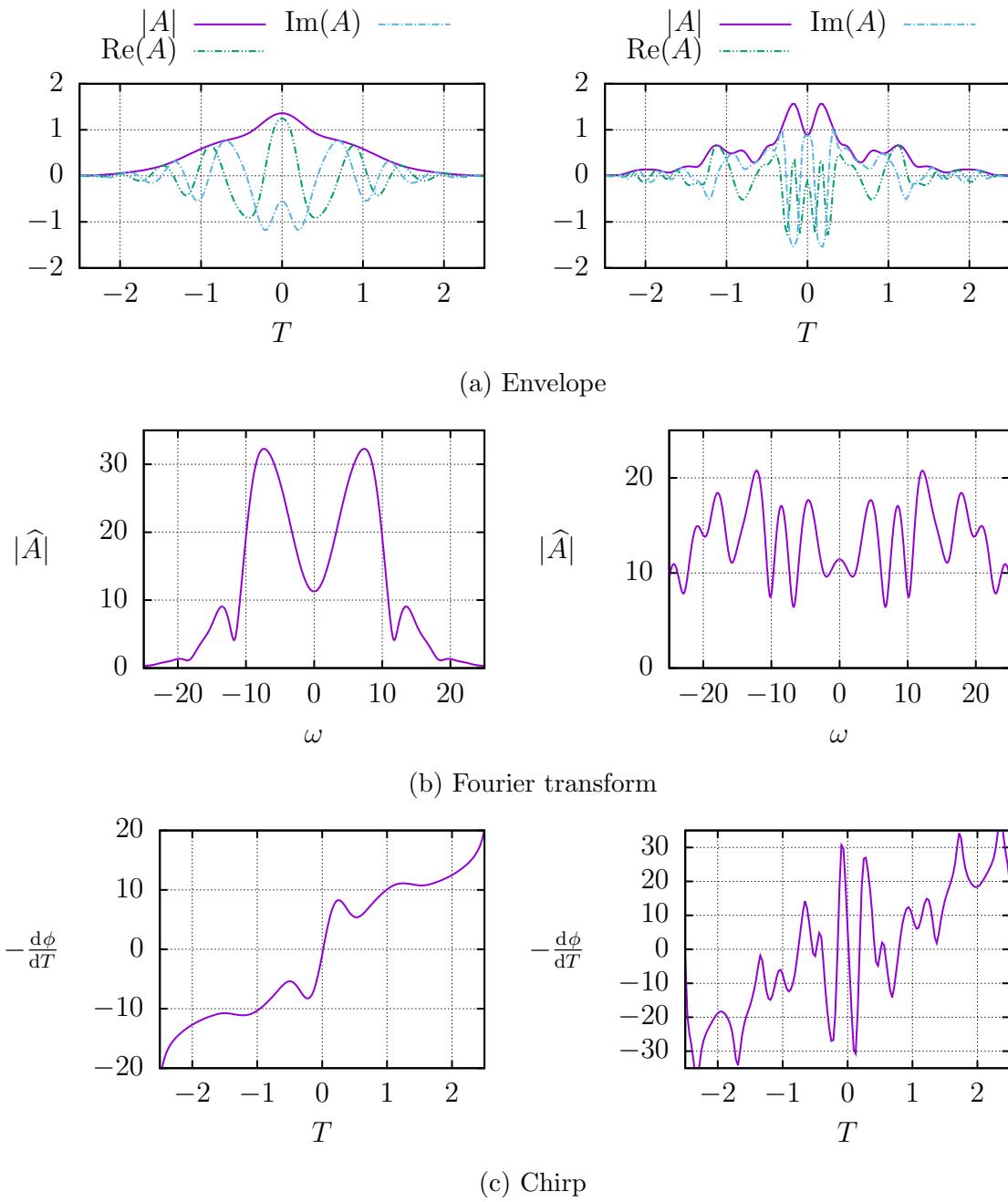
Recall once again that from (3.8b) that the nonlinearity takes the form

$$F(A) = A e^{ib|A|^2}.$$

The strength of the nonlinearity is thus controlled by $b|A|^2$, where $|A|^2$ is the power of the pulse. Although not the greatest, we can approximate the power of the pulse by the energy so that $b|A|^2 \sim bE$, this gives an estimate for what shall be referred to as the effective nonlinearity. Figure 4.6 shows the effective nonlinearity at equilibrium. Notice that as b increases along the strip $0.2 < s < 0.4$, the contours become steeper and steeper. In addition, the contours have been plotted at multiples of π , and exit the boundary at approximately the same places. Thus, the reason for this periodicity is that the strength of the nonlinearity is proportional to the modulus of the effective nonlinearity—the effective nonlinearity passes through the same points repeatedly.

Self-Phase Modulation and Wave Breaking

The noise in the energy exhibited for moderate to large values of b , and small values of s is a phenomenon called *wave breaking* [11, 16, 19, 20, 61]. Wave breaking is not limited to just optics; wave breaking occurs in areas such as plasmas, transmission lines, and fluid dynamics [20]. Wave breaking occurs because the pulse begins to interfere with itself in a way called self-phase modulation [9, 11, 13]. Self-phase modulation occurs because the index of refraction is intensity dependent [6, 9, 13, 20], which leads to additional chirp across the pulse [6, 11, 16, 20]. This in turn causes higher order frequencies to be injected into the pulse [11, 16], as we saw in Figure 4.3. These high frequencies compound with each trip around the cavity becoming parasitic very quickly—Figure 4.7 highlights this. Notice that the difference between this and Figure 4.3 is a difference in b of 0.05—this difference could be as small as adding a few

Figure 4.7: $s = 0.15$, $b = 2.15$, **Left:** 11 loops, **Right:** 16 loops

centimetres more of fibre between the gain and output coupler. The left figures show the pulse after 11 trips around the cavity; in the Fourier transform it is clear that the contributions from higher frequencies has increased—we obtain similar results as in [16, 20]. Additionally, the chirp starts losing its linearity causing it to start becoming unstable; the nature of this instability is again in agreement with [16, 20]. The parasitic nature of the high frequency contributions is evident by examining the right figures. After five additional trips around the cavity, the envelope of the pulse is much more rippled, and the real and imaginary parts become incoherent. Moreover, the Fourier transform has no clear structure and has essentially become noise, and the chirp has grown to be highly oscillatory and unstable. Once the pulse has reached a state such as this, there is no turning back—the envelope, Fourier transform, and chirp never reach a steady equilibrium state.

4.2.2 Convergence

To obtain a better understanding of how the pulse either converges to equilibrium, or diverges to wave breaking, we shall examine the difference between the envelopes of consecutive iterations. More precisely, we compute the error by

$$E = \frac{\| |A_i| - |A_{i-1}| \|_2}{\| A_{i-1} \|_2}, \quad (4.1)$$

where $\| \cdot \|_2$ denotes the $L^2(\mathbb{R})$ norm, which is computed numerically using the trapezoid rule. Notice as well that in the numerator we use the modulus of the pulses, again this is because we are uninterested about the phase shift between iterations. A plot of the error can be found in Figure 4.8, with $i = 100$.

Unsurprisingly, the error is largest in the region where the wave breaks. As men-

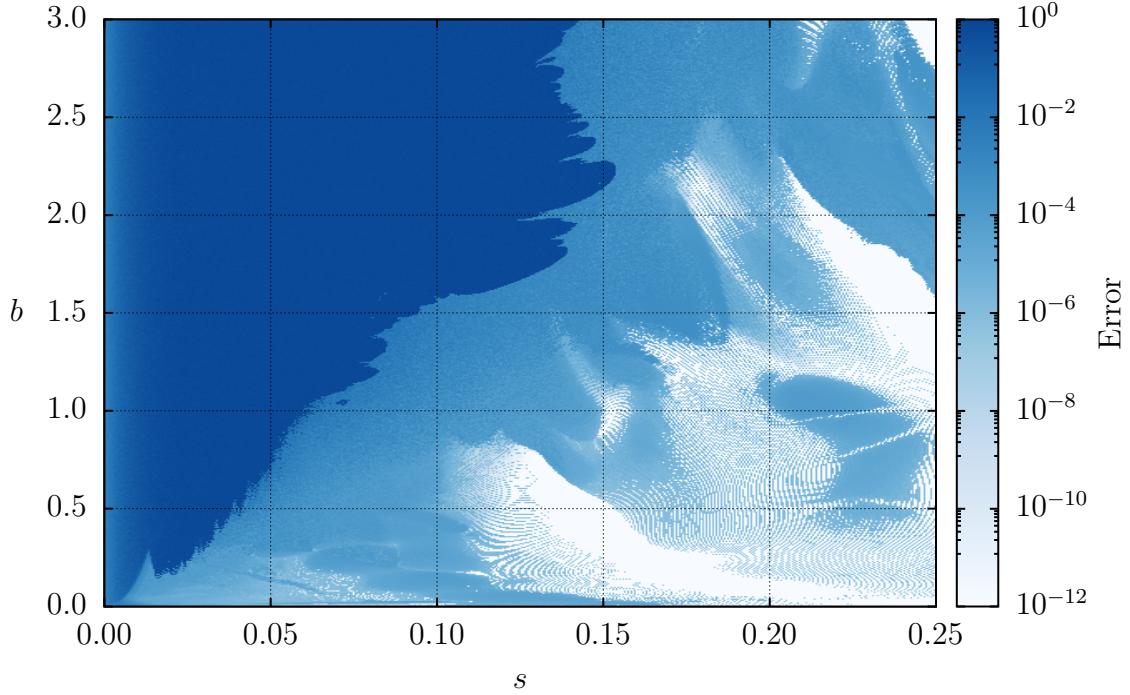


Figure 4.8: Error of the pulse (given by (4.1)) between iterations 99 and 100.

tioned in the previous subsection, the pulse does not reach a stable state in this region. As a consequence the envelope varies violently, which leads to this large error. On the other hand, the region where the energy appeared to be stable does not have a non-zero error—there are a few reasons for this. First, because of the numerical calculation, for all intents and purposes an error less than 10^{-10} can be considered 0. Understanding the other sources of these small errors requires a deeper examination.

The second cause of these errors is because some equilibrium states have a periodicity greater than 1, we redefine (4.1) to be instead

$$E_\Delta = \frac{\| |A_i| - |A_{i-\Delta}| \|_2}{\| A_{i-\Delta} \|_2}, \quad (4.2)$$

to pick up equilibrium states with period Δ , again with $i = 100$. Figure 4.9 shows the error for an assortment of Δ values. There are several key observations to make,

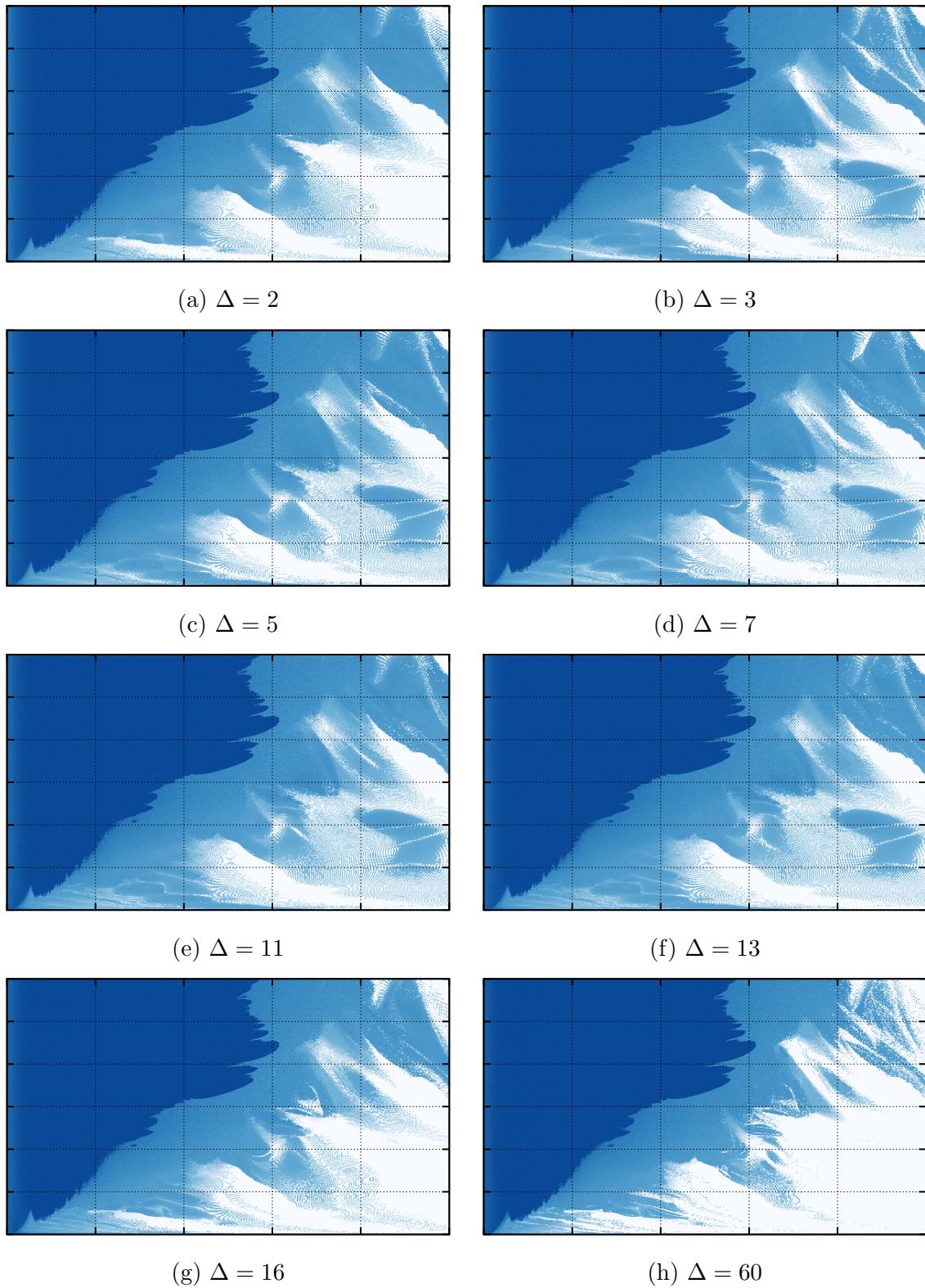


Figure 4.9: Error for various values of Δ , the colour bar and axis labels have been omitted for clarity, the scales and ranges are the same as in Figure 4.8.

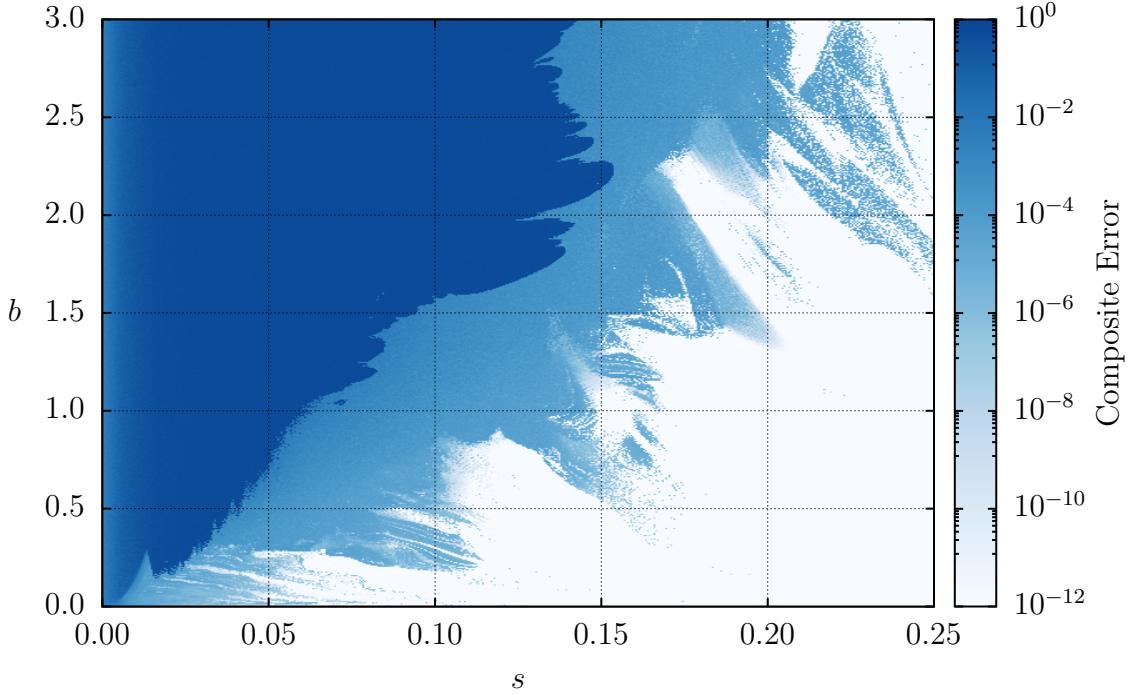


Figure 4.10: Composite error, (4.3), of Figure 4.9.

first, for $\Delta = 2$ the kidney-shaped blob in the lower right corner has vanished, thus, this entire region has a periodicity of 2. Second, for $\Delta = 3$ in the upper right corner a negative triangular-shaped structure emerges. Lastly, $\Delta \in \{5, 7, 11, 13\}$ may not seem to extract any higher order periods, however, there are in fact very small striations in the lower left. So far, the Δ values have been prime since they are the building blocks for composite numbers. However, there are two composite Δ values of interest—the first being 16. Since $16 = 2^4$ it will pick up any periodic behaviour with orders of 2, 4, 8, 16, which are frequently found in bifurcations. Lastly, we chose $\Delta = 60$ for a similar reason: because 60 is highly divisible—it is an abundant number. This extracts orders such as 12 or 30 without having to explicitly run the calculations³.

We now have the data to make a more meaningful estimate of the error between iterations. To compute this composite error we take the minimum of the error from

³In the case of $\Delta = 60$ a value of $i = 150$ is used to ensure sufficient iterations to converge.

each of these calculations:

$$E_c = \min_{\Delta} \{E_{\Delta}\}. \quad (4.3)$$

The intriguing structure of the composite error is shown in Figure 4.10. Compared with the error from Figure 4.8 the lower right region is much more well behaved, that is, the error is a few orders of magnitude lower. A peculiar trait of the band between the unstable and stable regions is that within this area the pulse is in a quasi-stable state—the pulse is stable and has reached a sort of equilibrium, however, the envelope of the pulse has minute variations with no clear period. Additionally, there appears to be no periodicity of the envelope within the wave breaking region, further supporting the claim that the pulse here is completely unstable.

4.2.3 Permutation of Components

The last item we wish to consider is the order in which the components are placed. In Section 3.4 a brief description for the choice of the order was given. We start with the loss component since this coincides with the output; the fibre nonlinearity follows the gain since this is where it has the largest impact; and the loss follows the nonlinearity in an attempt to mitigate its effect. Therefore, the loss is first, and the gain followed by the nonlinearity are last—leaving dispersion and modulation in the middle. We chose to put the dispersion block ahead of the modulator. However, there was no real reason behind this—modulation before dispersion is equally as valid—and in this subsection we explore the effect of modulating the pulse before it passes through the CFBG.

The result of this switch is shown in Figure 4.11. As a whole, unsurprisingly, we

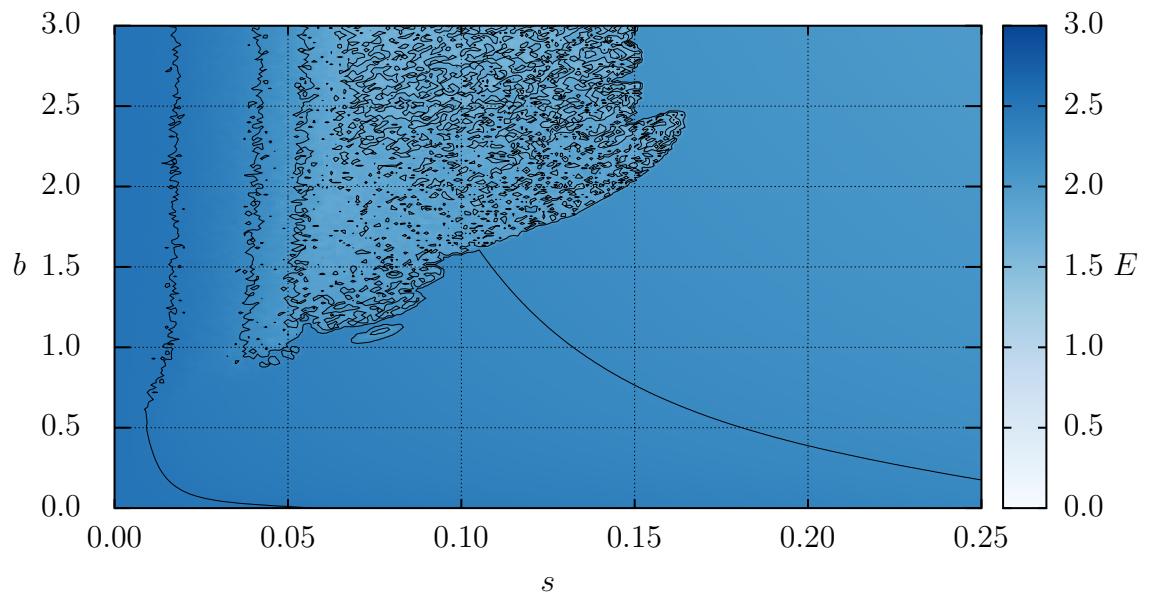
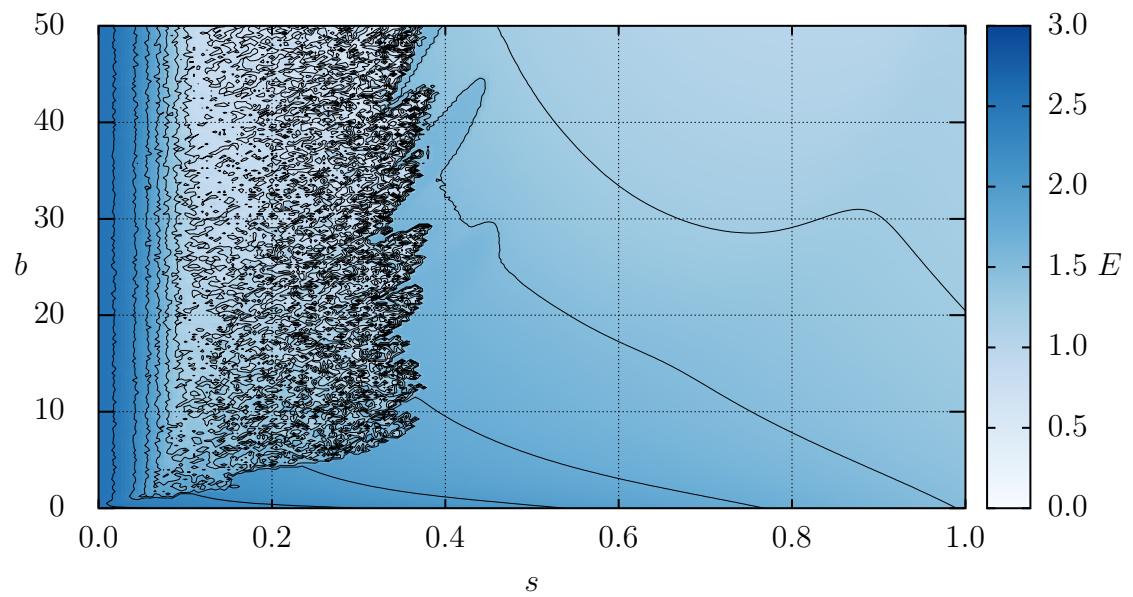
(a) Centralized for typical s , and b values.(b) Energy of the pulse within the s - b plane.

Figure 4.11: Energy of the pulse at equilibrium with the modulation and dispersion blocks switched.

find the same behaviour and structure, however, there are some intriguing differences. Perhaps the most interesting is the small island of instability in Figure 4.11a at around $s = 0.075$, $b = 1.1$ which was not present in the other ordering. Within this island the wave is unstable and breaks, but, curiously there is a small gap between this island and the main unstable region. We find more distinctions when considering the larger area in Figure 4.11b. Again, the structure and periodic nature of the boundary is similar to before, however, this boundary has shifted rightwards to a larger s value. Additionally, within the unstable region the density of the contour lines is much greater—suggesting it is in some sense more chaotic and random than with the components in their original permutation. The final main difference between the two orderings, is that in this case the energy contours are no longer monotonic functions of s . Instead we find a parabolic shape on the top contour, and two lobes on the second contour.

Conclusion

By expanding upon the ideas originally proposed by Cutler [34], and Kuizenga and Siegman [35, 36, 38], we developed a nonlinear functional model for tuneable lasers. By omitting the nonlinearity, we obtain similar linearly chirped Gaussian solutions. However, a consequence of showing Gaussians span $L^2(\mathbb{R})$ is that this model has been solved regardless of the choice of the modulation function.

In contrast, with the inclusion of the nonlinearity we were able to demonstrate wave breaking, and found a peculiar boundary of stability. This phenomenon has been demonstrated in a laboratory setting [11, 16, 19, 20, 61], but, mathematical models have been unable to thus far. The nonlinearity induces self phase modulation which injects higher frequency modes to the pulse. This causes the Fourier transform to become a bimodal distribution instead of Gaussian. As the nonlinearity increases so too does the separation between these two modes. If this effect becomes too strong, the wave becomes incoherent and deformed, and ultimately unsustainable—the wave has broken.

The shape of this boundary between a stable pulse, and an unstable broken pulse is quite intriguing, and surprisingly sharp. At a critical s value, about 0.3 in our

simulations, the boundary becomes periodic—this arises from the nature of the non-linearity. In a sense the nonlinearity is periodic in b , and thus, the stability of the solution is as well.

References

- [1] A. Al-Azzawi, *Fiber Optics: Principles and Advanced Practices*. CRC Press, 2 ed., 2017.
- [2] C. S. Bohun, Y. Cher, L. J. Cummings, P. Howell, T. Mitre, L. Monasse, J. Mueller, and S. Rouillon, “Modelling and Specifying Dispersive Laser Cavities,” in *Sixth Montréal Industrial Problem Solving Workshop*, vol. 2015, pp. 11–25, 2015.
- [3] B. Burgoyne and A. Villeneuve, “Programmable Lasers: Design and Applications,” in *Proc.SPIE*, vol. 7580, pp. 7580–7580–15, 2010.
- [4] S. Yamashita, Y. Nakazaki, R. Konishi, and O. Kusakari, “Wide and Fast Wavelength-Swept Fiber Laser Based on Dispersion Tuning for Dynamic Sensing,” *Journal of Sensors*, vol. 2009, 2009.
- [5] B. Burgoyne, A. Dupuis, and A. Villeneuve, “An Experimentally Validated Discrete Model for Dispersion-Tuned Actively Mode-Locked Lasers,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 20, pp. 390–398, Sept 2014.
- [6] W. T. Silfvast, *Laser Fundamentals*. Cambridge University Press, 2 ed., 2004.
- [7] H. A. Haus, *Waves and Fields in Optoelectronics*. Prentice-Hall, Inc., 1984.
- [8] M. A. Karim, “Electro-Optical and Acousto-Optical Devices,” in Malacara-Hernández and Thompson [62], pp. 409–458. Ch. 11. 2018.

- [9] G. Agrawal, *Fiber-Optic Communication Systems*. John Wiley & Sons, Inc., 3 ed., 2002.
- [10] D. H. Goldstein, “Anisotropic Materials,” in Malacara-Hernández and Thompson [62], pp. 695–726. Ch. 20. 2018.
- [11] G. Agrawal, *Nonlinear Fiber Optics*. Academic Press, 5 ed., 2013.
- [12] M. F. S. Ferreira, *Nonlinear Effects in Optical Fibers*. John Wiley & Sons, Inc., 2011.
- [13] P. C. Becker, N. A. Olsson, and J. R. Simpson, *Erbium-Doped Fiber Amplifiers Fundamentals and Technology*. Academic Press, 1 ed., 1999.
- [14] A. N. Starodoumov, “Optical Fibers and Accessories,” in Malacara-Hernández and Thompson [62], pp. 633–676. Ch. 18. 2018.
- [15] L. Dong, M. J. Cole, A. D. Ellis, M. Durkin, M. Ibsen, V. Gusmeroli, and R. Laming, “40 Gbit/s 1.55 μm Transmission Over 109 km of Non-Dispersion Shifted Fibre with Long Continuously Chirped Fibre Gratings,” in *Conference on Optical Fiber Communications*, Optical Society of America, 1997.
- [16] D. Anderson, M. Desaix, M. Lisak, and M. L. Quiroga-Teixeiro, “Wave Breaking in Nonlinear-Optical Fibers,” *J. Opt. Soc. Am. B*, vol. 9, pp. 1358–1361, Aug 1992.
- [17] B. Burgoyne, N. Godbout, and S. Lacroix, “Nonlinear Pulse Propagation in Optical Fibers Using Second Order Moments,” *Opt. Express*, vol. 15, pp. 10075–10090, Aug 2007.
- [18] E. Desurvire, *Erbium-Doped Fiber Amplifiers Principles and Applications*. John Wiley & Sons, Inc., 2002.

- [19] C. Finot, B. Kibler, L. Provost, and S. Wabnitz, “Beneficial Impact of Wave-Breaking for Coherent Continuum Formation in Normally Dispersive Nonlinear Fibers,” *J. Opt. Soc. Am. B*, vol. 25, pp. 1938–1948, Nov 2008.
- [20] J. E. Rothenberg, “Femtosecond Optical Shocks and Wave Breaking in Fiber Propagation,” *J. Opt. Soc. Am. B*, vol. 6, pp. 2392–2401, Dec 1989.
- [21] J. R. Ockendon and W. R. Hodgkins, eds., *Moving Boundary Problems in Heat Flow and Diffusion*. Oxford University Press, Mar 1975.
- [22] J. Peng, H. Luo, and L. Zhan, “In-Cavity Soliton Self-Frequency Shift Ultrafast Fiber Lasers,” *Opt. Lett.*, vol. 43, pp. 5913–5916, Dec 2018.
- [23] O. V. Shtyrina, A. V. Ivanenko, I. A. Yarutkina, A. V. Kemmer, A. S. Skidin, S. M. Kobtsev, and M. P. Fedoruk, “Experimental Measurement and Analytical Estimation of the Signal Gain in an Er-Doped Fiber,” *J. Opt. Soc. Am. B*, vol. 34, pp. 227–231, Feb 2017.
- [24] I. Yarutkina, O. Shtyrina, M. Fedoruk, and S. Turitsyn, “Numerical Modeling of Fiber Lasers with Long and Ultra-long Ring Cavity,” *Opt. Express*, vol. 21, pp. 12942–12950, May 2013.
- [25] F. Kärtner, “Lecture Notes in Ultrafast Optics.” Online, Spring 2005. Massachusetts Institute of Technology: MIT OpenCourseWare.
- [26] H. A. Haus, “A Theory of Forced Mode Locking,” *IEEE Journal of Quantum Electronics*, vol. 11, pp. 323–330, July 1975.
- [27] H. A. Haus, “Laser Mode Locking with Addition of Nonlinear Index,” *IEEE Journal of Quantum Electronics*, vol. 22, pp. 325–331, Feb 1986.
- [28] H. A. Haus, “Analytic Theory of Additive Pulse and Kerr Lens Mode Locking,” *IEEE Journal of Quantum Electronics*, vol. 28, pp. 2086–2096, Oct 1992.

- [29] H. A. Haus, “Mode-Locking of Lasers,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 6, pp. 1173–1185, Nov 2000.
- [30] K. Tamura and M. Nakazawa, “Dispersion-Tuned Harmonically Mode-Locked Fiber Ring Laser for Self-Synchronization to an External Clock,” *Opt. Lett.*, vol. 21, pp. 1984–1986, Dec 1996.
- [31] N. G. Usechak and G. P. Agrawal, “Rate-Equation Approach for Frequency-Modulation Mode Locking using the Moment Method,” *J. Opt. Soc. Am. B*, vol. 22, pp. 2570–2580, Dec 2005.
- [32] H. A. Haus, “Theory of Soliton Stability in Asynchronous Modelocking,” *Journal of Lightwave Technology*, vol. 14, pp. 622–627, April 1996.
- [33] H. A. Haus, J. G. Fujimoto, and E. P. Ippen, “Structures for Additive Pulse Mode Locking,” *J. Opt. Soc. Am. B*, vol. 8, pp. 2068–2076, Oct 1991.
- [34] C. C. Cutler, “The Regenerative Pulse Generator,” in *Proceedings of the IRE*, IEEE, Feb 1955.
- [35] A. E. Siegman and D. J. Kuizenga, “Simple Analytic Expressions for AM and FM Modelocked Pulses in Homogenous Lasers,” *Appl. Phys. Lett.*, pp. 181–182, Mar 1969.
- [36] D. J. Kuizenga and A. E. Siegman, “FM and AM Mode Locking of the Homogeneous Laser - Part I: Theory,” *IEEE Journal of Quantum Electronics*, vol. 6, pp. 694–708, Nov 1970.
- [37] D. J. Kuizenga and A. E. Siegman, “FM and AM Mode Locking of the Homogeneous Laser - Part II: Experimental Results in a Nd:YAG Laser with Internal FM Modulation,” *IEEE Journal of Quantum Electronics*, vol. 6, pp. 709–715, Nov 1970.

- [38] D. J. Kuizenga and A. E. Siegman, “FM-Laser Operation of the Nd:YAG Laser,” *IEEE Journal of Quantum Electronics*, vol. 6, pp. 673–677, Nov 1970.
- [39] O. E. Martinez, R. L. Fork, and J. P. Gordon, “Theory of Passively Mode-Locked Lasers Including Self-Phase Modulation and Group-Velocity Dispersion,” *Opt. Lett.*, vol. 9, pp. 156–158, May 1984.
- [40] O. E. Martinez, R. L. Fork, and J. P. Gordon, “Theory of Passively Mode-Locked Lasers for the Case of a Nonlinear Complex-Propagation Coefficient,” *J. Opt. Soc. Am. B*, vol. 2, pp. 753–760, May 1985.
- [41] A. M. Dunlop, W. J. Firth, and E. M. Wright, “Pulse Shapes and Stability in Kerr and Active Mode-Locking (KAML),” *Opt. Express*, vol. 2, pp. 204–211, Mar 1998.
- [42] L. Debnath and D. Bhatta, *Integral Transforms and Their Applications*. Chapman & Hall/CRC Press, 2 ed., 2007.
- [43] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*. Academic Press, 7 ed., 2007.
- [44] N. M. Litchinitser, B. J. Eggleton, and D. B. Patterson, “Fiber Bragg Gratings for Dispersion Compensation in Transmission: Theoretical Model and Design Criteria for Nearly Ideal Pulse Recompression,” *Journal of Lightwave Technology*, vol. 15, pp. 1303–1313, Aug 1997.
- [45] J. B. Conway, *A Course in Functional Analysis*, vol. 96 of *Graduate Texts in Mathematics*. Springer-Verlag New York Inc., 1 ed., 1985.
- [46] R. Courant and D. Hilbert, *Methoden der Mathematischen Physik*, vol. I. Julius Springer, 1924.
- [47] J. Teuwen, “A Cornucopia of Hermite Polynomials,” *Research Gate*, March 2016.

- [48] U. W. Hochstrasser, “Orthogonal Polynomials,” in Abramowitz and Stegun [63], pp. 771–802. Ch. 22. 1972.
- [49] E. Kreyszig, *Introductory Functional Analysis with Applications*. John Wiley & Sons, Inc., 1978.
- [50] G. Szegö, *Orthogonal Polynomials*, vol. XXIII of *American Mathematical Society Colloquium Publications*. American Mathematical Society, 4 ed., 1975.
- [51] D. H. Griffel, *Applied Functional Analysis*. Dover Publications, Inc., 2 ed., 2002.
- [52] A. N. Kolmogorov and S. V. Fomin, *Elements of the Theory of Functions and Functional Analysis*, vol. 2. Graylock Press, 1 ed., 1960.
- [53] J. C. P. Miller, “Parabolic Cylinder Functions,” in Abramowitz and Stegun [63], pp. 685–720. Ch. 19. 1972.
- [54] N. Higson and J. Roe, “Operator Algebras,” in Gowers [64], pp. 510–523. Ch. IV.15. 2008.
- [55] C. Calcaterra, “Linear Combinations of Gaussians with a Single Variance are Dense in L^2 ,” in *Proceedings of the World Congress on Engineering*, vol. II, 2008.
- [56] C. Calcaterra and A. Boldt, “Approximating with Gaussians,” *ArXiv*, May 2008. arXiv:0805.3795.
- [57] D. Zwillinger, *Handbook of Differential Equations*. Academic Press, 2 ed., 1992.
- [58] B. Burgoyne. Private Communication, 2018.
- [59] S. Li and K. T. Chan, “Electrical Wavelength Tunable and Multiwavelength Actively Mode-Locked Fiber Ring Laser,” *Applied Physics Letters*, vol. 72, no. 16, pp. 1954–1956, 1998.

- [60] Q. Chen, N. Lu, and F. Jiang, “Characterization of the Dispersion of Chirped Fiber Bragg Grating Through Fourier Transform Spectrometry Method,” in *Proc. SPIE*, vol. 6837, pp. 6837–6837–8, 2008.
- [61] W. J. Tomlinson, R. H. Stolen, and A. M. Johnson, “Optical Wave Breaking of Pulses in Nonlinear Optical Fibers,” *Opt. Lett.*, vol. 10, pp. 457–459, Sept 1985.
- [62] D. Malacara-Hernández and B. J. Thompson, eds., *Advanced Optical Instruments and Techniques*, vol. 2 of *Handbook of Optical Engineering*. CRC Press, 2 ed., 2018.
- [63] M. Abramowitz and I. A. Stegun, eds., *Handbook of Mathematical Functions*. United States Government Publishing Office, 10 ed., 1972.
- [64] T. Gowers, ed., *The Princeton Companion to Mathematics*. Princeton University Press, 2008.
- [65] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, “On the Lambert W function,” *Advances in Computational Mathematics*, vol. 5, Dec 1996.

Appendix A

The Lambert W Function

The Lambert W function is defined to be the inverse of the function $f(x) = xe^x$; its graph is shown in Figure A.1. In other words, if $z = xe^x$ then $x = W(z)$. Notice that by combining these relations we obtain the identities

$$z = W(z)e^{W(z)}, \quad x = W(xe^x). \quad (\text{A.1})$$

This function is called the Lambert W function because it is the logarithm of a special instance of Lambert's series—the letter W is used because of the work done by E. M. Wright [65].

Notice that the original function, $f(x) = xe^x$, is *not* injective, and as a consequence, the W function is multi-valued on the interval $[-1/e, 0]$. To alleviate this, occasionally the branch $W(x) \geq -1$ is denoted W_0 and is called the principal or upper branch, whereas the branch $W(x) < -1$ is denoted W_{-1} and is called the lower branch. However, in this work the W function will only take positive real values and so this distinction is not needed.

The Lambert W function has applications in various areas of math and physics [65]

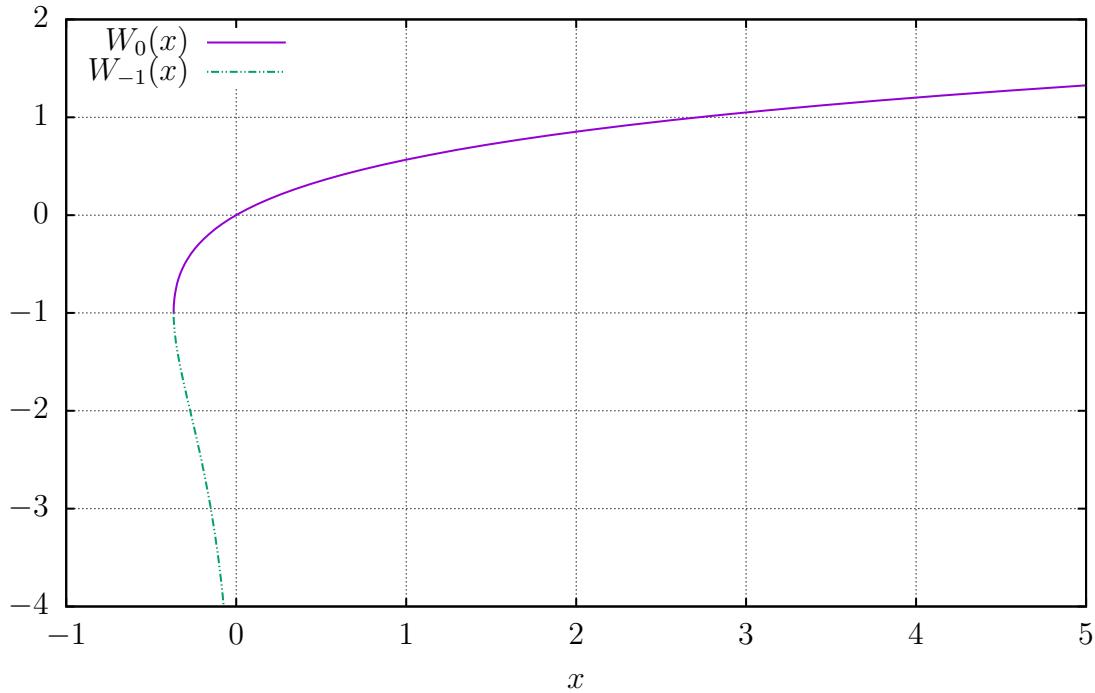


Figure A.1: The two branches of the Lambert W function.

including:

- Jet fuel problems
- Combustion problems
- Enzyme kinetics problems
- Linear constant coefficient differential delay equations
- Volterra equations.

Primarily, the W function arises when solving iterated exponentiation or certain algebraic equations. For example, consider the equation $z = x^x$. By taking the logarithm of each side we have

$$\log z = x \log x,$$

$$= \log x e^{\log x},$$

which after applying the W function reduces to $W(\log z) = \log x$ by (A.1). Finally, x as a function of z can be written as $x = \exp(W(\log z))$.

B

Appendix

Code

```
1 #####  
2 #          Brady Metherall  
3 #          MSc Thesis  
4 #####  
5 ...  
6 This is the code used for my MSc thesis. This code  
7 is separated into four parts each described below.  
8 ...  
9  
10 #####  
11 #      Part I  
12 #####  
13 ...  
14 Function definitions and initialization  
15 ...  
16  
17 import numpy as np
```

```
18 import scipy.special.lambertw as W
19 import matplotlib.pyplot as plt
20 import time
21 from scipy.optimize import curve_fit
22
23 def func(x, a, b):
24     return a * np.exp(-x**2 / (2 * b**2))
25
26 def Energy(A, dx):
27     return np.trapz(np.real(A * np.conj(A)), dx = dx)
28
29 # Functions for each component
30 def Gain(A, E, a = 8000):
31     return np.real(np.sqrt(W(a * E * np.exp(E)) / E)) * A
32
33 def Loss(A, h = 0.04):
34     return h * A
35
36 def Mod(A, T):
37     return np.exp(-T**2 / 2) * A
38
39 def Fibre(A, b = 1.0):
40     return np.exp(1j * b * np.abs(A)**2) * A
41
42 def Disp(A, T, s = 0.1):
43     F = np.fft.fft(A)
44     F = F * (np.abs(F) > 10**-4) # Numerical stability
```

```

45      dw = np.pi / T[-1]
46      w = np.fft.fftfreq(len(A)) * len(A) * dw
47      return np.fft.ifft(F * np.exp(1j * w**2 * s**2))
48
49 # 1 round trip
50 def Loop(A, T, dx, s, b, switch = False):
51     A = Loss(A)
52     if not switch:
53         A = Disp(A, T, s)
54         A = Mod(A, T)
55     else:
56         A = Mod(A, T)
57         A = Disp(A, T, s)
58     A = Gain(A, Energy(A, dx))
59     A = Fibre(A, b)
60     return A
61
62 N = 25 # Number of loops of the circuit
63 p = 2**12 # Number of points in the discretization
64 width = 64 # Size of window
65 E0 = 0.1 # Initial energy
66
67 # Initialization
68 T = np.linspace(-width, width, p, endpoint = False)
69 dx = T[1] - T[0]
70 A0 = 1 / np.cosh(2 * T) * np.exp(1j * np.pi / 4)
71 A0 = np.sqrt(E0 / Energy(A0, dx)) * A0 # Normalize

```

```
72 E = np.zeros(N)
73 data = np.zeros((2 * N, p))
74 A = A0
75
76 part = 2 # Select which part of the code to run
77
78 if part == 2:
79     #####
80     # Part II
81     #####
82     ,
83     On the fly animation of single realizations
84     ,
85     plt.ion()
86     fig = plt.figure()
87     ax = fig.add_subplot(111)
88     line1, = ax.plot(T, np.real(A), 'r-', label = 'Real')
89     line2, = ax.plot(T, np.imag(A), 'b-', label = 'Imaginary',
90
91     )
92     dw = np.pi / T[-1]
93     line3, = ax.plot(T, np.abs(A), 'g-', label = 'Magnitude')
94     fig.canvas.draw()
95     fig.canvas.flush_events()
96
97     plt.legend()
```

```
98     plt.xlim(-2, 2)
99     plt.ylim(-4, 4)
100
101    # N round trips of the laser
102    for i in range(N):
103        # Animate the plot
104        line1.set_ydata(np.real(A))
105        line2.set_ydata(np.imag(A))
106        line3.set_ydata(np.abs(A))
107        fig.canvas.draw()
108        fig.canvas.flush_events()
109        #time.sleep(2)
110        print i
111
112        A = Loop(A, T, dx, 0.15, 0.5)
113        E[i] = Energy(A, dx)
114        #time.sleep(0.5)
115
116        #np.savetxt('E.dat', E)
117
118        dw = np.pi / T[-1]
119        w = np.fft.freq(len(A)) * len(A) * dw
120
121        print np.trapz(T**2 * abs(A), dx=dx) / np.trapz(abs(A),
122                                                       dx=dx)
123        #np.savetxt('Linear_Solution.dat', np.vstack((T, np.real(
```

```
A) , np.imag(A) , np.abs(A) , w, np.abs(np.fft.fft(A)) , np.
angle(A))).T)

124

125 elif part == 3:
126     #####
127     # Part III
128     #####
129     ''
130     Run the simulation for an nxn grid in s-b space
131     ''
132     n = 501
133     #z = np.zeros((n**2, 4))
134     zoom = True
135     step = 1
136
137     if zoom:
138         s = np.linspace(0, 0.25, num = n)
139         #b = np.logspace(3.0, 5.0, num = n)
140         b = np.linspace(0, 3, num = n)
141     else:
142         s = np.linspace(0, 1, num = n)
143         #b = np.logspace(3.0, 6.0, num = n)
144         b = np.linspace(0, 50, num = n)
145
146     filename = 'Step' + str(step) + '.dat'
147     open(filename, 'w').close()
148     f = open(filename, 'ab')
```

```

149
150     for k in range(n):
151         print k
152         z = np.zeros((n, 7))
153         for j in range(n):
154             A0 = 1 / np.cosh(2 * T) * np.exp(1j * np.pi / 4)
155             A0 = np.sqrt(E0 / Energy(A0, dx)) * A0 #
156             Normalize
157             A = A0
158             flag = [0, 0, 0]
159             itera = np.array([100, 100, 100])
160             for i in range(100 / step + 1):
161                 old = np.abs(A)
162                 A = Loop(A, T, dx, s[j], b[k], switch =
163                           False)
164                 new = np.abs(A)
165                 if (flag[0] == 0) and (np.sqrt(np.trapz((old -
166                     - new)**2, dx = dx)) < 10**-3): # L2
167                     itera[0] = i
168                     flag[0] = 1
169                     elif (flag[1] == 0) and (np.trapz(np.abs(old -
170                         - new), dx = dx) < 10**-3): # L1
171                         itera[1] = i
172                         flag[1] = 1
173                         elif (flag[2] == 0) and (np.max(np.abs(old -
174                             - new)) < 10**-3): # Infinity
175                             itera[2] = i
176                             flag[2] = 1
177
178             if flag[0] == 1 and flag[1] == 1 and flag[2] == 1:
179                 break
180
181         if itera[0] > 100:
182             itera[0] = 100
183
184         if itera[1] > 100:
185             itera[1] = 100
186
187         if itera[2] > 100:
188             itera[2] = 100
189
190         if itera[0] < 100:
191             itera[0] = 100
192
193         if itera[1] < 100:
194             itera[1] = 100
195
196         if itera[2] < 100:
197             itera[2] = 100
198
199         if itera[0] < 100:
200             itera[0] = 100
201
202         if itera[1] < 100:
203             itera[1] = 100
204
205         if itera[2] < 100:
206             itera[2] = 100
207
208         if itera[0] < 100:
209             itera[0] = 100
210
211         if itera[1] < 100:
212             itera[1] = 100
213
214         if itera[2] < 100:
215             itera[2] = 100
216
217         if itera[0] < 100:
218             itera[0] = 100
219
220         if itera[1] < 100:
221             itera[1] = 100
222
223         if itera[2] < 100:
224             itera[2] = 100
225
226         if itera[0] < 100:
227             itera[0] = 100
228
229         if itera[1] < 100:
230             itera[1] = 100
231
232         if itera[2] < 100:
233             itera[2] = 100
234
235         if itera[0] < 100:
236             itera[0] = 100
237
238         if itera[1] < 100:
239             itera[1] = 100
240
241         if itera[2] < 100:
242             itera[2] = 100
243
244         if itera[0] < 100:
245             itera[0] = 100
246
247         if itera[1] < 100:
248             itera[1] = 100
249
250         if itera[2] < 100:
251             itera[2] = 100
252
253         if itera[0] < 100:
254             itera[0] = 100
255
256         if itera[1] < 100:
257             itera[1] = 100
258
259         if itera[2] < 100:
260             itera[2] = 100
261
262         if itera[0] < 100:
263             itera[0] = 100
264
265         if itera[1] < 100:
266             itera[1] = 100
267
268         if itera[2] < 100:
269             itera[2] = 100
270
271         if itera[0] < 100:
272             itera[0] = 100
273
274         if itera[1] < 100:
275             itera[1] = 100
276
277         if itera[2] < 100:
278             itera[2] = 100
279
280         if itera[0] < 100:
281             itera[0] = 100
282
283         if itera[1] < 100:
284             itera[1] = 100
285
286         if itera[2] < 100:
287             itera[2] = 100
288
289         if itera[0] < 100:
290             itera[0] = 100
291
292         if itera[1] < 100:
293             itera[1] = 100
294
295         if itera[2] < 100:
296             itera[2] = 100
297
298         if itera[0] < 100:
299             itera[0] = 100
300
301         if itera[1] < 100:
302             itera[1] = 100
303
304         if itera[2] < 100:
305             itera[2] = 100
306
307         if itera[0] < 100:
308             itera[0] = 100
309
310         if itera[1] < 100:
311             itera[1] = 100
312
313         if itera[2] < 100:
314             itera[2] = 100
315
316         if itera[0] < 100:
317             itera[0] = 100
318
319         if itera[1] < 100:
320             itera[1] = 100
321
322         if itera[2] < 100:
323             itera[2] = 100
324
325         if itera[0] < 100:
326             itera[0] = 100
327
328         if itera[1] < 100:
329             itera[1] = 100
330
331         if itera[2] < 100:
332             itera[2] = 100
333
334         if itera[0] < 100:
335             itera[0] = 100
336
337         if itera[1] < 100:
338             itera[1] = 100
339
340         if itera[2] < 100:
341             itera[2] = 100
342
343         if itera[0] < 100:
344             itera[0] = 100
345
346         if itera[1] < 100:
347             itera[1] = 100
348
349         if itera[2] < 100:
350             itera[2] = 100
351
352         if itera[0] < 100:
353             itera[0] = 100
354
355         if itera[1] < 100:
356             itera[1] = 100
357
358         if itera[2] < 100:
359             itera[2] = 100
360
361         if itera[0] < 100:
362             itera[0] = 100
363
364         if itera[1] < 100:
365             itera[1] = 100
366
367         if itera[2] < 100:
368             itera[2] = 100
369
370         if itera[0] < 100:
371             itera[0] = 100
372
373         if itera[1] < 100:
374             itera[1] = 100
375
376         if itera[2] < 100:
377             itera[2] = 100
378
379         if itera[0] < 100:
380             itera[0] = 100
381
382         if itera[1] < 100:
383             itera[1] = 100
384
385         if itera[2] < 100:
386             itera[2] = 100
387
388         if itera[0] < 100:
389             itera[0] = 100
390
391         if itera[1] < 100:
392             itera[1] = 100
393
394         if itera[2] < 100:
395             itera[2] = 100
396
397         if itera[0] < 100:
398             itera[0] = 100
399
400         if itera[1] < 100:
401             itera[1] = 100
402
403         if itera[2] < 100:
404             itera[2] = 100
405
406         if itera[0] < 100:
407             itera[0] = 100
408
409         if itera[1] < 100:
410             itera[1] = 100
411
412         if itera[2] < 100:
413             itera[2] = 100
414
415         if itera[0] < 100:
416             itera[0] = 100
417
418         if itera[1] < 100:
419             itera[1] = 100
420
421         if itera[2] < 100:
422             itera[2] = 100
423
424         if itera[0] < 100:
425             itera[0] = 100
426
427         if itera[1] < 100:
428             itera[1] = 100
429
430         if itera[2] < 100:
431             itera[2] = 100
432
433         if itera[0] < 100:
434             itera[0] = 100
435
436         if itera[1] < 100:
437             itera[1] = 100
438
439         if itera[2] < 100:
440             itera[2] = 100
441
442         if itera[0] < 100:
443             itera[0] = 100
444
445         if itera[1] < 100:
446             itera[1] = 100
447
448         if itera[2] < 100:
449             itera[2] = 100
450
451         if itera[0] < 100:
452             itera[0] = 100
453
454         if itera[1] < 100:
455             itera[1] = 100
456
457         if itera[2] < 100:
458             itera[2] = 100
459
460         if itera[0] < 100:
461             itera[0] = 100
462
463         if itera[1] < 100:
464             itera[1] = 100
465
466         if itera[2] < 100:
467             itera[2] = 100
468
469         if itera[0] < 100:
470             itera[0] = 100
471
472         if itera[1] < 100:
473             itera[1] = 100
474
475         if itera[2] < 100:
476             itera[2] = 100
477
478         if itera[0] < 100:
479             itera[0] = 100
480
481         if itera[1] < 100:
482             itera[1] = 100
483
484         if itera[2] < 100:
485             itera[2] = 100
486
487         if itera[0] < 100:
488             itera[0] = 100
489
490         if itera[1] < 100:
491             itera[1] = 100
492
493         if itera[2] < 100:
494             itera[2] = 100
495
496         if itera[0] < 100:
497             itera[0] = 100
498
499         if itera[1] < 100:
500             itera[1] = 100
501
502         if itera[2] < 100:
503             itera[2] = 100
504
505         if itera[0] < 100:
506             itera[0] = 100
507
508         if itera[1] < 100:
509             itera[1] = 100
510
511         if itera[2] < 100:
512             itera[2] = 100
513
514         if itera[0] < 100:
515             itera[0] = 100
516
517         if itera[1] < 100:
518             itera[1] = 100
519
520         if itera[2] < 100:
521             itera[2] = 100
522
523         if itera[0] < 100:
524             itera[0] = 100
525
526         if itera[1] < 100:
527             itera[1] = 100
528
529         if itera[2] < 100:
530             itera[2] = 100
531
532         if itera[0] < 100:
533             itera[0] = 100
534
535         if itera[1] < 100:
536             itera[1] = 100
537
538         if itera[2] < 100:
539             itera[2] = 100
540
541         if itera[0] < 100:
542             itera[0] = 100
543
544         if itera[1] < 100:
545             itera[1] = 100
546
547         if itera[2] < 100:
548             itera[2] = 100
549
550         if itera[0] < 100:
551             itera[0] = 100
552
553         if itera[1] < 100:
554             itera[1] = 100
555
556         if itera[2] < 100:
557             itera[2] = 100
558
559         if itera[0] < 100:
560             itera[0] = 100
561
562         if itera[1] < 100:
563             itera[1] = 100
564
565         if itera[2] < 100:
566             itera[2] = 100
567
568         if itera[0] < 100:
569             itera[0] = 100
570
571         if itera[1] < 100:
572             itera[1] = 100
573
574         if itera[2] < 100:
575             itera[2] = 100
576
577         if itera[0] < 100:
578             itera[0] = 100
579
580         if itera[1] < 100:
581             itera[1] = 100
582
583         if itera[2] < 100:
584             itera[2] = 100
585
586         if itera[0] < 100:
587             itera[0] = 100
588
589         if itera[1] < 100:
590             itera[1] = 100
591
592         if itera[2] < 100:
593             itera[2] = 100
594
595         if itera[0] < 100:
596             itera[0] = 100
597
598         if itera[1] < 100:
599             itera[1] = 100
600
601         if itera[2] < 100:
602             itera[2] = 100
603
604         if itera[0] < 100:
605             itera[0] = 100
606
607         if itera[1] < 100:
608             itera[1] = 100
609
610         if itera[2] < 100:
611             itera[2] = 100
612
613         if itera[0] < 100:
614             itera[0] = 100
615
616         if itera[1] < 100:
617             itera[1] = 100
618
619         if itera[2] < 100:
620             itera[2] = 100
621
622         if itera[0] < 100:
623             itera[0] = 100
624
625         if itera[1] < 100:
626             itera[1] = 100
627
628         if itera[2] < 100:
629             itera[2] = 100
630
631         if itera[0] < 100:
632             itera[0] = 100
633
634         if itera[1] < 100:
635             itera[1] = 100
636
637         if itera[2] < 100:
638             itera[2] = 100
639
640         if itera[0] < 100:
641             itera[0] = 100
642
643         if itera[1] < 100:
644             itera[1] = 100
645
646         if itera[2] < 100:
647             itera[2] = 100
648
649         if itera[0] < 100:
650             itera[0] = 100
651
652         if itera[1] < 100:
653             itera[1] = 100
654
655         if itera[2] < 100:
656             itera[2] = 100
657
658         if itera[0] < 100:
659             itera[0] = 100
660
661         if itera[1] < 100:
662             itera[1] = 100
663
664         if itera[2] < 100:
665             itera[2] = 100
666
667         if itera[0] < 100:
668             itera[0] = 100
669
670         if itera[1] < 100:
671             itera[1] = 100
672
673         if itera[2] < 100:
674             itera[2] = 100
675
676         if itera[0] < 100:
677             itera[0] = 100
678
679         if itera[1] < 100:
680             itera[1] = 100
681
682         if itera[2] < 100:
683             itera[2] = 100
684
685         if itera[0] < 100:
686             itera[0] = 100
687
688         if itera[1] < 100:
689             itera[1] = 100
690
691         if itera[2] < 100:
692             itera[2] = 100
693
694         if itera[0] < 100:
695             itera[0] = 100
696
697         if itera[1] < 100:
698             itera[1] = 100
699
700         if itera[2] < 100:
701             itera[2] = 100
702
703         if itera[0] < 100:
704             itera[0] = 100
705
706         if itera[1] < 100:
707             itera[1] = 100
708
709         if itera[2] < 100:
710             itera[2] = 100
711
712         if itera[0] < 100:
713             itera[0] = 100
714
715         if itera[1] < 100:
716             itera[1] = 100
717
718         if itera[2] < 100:
719             itera[2] = 100
720
721         if itera[0] < 100:
722             itera[0] = 100
723
724         if itera[1] < 100:
725             itera[1] = 100
726
727         if itera[2] < 100:
728             itera[2] = 100
729
730         if itera[0] < 100:
731             itera[0] = 100
732
733         if itera[1] < 100:
734             itera[1] = 100
735
736         if itera[2] < 100:
737             itera[2] = 100
738
739         if itera[0] < 100:
740             itera[0] = 100
741
742         if itera[1] < 100:
743             itera[1] = 100
744
745         if itera[2] < 100:
746             itera[2] = 100
747
748         if itera[0] < 100:
749             itera[0] = 100
750
751         if itera[1] < 100:
752             itera[1] = 100
753
754         if itera[2] < 100:
755             itera[2] = 100
756
757         if itera[0] < 100:
758             itera[0] = 100
759
760         if itera[1] < 100:
761             itera[1] = 100
762
763         if itera[2] < 100:
764             itera[2] = 100
765
766         if itera[0] < 100:
767             itera[0] = 100
768
769         if itera[1] < 100:
770             itera[1] = 100
771
772         if itera[2] < 100:
773             itera[2] = 100
774
775         if itera[0] < 100:
776             itera[0] = 100
777
778         if itera[1] < 100:
779             itera[1] = 100
780
781         if itera[2] < 100:
782             itera[2] = 100
783
784         if itera[0] < 100:
785             itera[0] = 100
786
787         if itera[1] < 100:
788             itera[1] = 100
789
790         if itera[2] < 100:
791             itera[2] = 100
792
793         if itera[0] < 100:
794             itera[0] = 100
795
796         if itera[1] < 100:
797             itera[1] = 100
798
799         if itera[2] < 100:
800             itera[2] = 100
801
802         if itera[0] < 100:
803             itera[0] = 100
804
805         if itera[1] < 100:
806             itera[1] = 100
807
808         if itera[2] < 100:
809             itera[2] = 100
810
811         if itera[0] < 100:
812             itera[0] = 100
813
814         if itera[1] < 100:
815             itera[1] = 100
816
817         if itera[2] < 100:
818             itera[2] = 100
819
820         if itera[0] < 100:
821             itera[0] = 100
822
823         if itera[1] < 100:
824             itera[1] = 100
825
826         if itera[2] < 100:
827             itera[2] = 100
828
829         if itera[0] < 100:
830             itera[0] = 100
831
832         if itera[1] < 100:
833             itera[1] = 100
834
835         if itera[2] < 100:
836             itera[2] = 100
837
838         if itera[0] < 100:
839             itera[0] = 100
840
841         if itera[1] < 100:
842             itera[1] = 100
843
844         if itera[2] < 100:
845             itera[2] = 100
846
847         if itera[0] < 100:
848             itera[0] = 100
849
850         if itera[1] < 100:
851             itera[1] = 100
852
853         if itera[2] < 100:
854             itera[2] = 100
855
856         if itera[0] < 100:
857             itera[0] = 100
858
859         if itera[1] < 100:
860             itera[1] = 100
861
862         if itera[2] < 100:
863             itera[2] = 100
864
865         if itera[0] < 100:
866             itera[0] = 100
867
868         if itera[1] < 100:
869             itera[1] = 100
870
871         if itera[2] < 100:
872             itera[2] = 100
873
874         if itera[0] < 100:
875             itera[0] = 100
876
877         if itera[1] < 100:
878             itera[1] = 100
879
880         if itera[2] < 100:
881             itera[2] = 100
882
883         if itera[0] < 100:
884             itera[0] = 100
885
886         if itera[1] < 100:
887             itera[1] = 100
888
889         if itera[2] < 100:
890             itera[2] = 100
891
892         if itera[0] < 100:
893             itera[0] = 100
894
895         if itera[1] < 100:
896             itera[1] = 100
897
898         if itera[2] < 100:
899             itera[2] = 100
900
901         if itera[0] < 100:
901             itera[0] = 100
902
903         if itera[1] < 100:
904             itera[1] = 100
905
906         if itera[2] < 100:
907             itera[2] = 100
908
909         if itera[0] < 100:
910             itera[0] = 100
911
912         if itera[1] < 100:
913             itera[1] = 100
914
915         if itera[2] < 100:
916             itera[2] = 100
917
918         if itera[0] < 100:
919             itera[0] = 100
920
921         if itera[1] < 100:
922             itera[1] = 100
923
924         if itera[2] < 100:
925             itera[2] = 100
926
927         if itera[0] < 100:
928             itera[0] = 100
929
930         if itera[1] < 100:
931             itera[1] = 100
932
933         if itera[2] < 100:
934             itera[2] = 100
935
936         if itera[0] < 100:
937             itera[0] = 100
938
939         if itera[1] < 100:
940             itera[1] = 100
941
942         if itera[2] < 100:
943             itera[2] = 100
944
945         if itera[0] < 100:
946             itera[0] = 100
947
948         if itera[1] < 100:
949             itera[1] = 100
950
951         if itera[2] < 100:
952             itera[2] = 100
953
954         if itera[0] < 100:
955             itera[0] = 100
956
957         if itera[1] < 100:
958             itera[1] = 100
959
960         if itera[2] < 100:
961             itera[2] = 100
962
963         if itera[0] < 100:
964             itera[0] = 100
965
966         if itera[1] < 100:
967             itera[1] = 100
968
969         if itera[2] < 100:
970             itera[2] = 100
971
972         if itera[0] < 100:
973             itera[0] = 100
974
975         if itera[1] < 100:
976             itera[1] = 100
977
978         if itera[2] < 100:
979             itera[2] = 100
980
981         if itera[0] < 100:
982             itera[0] = 100
983
984         if itera[1] < 100:
985             itera[1] = 100
986
987         if itera[2] < 100:
988             itera[2] = 100
989
990         if itera[0] < 100:
991             itera[0] = 100
992
993         if itera[1] < 100:
994             itera[1] = 100
995
996         if itera[2] < 100:
997             itera[2] = 100
998
999         if itera[0] < 100:
1000            itera[0] = 100
1001
1002        if itera[1] < 100:
1003            itera[1] = 100
1004
1005        if itera[2] < 100:
1006            itera[2] = 100
1007
1008        if itera[0] < 100:
1009            itera[0] = 100
1010
1011        if itera[1] < 100:
1012            itera[1] = 100
1013
1014        if itera[2] < 100:
1015            itera[2] = 100
1016
1017        if itera[0] < 100:
1018            itera[0] = 100
1019
1020        if itera[1] < 100:
1021            itera[1] = 100
1022
1023        if itera[2] < 100:
1024            itera[2] = 100
1025
1026        if itera[0] < 100:
1027            itera[0] = 100
1028
1029        if itera[1] < 100:
1030            itera[1] = 100
1031
1032        if itera[2] < 100:
1033            itera[2] = 100
1034
1035        if itera[0] < 100:
1036            itera[0] = 100
1037
1038        if itera[1] < 100:
1039            itera[1] = 100
1040
1041        if itera[2] < 100:
1042            itera[2] = 100
1043
1044        if itera[0] < 100:
1045            itera[0] = 100
1046
1047        if itera[1] < 100:
1048            itera[1] = 100
1049
1050        if itera[2] < 100:
1051            itera[2] = 100
1052
1053       
```

```

171         itera[2] = i
172         flag[2] = 1
173         z[j] = s[j], b[k], itera[0], itera[1], itera[2],
174         np.sqrt(np.trapz((old - new)**2, dx = dx)) / np.sqrt(np.
175         trapz(old**2, dx = dx)), Energy(A, dx)
176
177         np.savetxt(f, z)
178         f.write('\n')
179
180         f.close()
181
182 elif part == 4:
183
184     ######
185     # Part IV
186     #####
187
188     Compute features of linear model (E, P, sigma, C, phi)
189
190     ...
191
192     n = 50
193
194     z = np.zeros((n, 6))
195     s = np.linspace(0, 3, num = n)
196
197
198     for j in range(n):
199
200         A0 = 1 / np.cosh(2 * T) * np.exp(1j * np.pi / 4)
201
202         A0 = np.sqrt(E0 / Energy(A0, dx)) * A0 # Normalize
203
204         A = A0
205
206         for i in range(25):
207
208             old = np.angle(A[len(A)/2])

```

```
196         A = Loop(A, T, dx, s[j], 0, switch = False)
197         new = np.angle(A[len(A)/2])
198         while old > new:
199             old -= 2 * np.pi
200             r, sigma = curve_fit(func, T, np.abs(A))[0]
201             z[j] = s[j], Energy(A, dx), np.abs(A)[len(A)/2]**2,
202             sigma, -np.gradient(np.gradient(np.angle(A), dx), dx)[len(
203             A)/2] * sigma**2, new - old
204
205     np.savetxt('Linear.dat', z)
206
207 else:
208     print 'Please enter a valid part number (2--4)'
```