POLITECNICO DI TORINO

Master's Degree in Communication Engineering

Master's Degree Thesis

Modelling And Optimisation Of Ultra-Wide Band Long-Haul Raman-Amplified Coherent Optical Transmission Systems

Supervisors

Prof. Pierluigi POGGIOLINI

Dr. Yanchao JIANG

Candidate

Jad SARKIS

October 2024

Abstract

As data transmission demands grow, long-haul optical transmission links face increasing pressure. While bandwidth efficiency has been enhanced by using more complex modulation formats, further improvements are limited by signal-to-noise ratio (SNR) thresholds. Expanding usable bandwidth through Ultra-Wide Band (UWB) systems, utilizing the C, L, S and E bands to exploit up to 30 THz of bandwidth, has become the primary strategy for increasing transmission capacity.

However, UWB systems present challenges, such as the reliance on backward Raman amplification in the S band and the complications posed by inter-channel stimulated Raman scattering (ISRS), which causes uneven signal propagation across bands. To address these issues, accurate and efficient physical models are required for real-time optimization, which rely on the knowledge of the power of signals throughout the fibre span, the power profile.

This thesis develops a novel, more efficient method for computing the power profile of signals and pumps. Unlike traditional forward-backward approaches to solving the coupled Raman differential equations, the new method utilizes an integral form with matrix-based approximations, achieving up to a thirty-fold speed increase while maintaining an error margin under 0.05 dBm. These results represent a significant step forward towards reducing optimisation times, and enabling more extensive studies in ultra-wide band long haul optical transmission.

Acknowledgements

I would like to express my sincerest gratitude to everyone who contributed to the completion of this thesis.

First and foremost, I would like to thank my thesis supervisor, Professor Pierluigi Poggiolini. His availability for theoretical discussions, patience throughout this long process, and brilliant insights were crucial to the development of this work.

Secondly, I would like to extend my deepest thanks to Dr. Yanchao Jiang. Her support in understanding the existing software, her assistance in testing all the practical work I conducted, and her constant positivity helped me through the most challenging tasks—something I will always cherish. I also wish to acknowledge the support of the Department of Electronics and Telecommunications for taking care of the logistics that facilitated my work.

Finally, I would like to thank all my colleagues, friends, and family for their unparalleled support throughout this journey.

Table of Contents

List of Tables

List of Figures

Introduction

The exponential growth in data transmission demands is driven by different factors: the increasing numbers of users, the development of more demanding applications on the software level and the evolution of more complex network architectures allowing higher throughput to the end user. This growth has placed significant pressure on the long-haul optical transmission links to increase their provided throughput. To accomplish this increase, researchers are focusing on two possible approaches: Increasing the bandwidth efficiency by transmitting more bits per unit of frequency or increasing the exploitable bandwidth. The first method requires more complex modulation formats thus faces distance limitations in long-haul transmission. As a result, bandwidth expansion is becoming the primary solution to meet rising transmission needs.

The objective of the research of which this thesis is part of is to investigate Ultra-Wide Band (UWB) systems, which span across multiple frequency bands, including the C, L, S and E bands. These systems can provide up to 30 THz of exploitable bandwidth, significantly exceeding the 5 THz available in the C band alone, traditionally used in optical transmission. While UWB systems offer the potential for substantial throughput gains, they also introduce new challenges. One of these is Inter-channel Stimulated Raman Scattering (ISRS), a nonlinear effect that becomes significant between widely spaced channels, causing power transfer from higher-frequency channels to lower-frequency ones. Furthermore, as demonstrated in [1], backward Raman amplification becomes necessary to achieve higher throughput and maintain flat GSNR curves across the entire usable spectrum. Consequently, many input parameters such as individual channel and pump input power and pump frequency need to be optimised to mitigate the effects of ISRS and maximise the throughput on the link. For this reason, a physical model able to predict the achievable GSNR on the link according to the input parameters is a must for the optimal operation of such systems. The most prominent physical models already in use in the industry are based on the GN and EGN models developed in [2] and [3]. However, to enable more complex wide-band optimisations and real time operation, the requirement on these physical models regarding computation

speed has become increasingly more demanding. To address this, Closed Form Models (CFM) of the GN and EGN models have been developed by Politecinco di Torino in collaboration with CISCO in [4] and [5], the last of which has been extensively validated experimentally in [6].
However, one part of the CFM remains dependent on numerical calculations: The computation of the power profile. That is determining the power of each channel and pump at every point in the fibre, while taking into account ISRS and backward Raman amplification. While the CFMs allow for a substantial time gain over the integrated GN and EGN models, power profile calculations based on numerical solving of the coupled Raman differential equations now consist the bottleneck, consuming a staggering 95% of the computation time.

The main objective of this thesis is to develop a more efficient algorithm to compute the power profile, that is both fast and reliable compared to the conventional method. The successful development of such an algorithm will significantly reduce the computation time required to assess the achievable GSNR on any given link. This, in turn, will enable more efficient and large-scale optimization of UWB systems, facilitating future research of such systems. In practical terms, this advancement would allow network operators to manage and optimize multi-band systems in real-time, ensuring robust performance even under dynamic network conditions. By enhancing the speed of power profile calculations, this algorithm will lay the groundwork for more widespread adoption of UWB technologies, ultimately helping to meet the growing demand for high-throughput long-distance optical communication systems.

Chapter 1

The Closed Form Physical Model

In this chapter, the closed form physical model (CFM) of the Enhanced Gaussian Noise (EGN) model developed by the group at Politecnico di Torino in collaboration with CISCO will be examined. This model plays a key role in simulating and optimizing Ultra-Wideband (UWB) systems. The study will be done by first discussing the importance of physical models in optical networks, followed by an exploration of how UWB systems impact these models. Finally, the theory behind the current model will be described, highlighting its reliance on the power profile and discussing its limitations. A thorough understanding of this model is crucial for appreciating the potential benefit of a more efficient power profile calculator.

1.1 Physical-Layer Models

Physical-layer models (PLMs) are mathematical representations that rely on well-established equations and laws to describe and simulate various physical processes. In the case of optical networking, these models have been developed to analyze the transmission characteristics of light signals within the fiber. From signal degradation to the introduction of unwanted noise, a good physical model aims to account for all the physical effects that impact the signal, allowing for accurate predictions of the resulting signal-to-noise ratio at the end of the link. In the following, the important role of such models in the context of optical transmission will be described.

1.1.1 The Role of Physical Layer Models in Optical Transmission Systems

Physical layer modelling has become crucial for all aspects of optical communications. Having a tool able to predict signal degradation and the final SNR on a given link is invaluable for the design and optimisation of optical networks as it allows network engineers to simulate the performance of different system configurations before deployment. This enables the optimisation of various network parameters such as modulation formats, optical components and amplification scheme. An accurate physical model thus ensures that a given system is working in its optimal point (or local optimum at least), allowing for more efficient bandwidth use.
Additionally, PLMs are essential for capacity planning of a long-haul optical link. By simulating signal propagation over diverse fibre spans, the model can give insight into the quality of the signal, allowing operators to choose the most complex modulation format affordable, in such a way to maximise spectral efficiency while keeping the quality of transmission intact.

1.1.2 Challenges and Implications of Ultra-Wide band Systems

In the following, the role of PLMs in UWB systems is going to be discussed, for which a small definition of these systems is presented.

To better understand what is meant by ultra-wide-band systems, one needs to explore the different bands of frequency established in optical communications. The concept is illustrated by examining the curve representing the attenuation constant α as a function of frequency reported below:

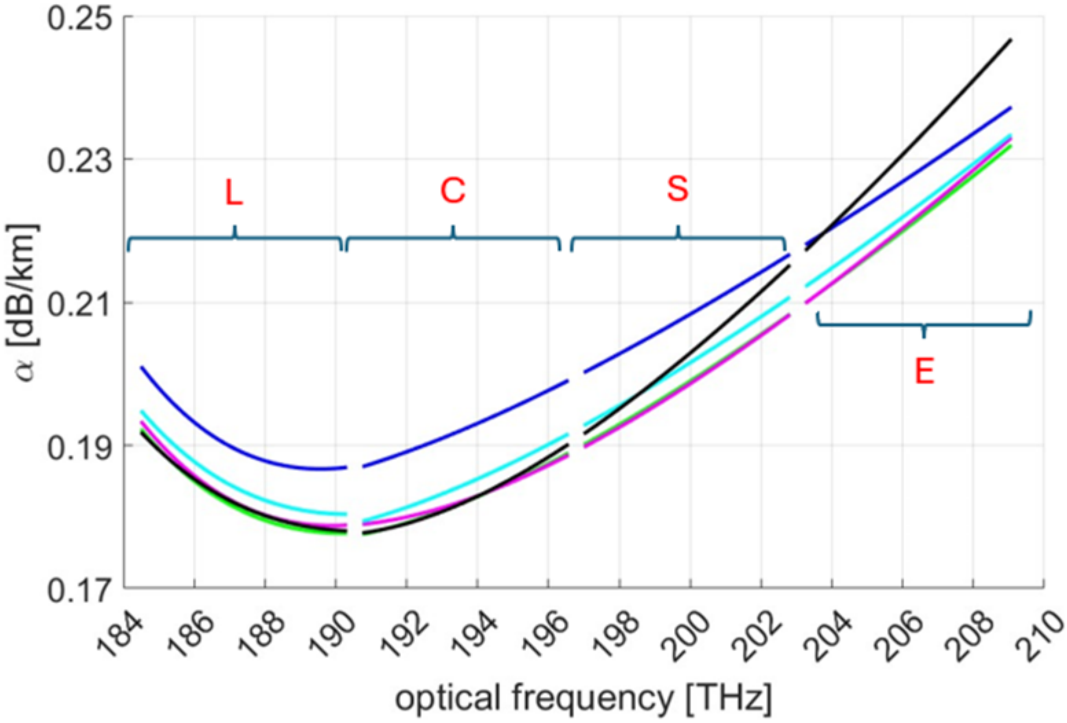

Figure 1.1: Signal power attenuation per kilometer as a function of frequency

α represents the loss in signal power per kilometer traveled inside the fibre. As shown in the figure, attenuation varies across different frequencies. In fact, some frequencies, with fairly similar values for α have been grouped in the so-called frequency bands labeled with the different letters shown in the figure. Particularly, the C band, also known as the conventional band, came to use due to 2 reasons: It has the lowest attenuation values and erbium-doped fibre amplifiers (EDFA) provide a uniform amplification over its whole band. Another set of amplifiers, happen to work in a uniform way on another set of frequencies that came to be known as the L band. We note then that the different frequency bands have been separated by some physical and technological boundaries that facilitated the use of some over others.
Recently, as mentioned in the introduction, research has focused on utilizing multiple frequency bands for optical transmission, significantly increasing the available bandwidth. By exploiting multiple bands simultaneously, ultra-wide-band systems can accommodate more channels in the same fiber, dramatically enhancing throughput.

This ultra-wide bandwidth has various impacts on the design of physical models primarily in two areas: First, certain fibre parameters that could once be approximated as constants with respect to frequency in single-band systems can no longer be considered as such without a significant loss of accuracy. These parameters include:

1. Attenuation coefficient α.
2. Dispersion coefficient β_2.
3. Fibre non linearity coefficient γ.
4. Amplifier noise figure F.

Therefore, physical models designed for UWB systems must account for this variation by using measured or estimated values of these parameters as they change with frequency.

The second area that physical models need to address is a nonlinear effect called Stimulated Raman Scattering (SRS) also known as the Raman effect.

As discussed in [7], when an incident light beam passes through a set of molecules in a material, a small part of the incident photons gets scattered around in different directions. Most of the scattered light is of unchanged wavelength, however a small percentage of the scattered photons end up loosing energy to the scattering molecules thus moving to a lower frequency. From an optical communications point of view, this effect can be seen as power transferred from a higher frequency light channel to other channels at lower frequencies. The amount of power transferred from a given channel i to another channel j at a given point z_0 in the fiber is expressed as follows:

$$\Delta P = g_{ij} \cdot P_i(z_0) * P_j(z_0) \tag{1.1}$$

where $P_i(z_0)$ and $P_j(z_0)$ are the powers present in channels i and j respectively at the point $z = z_0$ in the fibre, and g_{ij} is the Raman gain coefficient that depends on the frequency difference between the two channels.
The following figure shows an experimental curve representing the variation of the Raman gain coefficient with respect to the frequency for an amplifying channel at $f_c = 205$ THz.

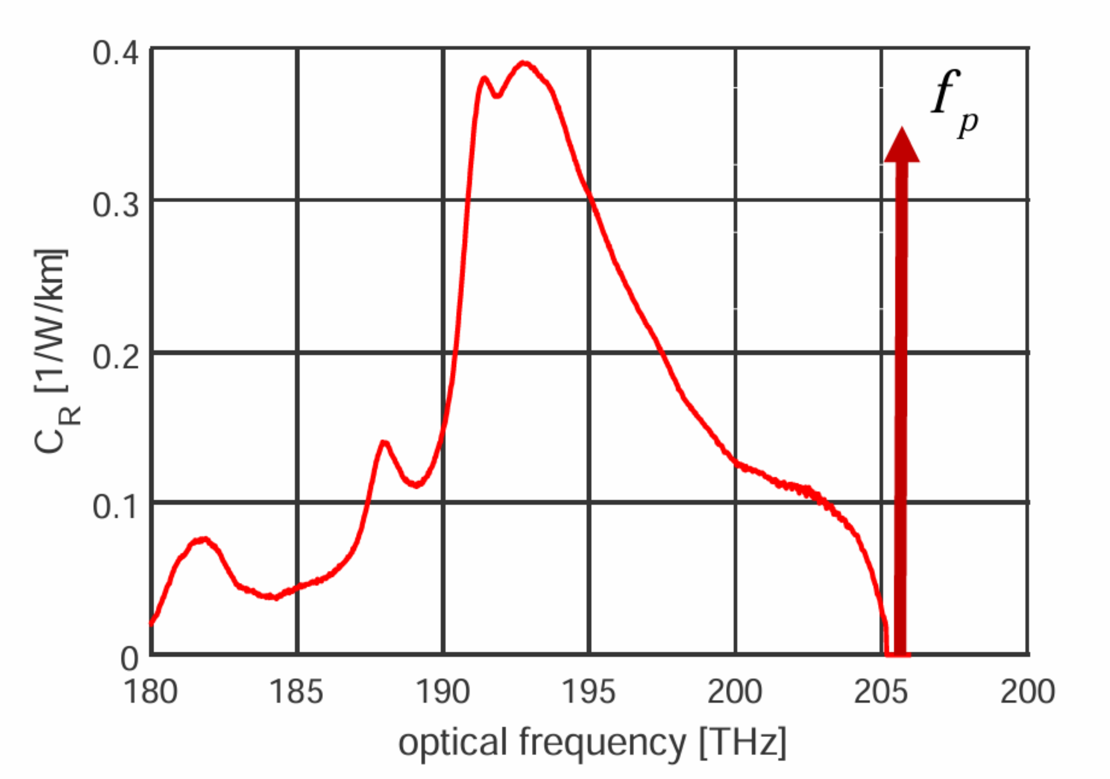

Figure 1.2: Raman gain coefficients for an amplifying channel at f_c=205 THz

By examining this curve, it can be confirmed that a given channel only amplifies signals with lower frequencies than itself. The figure further demonstrates that substantial power transfer is experienced within a frequency range of 5 to 15 THz away from the channel, peaking at around 13 THz. It should be noted that for amplifying channels at different frequencies than the one depicted in the above figure, the Raman gain coefficient curve remains the same up to a scaling factor as shown in figure 3 of [1].

For systems operating over a broad frequency range greater than 5 THz, such as in UWB systems, substantial stimulated Raman scattering can occur between channels, resulting in Inter-Channel Stimulated Raman Scattering (ISRS).

SRS has both drawbacks and advantages. On the one hand, ISRS can cause unwanted power transfer between channels, leading to signal degradation, especially in high frequency channels over wide bands. On the other hand, SRS can be leveraged beneficially through backward Raman amplification. This is achieved by using high-power pumps containing no useful information positioned at higher frequencies relative to the channels. These pumps boost signal power and improve performance across the entire band through SRS, effectively providing continuous amplification around the end of a span.

Consequently, a physical model designed to work on UWB systems has to be able to account for SRS.

In conclusion, PLMs are essential for the optimal design and operation of any optical transmission system. Their importance becomes even more pronounced when working with UWB systems, which require accounting for frequency-dependent parameters and handling stimulated Raman scattering (SRS) between channels and between channels and Raman pumps.
Since ISRS depletes higher-frequency channels, input power compensation for these channels should be considered, by giving them additional power in input with respect to the lower-frequency ones. Through optimization, PLMs can help determine the optimal input power distribution that maximizes a key performance metric, such as link throughput. For real-time operation, it is crucial that these models complete their computations within a reasonable time frame.
This section has thus highlighted the significance of PLMs in optical communication, particularly in UWB systems, and stressed the importance of making them more computationally efficient.

1.2 Model Description

In this section, the closed form Physical Layer model developed by the group at Politecnico di Torino will be thoroughly described. This model builds upon the Enhanced Gaussian Noise (EGN) model developed in [3], a widely used method for estimating non-linear interference in optical systems. The EGN model relies on multi-dimensional numerically evaluated integrals, which can be computationally intensive. To address this, an approximate closed-form model (CFM) was introduced in [4] to significantly reduce computation time. This section will first present the CFM equations, followed by a discussion of its critical dependence on accurately modeling the channel power profile. Finally, the practical implementation of the CFM and its limitations will be explored.

1.2.1 The closed form GN model

It is important to first start with a brief distinction between the GN and EGN models. The Gaussian Noise (GN) model provides a simplified method for estimating non-linear interference (NLI) in optical fibre communication systems by treating the NLI as Gaussian-distributed noise and estimating its power spectral density (PSD). It assumes certain conditions, such as Gaussian-like signals due to high chromatic dispersion, and relatively low NLI power allowing the interference to be considered negligible compared to the signal. This makes it less accurate in the first couple of spans when the signal does not have a fully Gaussian behaviour yet as well as in low dispersion fibres and low baud rates. The detailed analytical derivation of the GN model can be seen here [2].

The Enhanced Gaussian Noise (EGN) model, on the other hand, extends the GN model by incorporating more detailed interactions between the signal and non-linear effects, notably by considering the effect of different modulation formats and completely removing the signal Gaussianity approximation. It also extends the simple second order fibre dispersion taken by the GN model to the third order making it more accurate on zero dispersion fibres. This allowed to mitigate the GN model's overestimation of NLI noise in the cases mentioned above. However, the EGN model includes multi-dimensional integrals (triple and quadruple) to account for these broader interactions, making it more accurate but significantly more computationally intensive compared to the GN model.

The closed-form EGN model was developed to strike a balance between accuracy and computational efficiency. It is built by approximating the integral equations in the original GN model and adding finely tuned machine learning correction factors to make it better approximate the EGN model.

The closed form equations (1.2) to (1.6) derived in [8] that approximate the double integrals present in the GN model are reported below:

$$G_{\text{NLI}}^{Rx}(f_{\text{CUT}}) = \sum_{n=1}^{N_{\text{span}}} \left(G_{\text{NLI}}^{(n)}(f_{\text{CUT}}) \prod_{k=n+1}^{N_{\text{span}}} \Gamma^{(k)}(f_{\text{CUT}}) \cdot e^{-2\alpha^{(k)}(f_{\text{CUT}}) L_{\text{span}}^{(k)}} \right) \tag{1.2}$$

$$\begin{aligned} G_{\text{NLI}}^{(n)}(f_{\text{CUT}}) = \frac{16}{27} (\gamma^{(n)})^2 \Gamma^{(n)}(f_{\text{CUT}}) \cdot e^{-2\alpha^{(n)}(f_{\text{CUT}}) L_{\text{span}}^{(n)}} \cdot \bar{G}_{\text{CUT}}^{(n)} \cdot \\ \left(\rho_{\text{CUT}}^{(n)} \cdot (\bar{G}_{\text{CUT}}^{(n)})^2 I_{\text{CUT}}^{(n)} + \sum_{n_{\text{ch}}=1, n_{\text{ch}} \neq n_{CUT}^{(n)}}^{N_{\text{ch}}^{(n)}} \left(2\rho_{n_{\text{ch}}}^{(n)} \cdot (\bar{G}_{n_{\text{ch}}}^{(n)})^2 I_{n_{\text{ch}}}^{(n)} \right) \right) \end{aligned} \tag{1.3}$$

$$\begin{aligned} I_{\text{CUT}}^{(n)} = \frac{1}{2\pi |\bar{\beta}_{2,\text{CUT}}^{(n)}| 2\alpha^{(n)}(f_{\text{CUT}})} \cdot \left(\text{asinh} \left(\frac{\pi^2}{2} \left| \frac{\bar{\beta}_{2,\text{CUT}}^{(n)}}{2\alpha^{(n)}(f_{\text{CUT}})} \right| R_{\text{CUT}}^2 \right) \right. \\ \left. + 2 \frac{S_i \left(\pi^2 |\bar{\beta}_{2,\text{CUT}}^{(n)}| L_{\text{span}}^{(n)} B_{\text{CUT}}^2 \right)}{\pi \alpha^{(n)}(f_{\text{CUT}}) L_{\text{span}}^{(n)}} \cdot \left(\text{HN}(N_{\text{span}} - 1) + \frac{1 - N_{\text{span}}}{N_{\text{span}}} \right) \right) \end{aligned} \tag{1.4}$$

$$I_{n_{ch}}^{(n)} = \frac{\text{asinh} \left(\frac{\pi^2}{2} \left| \frac{\bar{\beta}_{2,n_{ch}}^{(n)}}{2\alpha^{(n)}(f_{n_{ch}}^{(n)})} \right| (f_{n_{ch}}^{(n)} - f_{\text{CUT}} - \frac{R_{n_{ch}}^{(n)}}{2}) \cdot R_{\text{CUT}} \right) - \text{asinh} \left(\frac{\pi^2}{2} \left| \frac{\bar{\beta}_{2,n_{ch}}^{(n)}}{2\alpha^{(n)}(f_{n_{ch}}^{(n)})} \right| (f_{n_{ch}}^{(n)} - f_{\text{CUT}} - \frac{R_{n_{ch}}^{(n)}}{2}) \cdot R_{\text{CUT}} \right)}{4\pi \left| \bar{\beta}_{2,n_{ch}}^{(n)} \right| \cdot 2\alpha^{(n)}(f_{n_{ch}}^{(n)})} \tag{1.5}$$

$$\bar{\beta}^{(n)}_{2,\mathrm{CUT}} = \beta^{(n)}_2 + \pi\beta^{(n)}_3(2f_{\mathrm{CUT}} - 2f^{(n)}_c), \quad \bar{\beta}^{(n)}_{2,n_{ch}} = \beta^{(n)}_2 + \pi\beta^{(n)}_3 \cdot (f^{(n)}_{n_{ch}} + f_{\mathrm{CUT}} - 2f^{(n)}_c) \tag{1.6}$$

As a general notation, the superscripts (n) and (k) refer to the nth and kth spans respectively, while the subscript n_{ch} is an index that refers to any channel in the WDM spectrum and the subscript CUT refers to the channel under test, which is the one for which the NLI is being estimated.
The quantity $G^{Rx}_{\mathrm{NLI}}(f_{\mathrm{CUT}})$ in equation 1.2 refers to the PSD of the accumulated NLI at the receiver and at the frequency of the CUT, this is ultimately the quantity that the CFM is estimating since it allows computing the final GSNR of the CUT through the following formula:

$$GSNR = \frac{P_{CUT}}{G^{Rx}_{NLI}(f_{CUT}) \cdot R_{CUT} + P_{ASE}} \tag{1.7}$$

The quantities $G^{(n)}_{\mathrm{NLI}}(f_{\mathrm{CUT}})$ present in equation 1.2, that can be calculated using equation 1.3, represent the PSD of NLI produced on the n^{th} span alone on the frequency of the CUT.
$I^{(n)}_{\mathrm{CUT}}$ and $I^{(n)}_{n_{\mathrm{ch}}}$ represent the estimations of Self-Channel Interference (SCI) caused by the CUT on itself and Cross-Channel Interference (XCI) caused by all other WDM channels on the CUT in a given span n, respectively.

In equation 1.4, HN stands for the harmonic number and S_i is the sine-integral function.

Finally, the terms ρ^n_{CUT} and $\rho^n_{n_{ch}}$ present in equation 1.3 are the machine learning factors added to allow the CFM to approximate the EGN model. In older versions of the CFM approximating the GN model only, these factors were set to 1. Their equations are as follows:

$$\begin{aligned} \rho^{(n)}_{\mathrm{nch}} &= (1 + a_{19} \cdot r^{a_{20}}_{\mathrm{CUT}} + a_{21} \cdot (r^{(n)}_{n_{ch}})^{a_{22}}) \cdot \{a_1 + a_2 \cdot (\Phi^{(n)}_{n_{ch}})^{a_3} + \\ &\qquad a_4 \cdot (\Phi^{(n)}_{n_{ch}})^{a_5} \cdot (1 + a_6 \cdot [|\beta_{2,acc}(n.n_{ch})| + a_7]^{a_8})\} \\ \rho_{\mathrm{CUT}} &= (1 + a_{23} \cdot r^{a_{24}}_{\mathrm{CUT}}) \cdot \{a_9 + a_{10} \cdot (\Phi^{a_{11}}_{\mathrm{CUT}} + a_{12} \cdot (\Phi_{\mathrm{CUT}})^{a_{13}} \cdot \\ &\qquad (1 + a_{14} \cdot R^{a_{15}}_{\mathrm{CUT}} + a_{16} \cdot [|\beta_{2,acc}(n.n_{\mathrm{CUT}})| + a_{17}]^{a_{18}})\} \end{aligned} \tag{1.8}$$

Where R_{CUT} represents the symbol rate of the CUT, r_{CUT} and $r_{n_{ch}}$ are the roll-off factors of the CUT and the current WDM channel n_{ch}, finally, $\Phi_{n_{ch}}$ is a constant that depends on the modulation format of the specific channel. a_1-a_{24} are parameters that were found through a machine learning optimisation approach.

The process of finding the machine learning factors is described broadly in the following:

The optimisation aims to minimise the error between the PSD estimated by the CFM and the one estimated by the fully integrated EGN model:

$$\Delta = \frac{|P^{(n)}_{NLI,CFM} - P^{(n)}_{NLI,EGN}|^2}{|P^{(n)}_{NLI,EGN}|^2}$$

In the purpose of this optimisation, 1500 C-band optical transmission systems, equally divided between fully and partially loaded randomly generated WDM spectra and modulation formats were used as a training set. It is noted that the cost function was evaluated at each span of these systems, providing more than 11500 error contributions. This process provided specific values for these parameters, which achieve a peak error of 0.19 dB. The parameters are reported in the table below.

Parameter	**Value**	**Parameter**	**Value**
a_1	$+1.0436e0$	a_{13}	$+1.0229e0$
a_2	$-1.1878e0$	a_{14}	$-1.1440e0$
a_3	$+1.0573e0$	a_{15}	$+1.1393e-2$
a_4	$-1.8309e+1$	a_{16}	$+3.8070e+5$
a_5	$+1.6665e0$	a_{17}	$+1.4785e+3$
a_6	$-1.0020e0$	a_{18}	$-2.2593e0$
a_7	$+9.0933e0$	a_{19}	$-6.7997e-1$
a_8	$+6.6420e-3$	a_{20}	$+2.0215e0$
a_9	$+8.4481e-1$	a_{21}	$-2.9781e-1$
a_{10}	$-1.8530e0$	a_{22}	$+5.5130e-1$
a_{11}	$+9.4539e-1$	a_{23}	$-3.6718e-1$
a_{12}	$-1.5421e+1$	a_{24}	$+1.1486e0$

Table 1.1: Optimised values of the parameters a_1-a_{24} for the machine learning factors in the CFM equations

In conclusion, the equations 1.2 - 1.8 along with the parameters in table 1.1 fully describe the EGN CFM used in the software, able to provide an estimation for NLI power for a given set of input parameters.

1.2.2 Dependence on the power profile: A key Consideration

In the following, the equations of the CFM are analyzed to explore their dependence on the power profile. This dependence is crucial because the calculation of the power profile, especially in backward Raman amplified systems, cannot be done in closed form, posing a significant computational bottleneck. A major objective of this thesis, discussed in the following section, is the development of a novel algorithm for power profile computation that can potentially make the entire model faster and more efficient.

In particular, the channel and pump input powers are reflected in key terms such as $\bar{G}^{(n)}_{\text{CUT}}$ and $\bar{G}^{(n)}_{n_{\text{ch}}}$, which represent the input PSDs at the start of the n^{th} span for the channel under test (CUT) and a WDM channel, respectively. In fact these terms can be related to the channel input power as follows:

$$\bar{G}^{(n)}_{\text{CUT}} = \frac{P^{(n)}_{\text{CUT}}}{R_{\text{CUT}}}$$

$$\bar{G}^{(n)}_{n_{ch}} = \frac{P^{(n)}_{nch}}{R_{n_{ch}}}$$

Moreover, to account for SRS, the model uses an edited version of the attenuation constant that contains a z-independent component and a z-dependent one. Its expression thus becomes:

$$\alpha(f,z) = \alpha_0(f) + \alpha_1(f) \cdot e^{-\sigma(f)\cdot Z} \tag{1.9}$$

This expression, with the correct values for $\alpha_0(f_{ch}), \alpha_1(f_{ch})$and $\sigma(f_{ch})$ is able to fit the power variation of each channel along the fibre, subject to Raman gain and loss. While the factor α_0 accounts for fibre attenuation, the inclusion of the z-dependent exponential term allows to capture the equivalent gain or loss experienced by the channel due to the Raman effects.
The values of these parameters are effectively obtained by best-fitting the curve $P \cdot e^{-\alpha(f_{n_{ch}},z)\cdot z}$ to the power profile curve. This means that the accurate description of each channel power along the fibre, namely the power profile, needs to be known for determining $\alpha(f,z)$ and for the model to function.

1.2.3 Implementation and limitations

The equations presented thus far are all implemented in MATLAB to form the CFM able to take in input the link parameters, input channel and pump powers and frequencies, and accurately estimate the available GSNR on the link for each channel while taking care for stimulated raman scattering. All of which is done in closed form aside from the part of the power profile calculation. In fact, the evaluation of the power profile is governed by the system of coupled non-linear Raman differential equations that read:

$$\begin{cases} \pm\dfrac{dP_i(z)}{dz} = -\alpha_{0,i}P_i(z) + \sum\limits_{j=1}^{M+N} g_{ij}P_j(z)P_i(z) \\ P_i(0) = P_{n_{ch},in} \quad \text{for } 1 \leq i \leq M \\ P_i(L_s) = P_{n_{pump},in} \quad \text{for } M+1 \leq i \leq N+M \end{cases} \tag{1.10}$$

Where M is the number of channels in the system and N is the number of pumps. The parameter $\alpha_{0,i}$ represents the attenuation coefficient for channel i.

In the original software, this system is solved using MATLAB's functions 'ode45' and 'bvp4c', which are capable of solving initial value problems and boundary value problems, respectively. It is important to note that if backward Raman amplification is employed, then $N \neq 0$, and the system becomes a boundary value problem.

In this latter case, the numerical algorithm utilized by MATLAB in the function 'bvp4c' is a finite difference code that implements the three-stage Lobatto IIIa formula. This algorithm can be highly inefficient, especially with an increasing number of channels, which is often the case in UWB systems.
From here stems a limitation of the current CFM software as it takes a substantial amount of time to compute the power profile with respect to all the other computations including the estimations of the PSDs of NLI and ASE. Two examples of the time taken by these calculations are shown to illustrate this fact:

- A C+L system with 76 channels and 5 pumps is considered, the channels have a uniform input power of 0 dBm while the pumps have the following frequencies and respective power:
  Frequencies in THz :[210.56, 208.87, 206.72, 204.51, 200.55]
  Power in mW :[360, 320, 200, 130, 180]
  The following table shows a profiling of the times taken by the different processes in the software averaged over 20 runs:

Process	**Time Elapsed (s)**
Power Profile	8.925
Curve Fitting	0.3935
NLI Estimation	0.1767
Total	9.496
$\frac{\text{Power Profile}}{\text{Total}}$	0.939

Table 1.2: Time elapsed for each process in the CFM model for a C+L system

- A C+L+S system with 114 channels and 5 pumps is considered, with the same power and frequency except the last pump is shifted by 2 THz to avoid channel overlap, the pump frequencies become in THz: [210.56, 208.87, 206.72, 204.51, 202.55] .

The following table shows a profiling of the times taken by the different processes in the software averaged over 20 runs:

Process	**Time Elapsed (s)**
Power Profile	30.043
Curve Fitting	0.662
NLI Estimation	0.999
Total	31.042
$\frac{\text{Power Profile}}{\text{Total}}$	0.968

Table 1.3: Time elapsed for each process in the CFM model for a C+L+S system

As shown in the tables, the power profile calculation accounts for approximately 95% of the total simulation time. This proportion increases as the system size grows, indicating further complications when simulating larger systems, such as those incorporating the S and E bands.

While tens of seconds may not seem like a significant amount of time on their own, as outlined in Section 1.1, UWB systems rely on the PLM for tasks like optimizing channel and pump input power to maximize overall link throughput. Such optimization processes require running the program tens of thousands of times to converge to a local minimum, potentially taking several days or even weeks. Therefore, reducing the complexity of the power profile calculations by even a modest factor can lead to substantial time saving in these optimization processes.

In conclusion, this chapter has effectively described the importance of PLMs in optical communication. It has described the current CFM EGN model used for NLI estimation and explored its crucial dependence on the knowledge of the power profile. Finally, it has shown the inefficiency of the power profile calculations done by exploiting MATLAB's functions and the need of a novel algorithm able to evaluate the power profile in a more efficient way. In the next chapter, a take on the novel algorithm developed by this thesis is presented in detail.

Chapter 2

The New Algorithm

In this chapter, the new algorithm for the calculation of the power profile developed in MATLAB during the course of this thesis will be presented and detailed. Understanding the underlying concepts and algorithms of this method is essential for anyone looking to further develop or generalize it.

The chapter will go over the key concept of the algorithm, the two methods employed to solve the problem and their limitations. Finally the chapter will describe the merging of the two methods into the final robust and reliable algorithm, and the automation of the various parameters enabling it to adapt to various system scenarios.

2.1 Key Concept

The algorithm to be developed is one that presents a more efficient way to compute the power profile of a given UWB system in which ISRS is significant and pumps are injected from the other side of the fibre for amplification over the whole useful band. This is a boundary value problem where the input powers of the signal channels at the beginning of the fibre, noted $P_{n_{ch},in}$, are known and the input powers of the amplifying pumps at the end of the fibre, noted $P_{n_{pump},in}$, are also known, where n_{ch} and n_{pump} are the respective indexes of channels and pumps. The power variation across the fibre of each channel is affected by attenuation, Raman gain from the pumps and Raman gain/loss between itself and other channels at higher/lower frequencies. This interaction can be summarised by the Raman coupled differential equations. The system to be resolved is thus the one represented by equation 1.10 of section 1.
The new approach of this algorithm is based on the suggestion by [9] of moving to the integral form of the equations and estimating the resulting integrals using the trapezoidal rule. This section will explore this passage and will provide an analysis

of the resulting form of equations.

2.1.1 Differential to integral form

The set of coupled Raman differential equations introduced in section 1.2.3 are written again in the following:

$$\pm\frac{dP_i(z)}{dz} = -\alpha_{0,i}P_i(z) + \sum_{j=1}^{M+N} g_{ij}P_j(z)P_i(z) \tag{2.1}$$

These equations state that the power variation of a channel i in an increment of the fibre length dz, is determined by the proportion of power lost to attenuation and that of power lost or gained due to ISRS or Raman pumps amplification. The upper and lower sign indicate a co-propagating wave (signals or co-propagating pumps) and a counter-propagating wave (backward pumps) respectively.

The main problem in numerically resolving this system of differential equations lies in the fact that the system is a boundary value problem where boundary conditions are imposed at both ends of the fibre. This requires multiple iterations along the whole fibre length to re-conciliate both boundary conditions. Moreover, due to the physical properties of the system, the variation of channel power can change drastically in different regions of the fibre. For example, with backward Raman amplified systems, the channels typically see no Raman amplification in the begging of the fibre where pump power is low but then end up being amplified much more towards the last 10 Km of fibre where pump power is at its highest. This requires the solver to use variable step sizes for different regions of the fibre which might render it slower.

Instead, another approach to the resolution of this system was proposed by [9], by moving to the integral form of the equations, seeking solutions along the iteration axis rather than the fibre propagation axis.

By dividing both parts of the equation 2.1 by P_i and integrating, the following integral form results:

$$P_i(z) = P_i(0) \cdot \exp\left(\mp\alpha_{0,i}z \pm \sum_{j=1}^{M+N} g_{ij}\int_0^z P_j(\psi)\,d\psi\right) \tag{2.2}$$

In this form, the power profile of channel i at any point z of the fibre, $P_i(z)$ is explicitly expressed as a function of z. The difference here is that the coupled system of differential equations are no longer solved directly, which involved multiple variables affecting each other's rate of change. Rather the solution is obtained by iterating through a formula of $P_i(z)$ where each value is updated based on the values obtained from previous steps. This means that an initial guess for the power profiles should be taken, feeding this guess to equation 2.2, an updated power profile can be obtained. By doing small changes on the obtained profile and re-feeding it to equation 2.2, a slightly refined power profile can be obtained. This goes on until the power profile in output from the equations is deemed satisfactory. More information on this will be given in the next subsection.

2.1.2 Equation resolution and analysis

To use equation 2.2 for generating a power profile, a vectorized approach is used. MATLAB is optimised for vector operations, this is in fact a big factor as to why this method is faster and more efficient than solving the differential equations directly.

A vector Z is created to represent all the steps inside the fiber, with a given step size ΔZ. This vector is expressed as:

$$Z = [0, \Delta Z, 2\Delta Z, \ldots, L - \Delta Z, L]$$

where L is the total length of the fiber. The number of steps is defined as $N_z = \lceil \frac{L}{\Delta Z} \rceil$ and the total number of channels including signals and pumps is defined as $N_{ch} = M + N$.

Another vector P_j of length N_z is created. It represents the powers of channel j at all the steps in Z. The $N_{ch} \times N_z$ matrix P is then formed by overlaying the vectors P_i in a vertical fashion, forming the power profile matrix:

$$P = \begin{pmatrix} P_1(0) & P_1(\Delta z) & \cdots & P_1(L) \\ P_2(0) & P_2(\Delta z) & \cdots & P_2(L) \\ \vdots & \vdots & \ddots & \vdots \\ P_{N_{\text{ch}}}(0) & P_{N_{\text{ch}}}(\Delta z) & \cdots & P_{N_{\text{ch}}}(L) \end{pmatrix}$$

Furthermore, the integral $\int_0^z P_j(\psi) d\psi$ present in the equation must be numerically solved. This is accomplished using the trapezoidal rule, which approximates the integral by dividing the area under the curve into trapezoids.

In this method:

- The width of each trapezoid is ΔZ, the step size.
- The two heights of the trapezoid are the values of the function $P_j(\psi)$ at the two points defining the base, i.e., the values at Z_i and Z_{i+1}.

Mathematically, this approximation can be written for an interval $[Z_i, Z_{i+1}]$ as:

$$\int_{Z_i}^{Z_{i+1}} P_j(\psi) d\psi \approx \Delta Z \frac{(P_j(Z_i) + P_j(Z_{i+1}))}{2}$$

In equation 2.2, the integral always has the bounds [0 z], with z being the point in the fibre on which the equation is estimating the power value $P_i(z)$. The approximation using the trapezoidal rule will look like this:

$$\begin{aligned}\int_0^z P_j(\psi)d\psi \approx \Delta Z\frac{(P_j(0)+P_j(\Delta Z))}{2} &+ \Delta Z\frac{(P_j(\Delta Z)+P_j(2\Delta Z))}{2}\\ &+\cdots+\Delta Z\frac{(P_j(z-\Delta Z)+P_j(z))}{2}\end{aligned}$$

By grouping the different powers figuring in the expression the following is obtained:

$$\int_0^z P_j(\psi)d\psi \approx \Delta Z\left(\frac{1}{2}P_j(0)+P_j(\Delta Z)+P_j(2\Delta Z)+...+P_j(z-\Delta Z)+\frac{1}{2}P_j(z)\right)$$

This can be written as the scalar product of the vector P_j with the vector $T = [1/2, 1, ..., 1, 1/2,0\dots0]$ with length equal to the number of steps between the beginning of the fibre and z then padded with zeros to reach the length of P_j, N_z:

$$\int_0^z P_j(\psi)d\psi \approx \Delta Z \cdot P_j \cdot T$$

To compute the integral for each possible value of z in the vector Z, an $N_z \times N_z$ matrix, T_{trig}, is constructed. Each column of this matrix represents a version of the vector T, corresponding to the specific value of z for that column, which means it contains the value 1/2 at its beginning and end, with $\frac{z}{\Delta Z} - 1$ ones in between. The column is then padded with zeros to match the length N_z. The matrix takes the following form:

$$T_{trig} = \begin{bmatrix} 0 & 1/2 & 1/2 & 1/2 & \dots & 1/2 \\ 0 & 1/2 & 1 & 1 & \dots & 1 \\ 0 & 0 & 1/2 & 1 & \dots & 1 \\ 0 & 0 & 0 & 1/2 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1/2 \end{bmatrix}$$

It is noted that the first column is all zeros since it corresponds to the integral evaluated at the bounds [0;0] which should equal 0. The value of the integral at every step of the vector Z is then obtained by the matrix product

$$P_j \times T \cdot \Delta Z \tag{2.3}$$

To finalize the vector equation, the vector α_0 containing all the attenuation coefficients for the channels and pumps in order and the $N_{ch} \times N_{ch}$ matrix G

containing all the Raman gain coefficients g_{ij} are constructed.
The vector equation that computes the power profile vector P_i, containing the powers of a given channel i sampled at equal steps ΔZ of the fibre thus becomes:

$$P_i = P_i(0) \cdot \exp\left(\mp\alpha_0(i)Z \pm G(i,:) \times P \times T_{trig} \cdot \Delta Z\right)$$

The above equation can be extended to compute the whole power profile P in one shot. For this purpose, an additional vertical vector of length N_{ch} called Direction is constructed by putting 1 in the position relative to a signal or a co-propagating pump, and -1 in the position relative to a pump indicating counter-propagation. Additionally, the vector Z is replicated in a vertical fashion N_{ch} times to form the $N_{ch} \times N_z$ matrix Zk. The final vector equation, computing the whole power profile P sampled at equal steps of the fibre, has the following form:

$$P = P(:,0) \cdot \exp\left(-Direction \cdot \alpha_0 \cdot Zk + Direction \cdot G \times P \times T_{trig} \cdot \Delta Z\right) \quad (2.4)$$

Equation 2.4 serves as the foundation for the efficient power profile computation method presented in this thesis. Starting with an initial guess of the power profile P and applying the boundary conditions, this equation allows for the propagation of the profile according to the Raman differential equations, providing a numerical solution for the system in vector form. However, due to the numerical nature of the method, the accuracy of the output depends heavily on how close the initial guess is to the actual solution.

To determine when an algorithm has converged, it is essential to define what constitutes a "satisfactory solution." A correct solution must first remain stable when propagated through equation 2.4. Additionally, if the profile obtained after propagation matches the boundary conditions within a certain error tolerance, it is considered valid.

Since the initial guess is rarely accurate enough to produce a satisfactory solution on the first try, an iterative approach is necessary. With each step, the obtained profile is adjusted slightly for the next iteration, and the boundary conditions are monitored to ensure convergence towards the correct solution. Three methods that implement this process in different ways, namely dynamic pump allocation method, progressive signal injection method and the hybrid method will be described in the following sections.

2.2 Dynamic Pump Calibration Method

As specified in the previous section, there is the need for algorithms that exploit equation 2.4 to achieve a converged solution. They should take an initial guess of the power profile, along with the boundary conditions and get closer to the correct power profile with every iteration if possible. The Dynamic Pump Calibration (DPC) Method is one of these algorithms.

2.2.1 Loop description

To understand how this method works, the first couple of iterations will be discussed in detail. This will help motivate the choices made in the method logic.

First, an initial guess of the power profile should be provided. A simple guess is one where all channels, signals and pumps, propagate from the respective ends of the fibre subject only to fibre attenuation, with no SRS taken into account. This guess can be expressed as:

$$P_i(z) = P_i(0) \cdot \exp\left(-\text{Direction}(i) \cdot \alpha_0(i) z\right)$$

Furthermore, a C+L system will be adopted as the case example for the whole method development. This system has 38 channels in the C band and 38 channels in the L band, with frequencies spanning in the range [186;195] THz. The channel spacing is 125 GHz and a symbol rate of 100 Gbaud is used.

For the purpose of amplification 5 counter-propagating pump channels are used in the frequencies : [210.56, 208.87, 206.72, 204.51, 200.55] THz. The power of each signal and pump channel at the respective ends of the fibre, forming the boundary conditions, are variable during the different tests to ensure the generalization of the method for different power scenarios.
One specific case is taken with uniform signal power per channel:$P_{n_{signal},in} = -10$ dBm $\forall n_{signal} \in [1; M]$ and pump powers [360 320 200 130 180] mWatts equivalent to [25.56 25.05 23.01 21.13 22.55] dBm respectively.
The initial guess of this case is represented below in logarithmic scale, as will always be the case when representing power profiles:

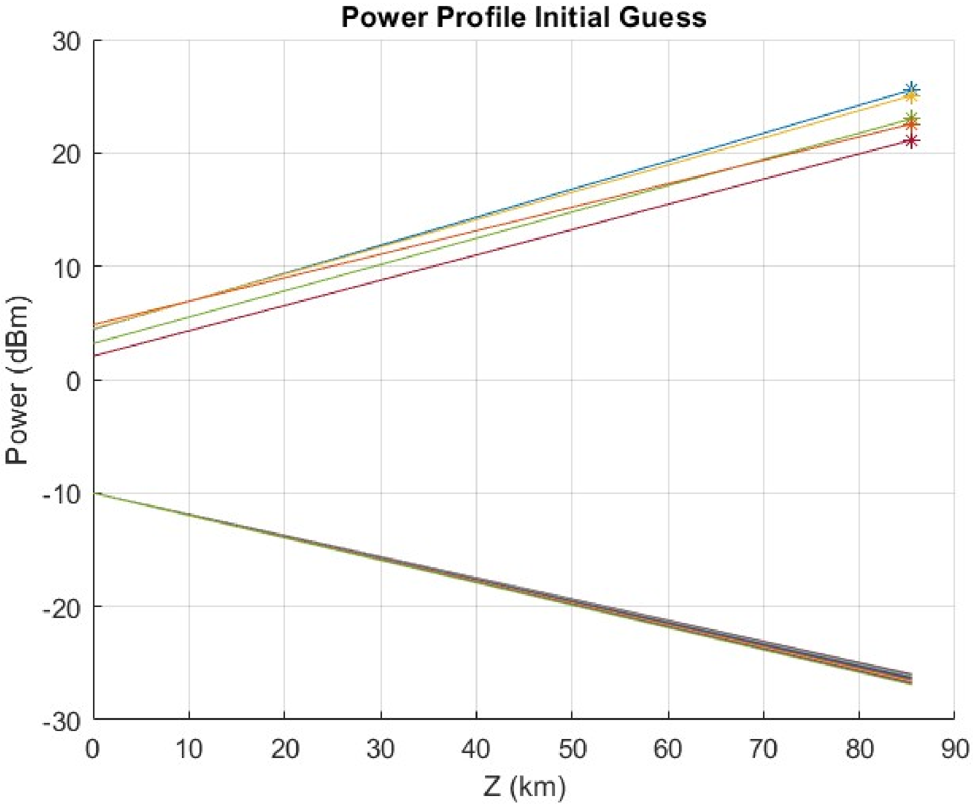

Figure 2.1: Power profile initial guess for a C+L system

Where the 5 curves with relatively high power are representing the pumps, and the star at the end of each of these curves is representing their boundary conditions. The variation of all the curves is linear with different slopes. This is because the attenuation is exponential and the figure is in logarithmic scale, moreover each channel has a slightly different value for the attenuation coefficient α_0, explaining the different slopes.

The method propagates this initial guess using equation 2.4. The resulting power profile is shown below:

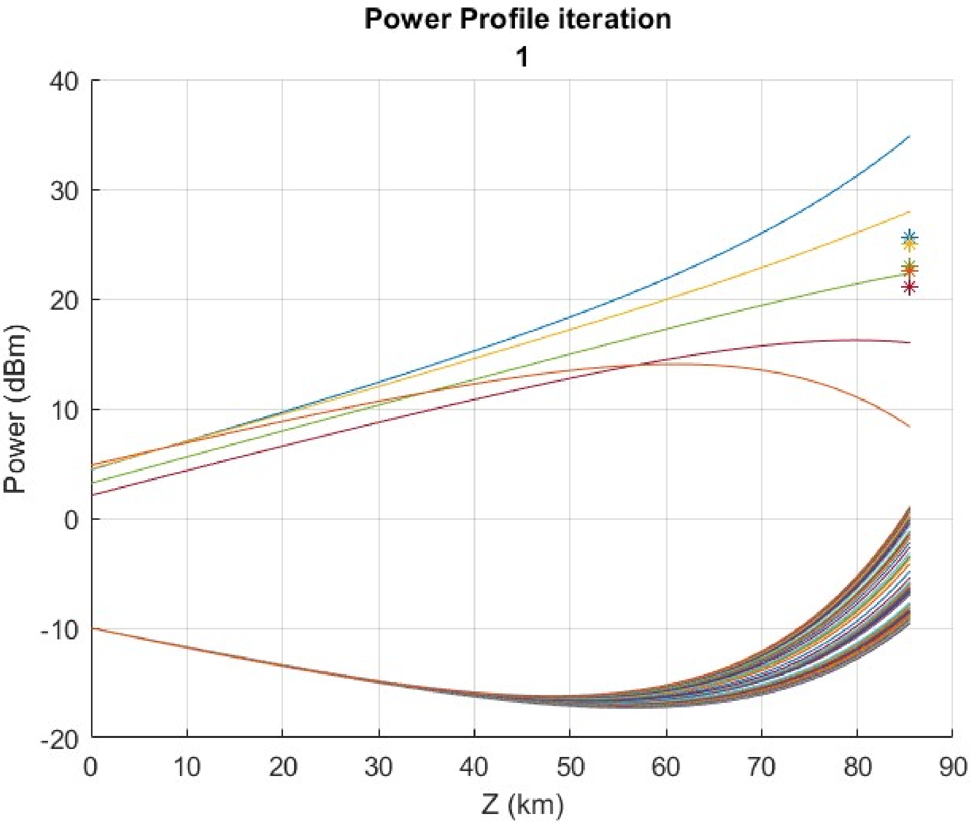

Figure 2.2: Power profile after one iteration for the same C+L system

It is seen that some pump values increased substantially, with respect to their boundary conditions while others have decreased. As a result, the signals are amplified in the last 30 Km of the fibre in a significant way. It is logical that an increase in pump values will be reflected in the signal channels as backward Raman amplification is significant, since the pumps are positioned around 5-15 THz away from the signals as described in section 1.1.2. For this reason, this method focuses on getting correct guesses for the pump profile to achieve an overall accurate power profile. Thus, the method fixes the signal values at the input of the fibre to the signal initial conditions and works on the pump profile by pushing it towards the boundary conditions.

For this purpose, the method uses a metric that measures how far the pumps at the end of the fibre are from their respective boundary conditions called Pump_Error:

$$\text{Pump\_Error}(i) = P_{\text{pump,in}}(i) - P(N_{ch_{signal}} + i, L) \tag{2.5}$$

where the first term is the boundary condition of pump i in Watts and the second term is the calculated value at the output of the fibre of pump i in Watts. It is noted that when the value of Pump_Error(i) is positive, the pump is lower than it should be and has been under-calculated while in the opposite case it has been over-calculated. Consequently, based on the Pump_Error metric, the method will correct the power profile guess for the next iteration by adding to it a correction value obtained by multiplying the profile of each pump i with a value proportional to Pump_Error(i) in the following way:

$$P(N_{ch_{signal}} + i, :) = P(N_{ch_{signal}} + i, :) + P(N_{ch_{signal}} + i, :) \cdot k \cdot \text{Pump\_Error}(i)$$

Leading to the correction equation:

$$P(N_{ch_{signal}} + i, :) = P(N_{ch_{signal}} + i, :) \cdot (1 + k) \cdot \text{Pump\_Error}(i) \tag{2.6}$$

The proportionality coefficient k is called the correction factor. Moreover, a distinction is made between corrections of under-calculation and over-calculation, as it will turn out later that the latter is more critical in divergent cases while the former is important in oscillation, more on this in section 2.2.2. This introduced two correction factors: Correction factor higher (CH) used in the case of over-calculation and Correction factor Lower (CL) used in the other case. An analysis of the correction factor values will be provided in section 2.2.2, for the current case, the values $CL = 1$ and $CH = 4$ are used. By doing this correction step, the method is changing the guess given to equation 2.4, to a better one, while keeping the boundary conditions the same, thus pushing the equation to give an estimation to the power profile slightly more accurate.

The corrected pumps according to 2.6 are represented in the graph below:

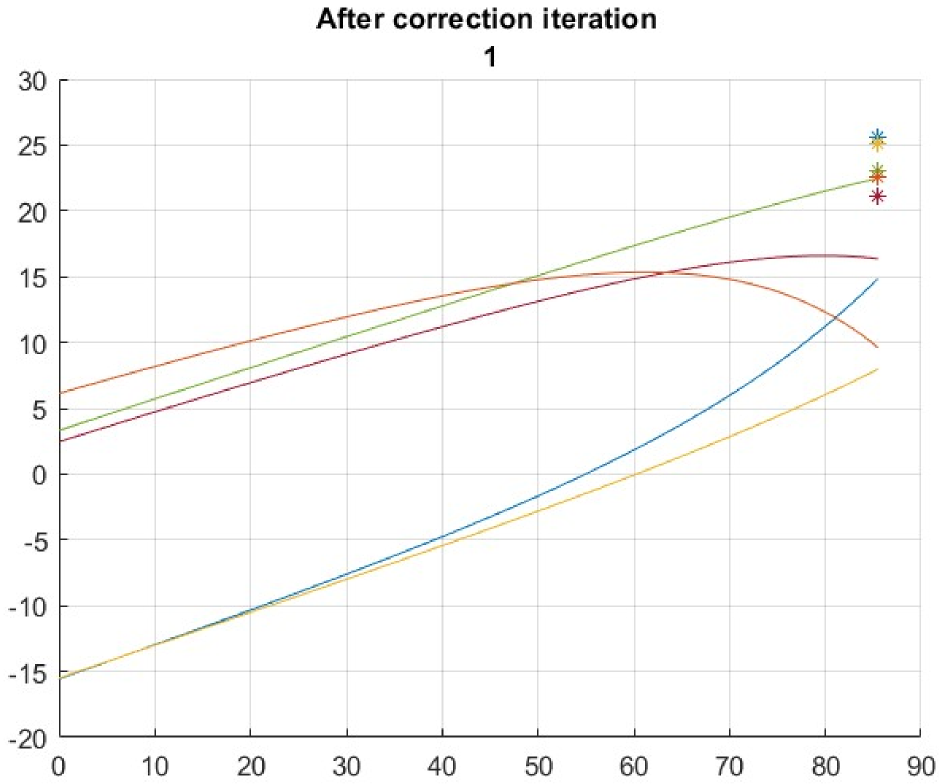

Figure 2.3: Power profile after first pump correction

The pumps that were under-calculated were pushed up by a small factor proportional to the error and are now closer to their boundary conditions. While the pumps that were above their conditions were pushed down in a more important way because of the higher value of CH. This is done to remove unnecessary numerically-introduced high powers in the equation and avoid divergence. In later iterations these pumps will be pushed back slowly towards their boundary condition. The corrected pumps are then propagated in the equation 2.4 and the following power profile is found in output of the second iteration:

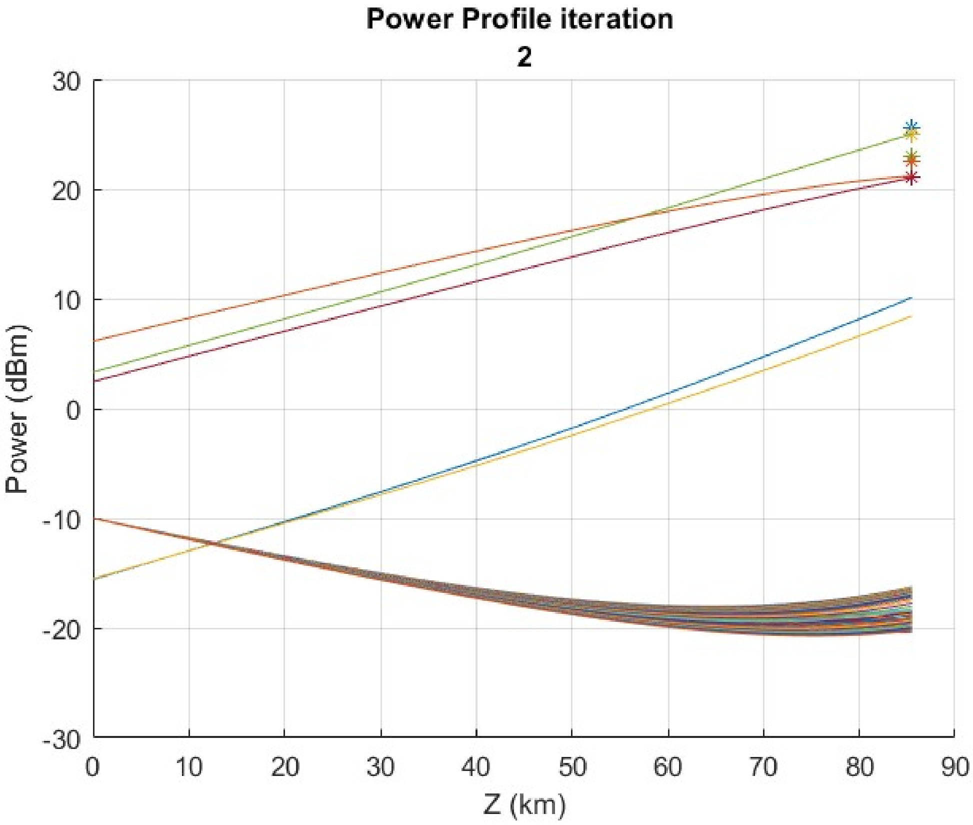

Figure 2.4: Power profile after two iterations for the same C+L system

After this second propagation ISRS between pumps is more visible, and some pumps are closer to their correct values. The output of the correction stage of the second iteration is shown:

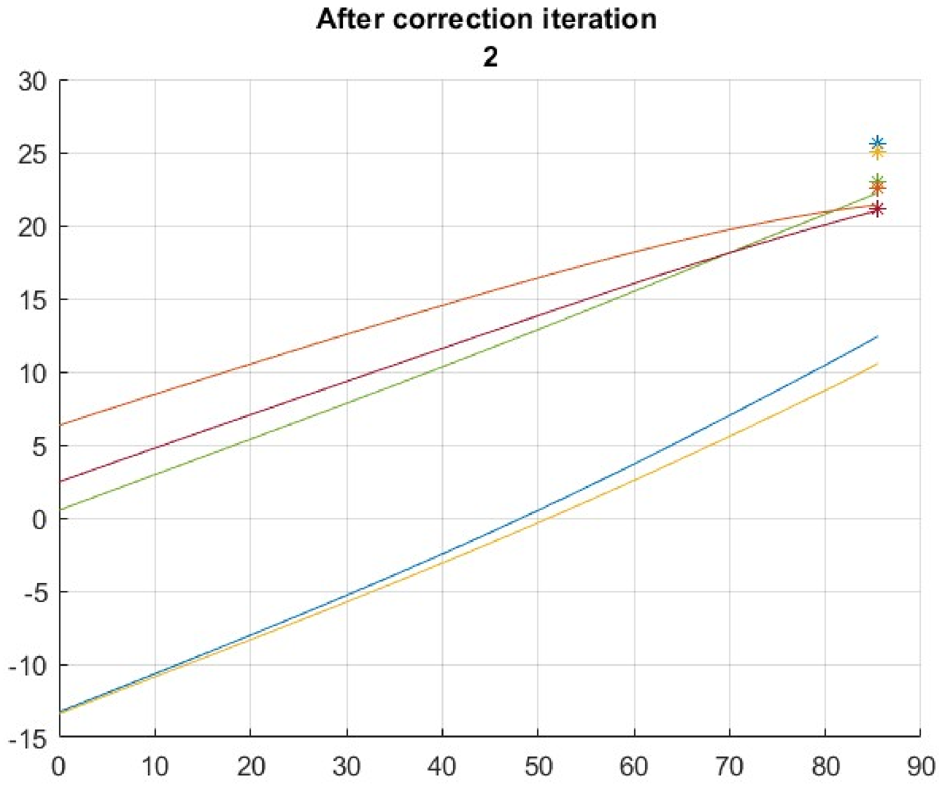

Figure 2.5: Power profile after second pump correction

From the pump color coding, it can be seen how the blue and yellow pumps that are below their boundary value are pushed up by the correction stage, while the green pump that is over-calculated has been pushed down. This new guess is a better one in terms of closeness to the boundary conditions than the previous one.

To show the inner workings of the method, two more iterations are shown down below, number 3 and 10. For each of them, the propagation output is shown along with the correction that followed, it is noted that all power profile curves are shown in logarithmic scale for illustration purposes:

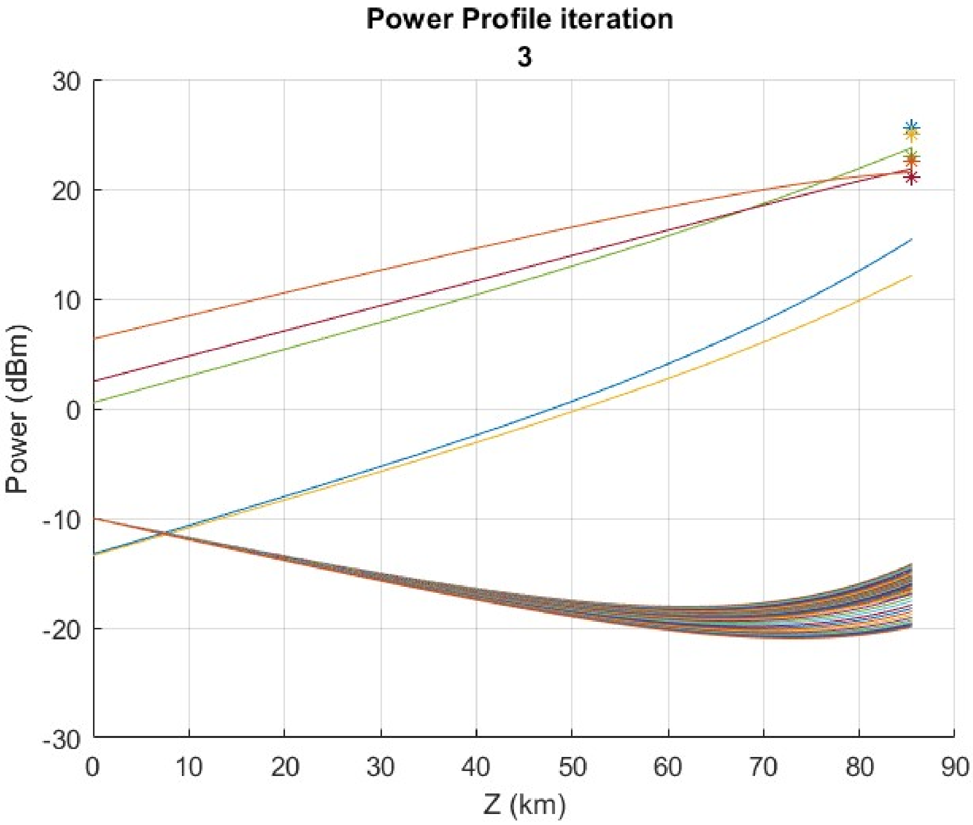

Figure 2.6: Power profile after three iterations of the same C+L system

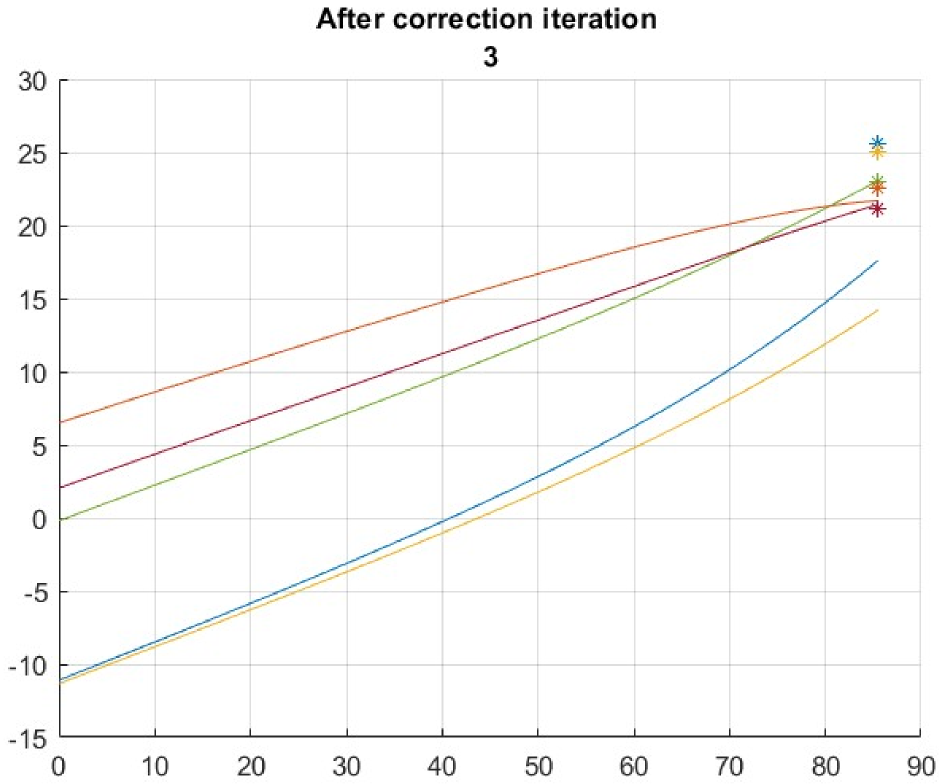

Figure 2.7: Power profile after third pump correction

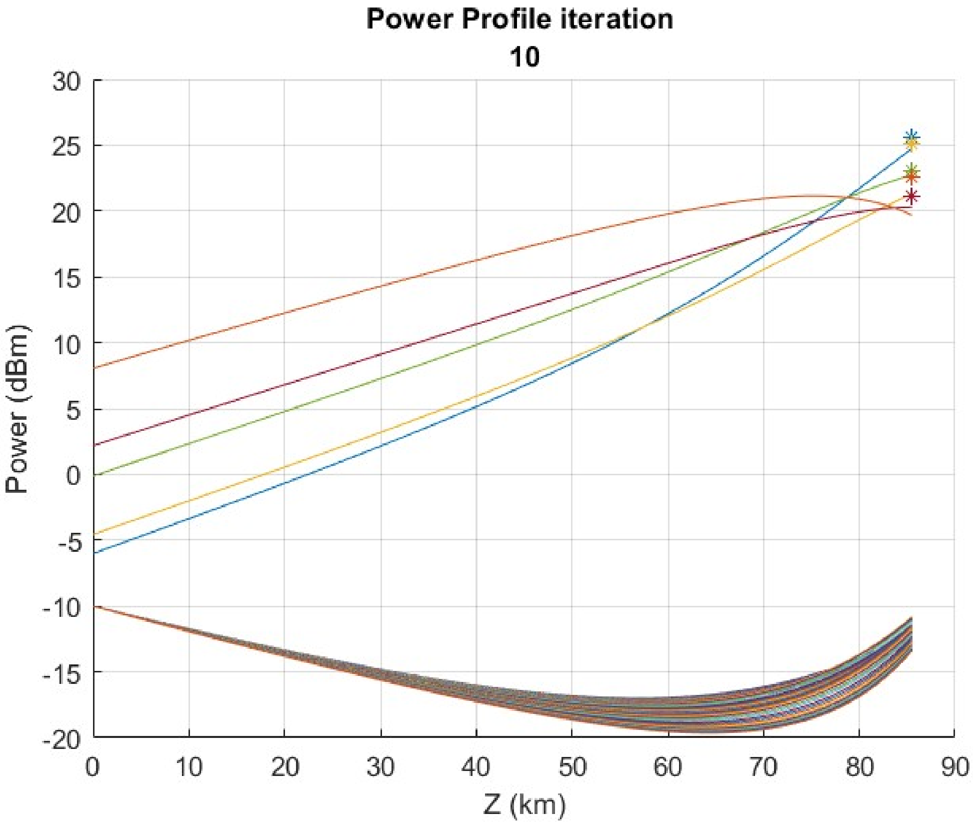

Figure 2.8: Power profile after ten iterations of the same C+L system

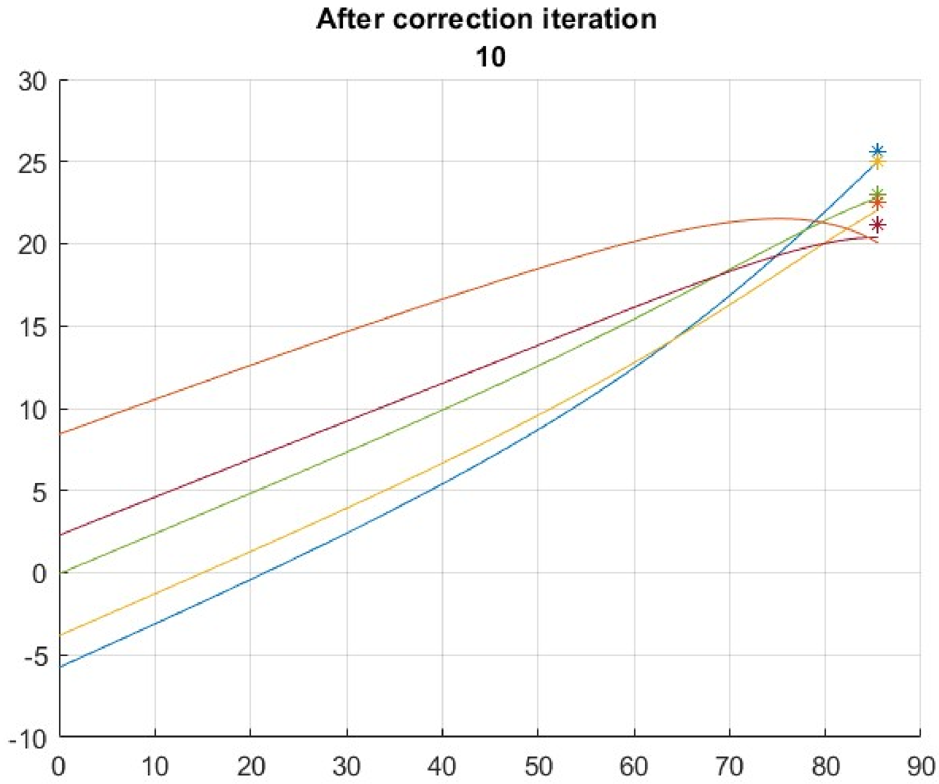

Figure 2.9: Power profile after tenth pump correction

Already at iteration 10, the pumps got close to their boundary values and the signals start to be amplified in a correct way.

The loop yields a converged result by fixing a value for the error tolerance, called *tol*, below which the pump power error at the end of the link is deemed satisfactory. In this example *tol* is set to 2e-3.

Over 40 iterations of propagation and correction, the power profile converges to a satisfactory solution in the sense defined at the start of this section. The converged solution is shown below:

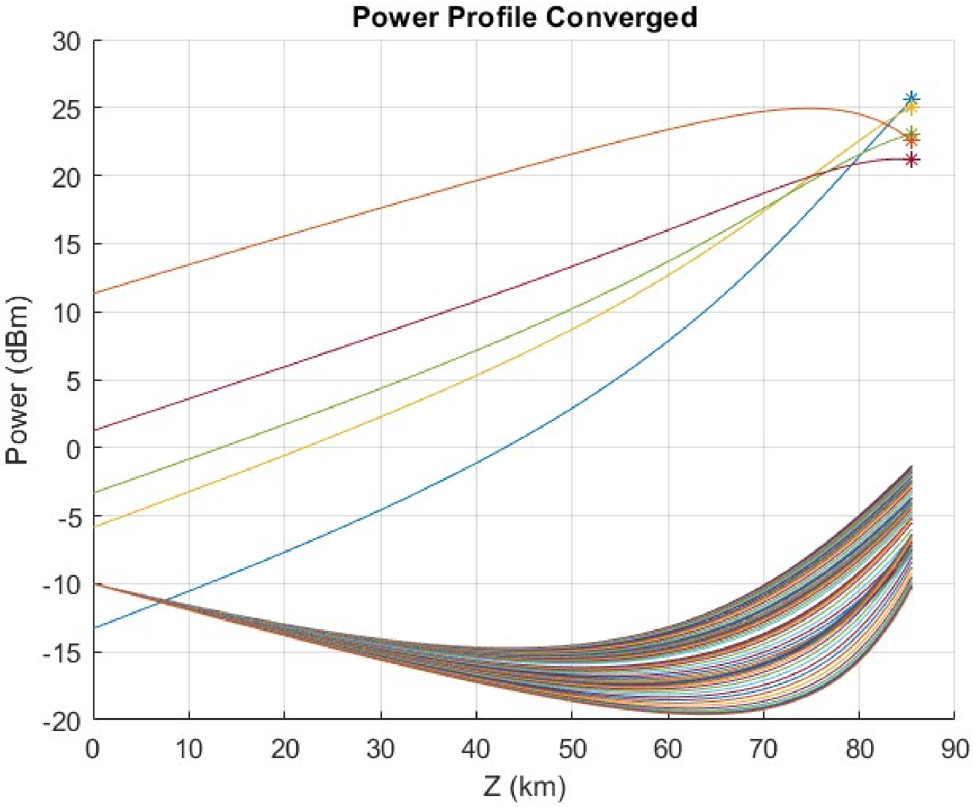

Figure 2.10: Converged power profile

Finally, a pseudo-code of the algorithm of DPC is presented below:

Algorithm 1 Dynamic Pump Calibration Method

```
Initialize: Set initial guess for power profile $P$ and error tolerance tol
while Any(|Pump_Error|) $\geq$ tol do
    Pump_Error($i$) = $P_{\text{pump,in}}(i) - P(N_{ch_{signal}} + i, L)$
    for each pump $i$ do
        if Pump_Error($i$) $> 0$ then
            Correct pump under-calculation:
            $P(N_{ch_{signal}} + i, :) = P(N_{ch_{signal}} + i, :) \cdot (1 + CL) \cdot$ Pump_Error($i$)
        else
            Correct pump over-calculation:
            $P(N_{ch_{signal}} + i, :) = P(N_{ch_{signal}} + i, :) \cdot (1 + CH) \cdot$ Pump_Error($i$)
        end if
    end for
    Propagate new power profile:
    $P = P(:,0) \cdot \exp\left(-\text{Direction} \cdot \alpha_0 \cdot Zk + \text{Direction} \cdot G \times P \times T_{trig} \cdot \Delta Z\right)$
end while
```

To show the convergence achieved by the method, a plot of the Pump_Error values with respect to the iteration number in the specific case being treated is shown:

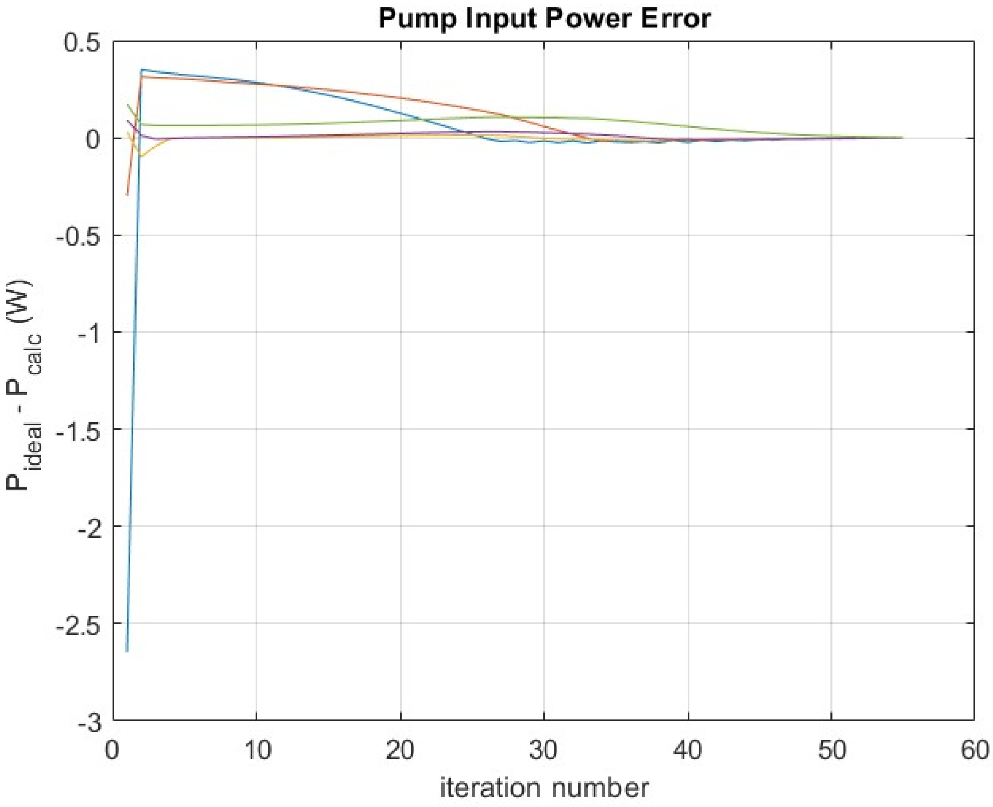

Figure 2.11: Pump error throughout the iterations

Convergence was achieved in the above case in 54 iterations leading to the following power profile in 0.344 seconds :

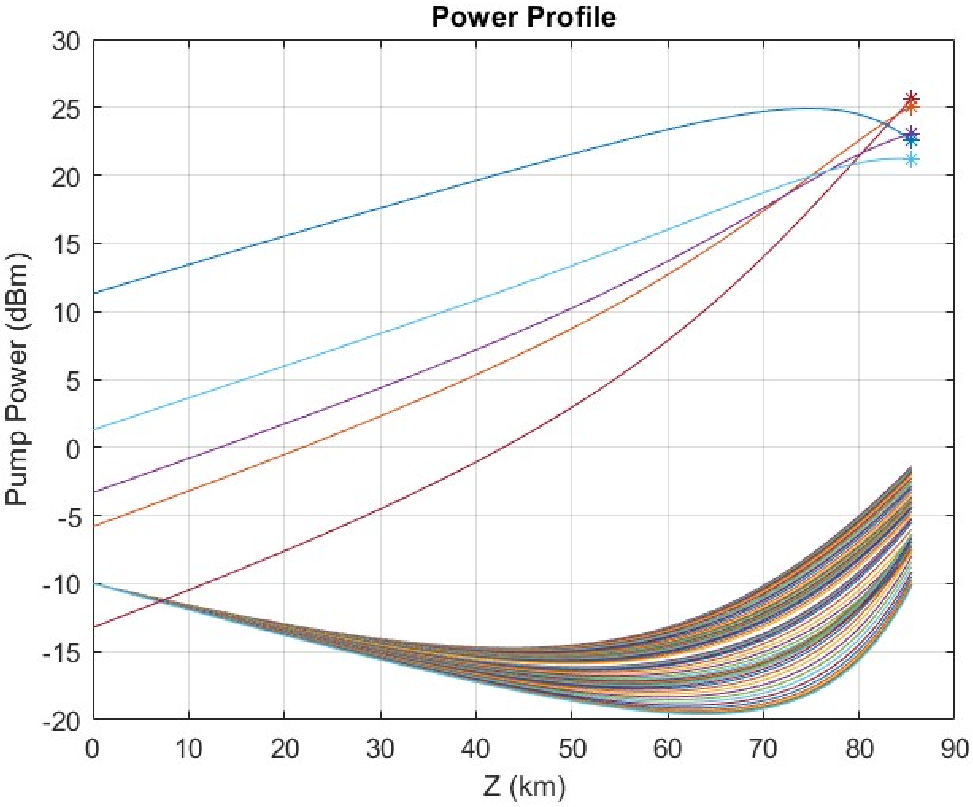

Figure 2.12: Power profile for the C+L system

For reference, the same system is run using MATLAB's bvp4c function, which further on will be referred to as the conventional method. It took 6.185 seconds for that method to converge. The maximum error at any point in the fibre and for any signal (pump error is not considered as it does not affect the following span and is not important for the CFM) between the 2 solutions is 0.0546 dBm. This is a time gain factor of approximately 20 for an error that can be considered negligible.

This section has treated the basic logic behind the Dynamic Pump Calibration Method and showed a pseudo-code of the operations done inside the loop. It also showed one result for a specific system, comparing it to the result provided by the conventional method, showcasing the potential speed-up to power profile calculation that this method can achieve. In the following a discussion of the technical issues faced during development and how they were overcome will be provided.

2.2.2 Technical issues

Some issues in convergence were faced in the deployment of the code described above for a subset of the boundary conditions imposed on the C+L system. Namely, these issues are divergence, and oscillation of the system around the correct value. In the following, each of these issues and how they were resolved will be discussed.

Divergence

The first issue faced, whose resolution was a crucial part of making the algorithm work for high pump power was divergence.

Because of the exponential nature of the vector equation 2.4, if the value of the power profile being fed to it gets too high, a positive feedback loop starts to take place that increases the values of the next guess, eventually leading to MATLAB NaN values in all of the power profile matrix P. This phenomenon was happening in the matter of a few iterations showing the high instability of this algorithm and the need to have a mechanism in place able to deal with that instability. To showcase divergence, the same C+L system is taken again with different boundary conditions. A uniform signal input power of 10 dBm is used and higher pump powers are obtained by scaling the initial case by a factor $Adjustment = 0.7$ like so: $[360, 320, 200, 130, 180]/Adjustment$ mW.

For simplicity, from this point on, to increase or decrease input pump power the parameter *Adjustment* will be tweaked while keeping the base powers the same, moreover the amount of pump power is going to be referred to using the value of *Adjustment* alone. It is also noted that because of the division sign, for lower values of *Adjustment*, higher pump powers are thus used. Below is a figure showing the Pump_Error curves throughout the iterations for the system described above:

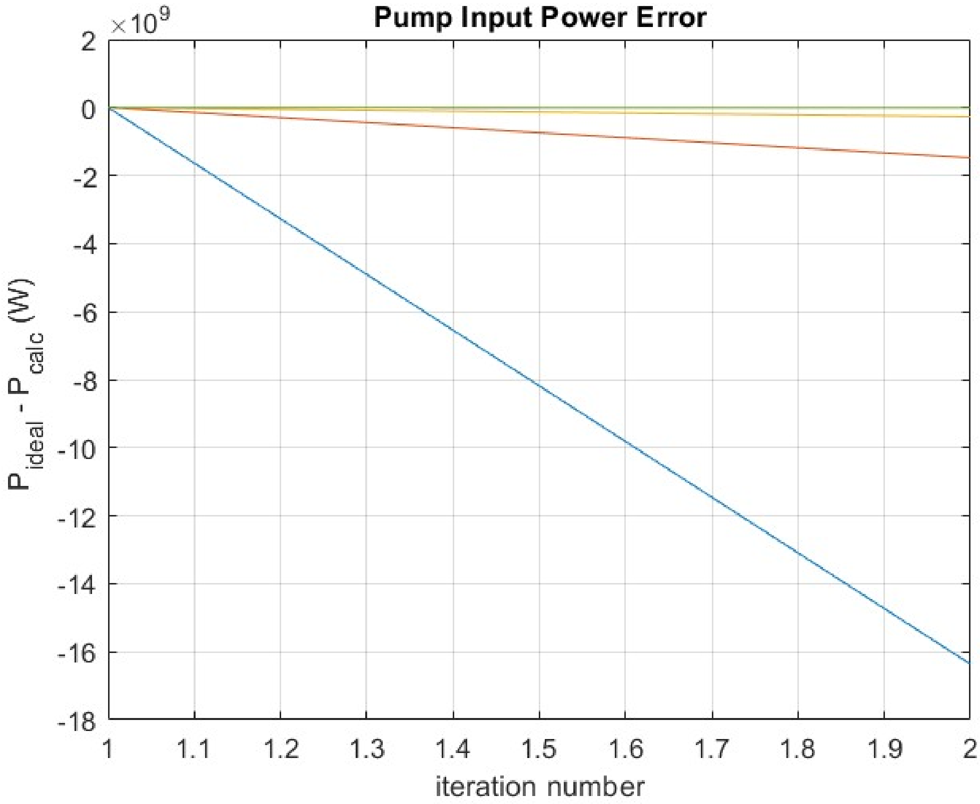

Figure 2.13: Pump error throughout the iterations

It is noticed that already from iteration number 2 a maximum error of $-1.176{\cdot}10^{10}$ is reached, after which the P matrix takes up all NAN values.
The first thing to notice is that divergence happened when pump and signal input powers were increased. The idea is that when power is high enough inside the initial guess, propagation through equation 2.4 leads to higher values in P. If those values are not extremely high, the correction phase should be able to push the pump profiles down. Picking higher values of the correction factor CH, promotes this behaviour.
However, if the calculated values of P are too high, the correction phase of the algorithm may not be powerful enough to enforce stability.
Consequently, another mechanism needs to be put into place to ensure divergence does not happen.
Because divergence occurs when high values are in the initial guess leading to a lot of over-calculations, and because this method fixes signals and works with pumps instead, the solution to this problem turns out to be to scale down the pump powers in the initial guess by a factor called factor$_{pump}$. This will ensure that the system

is in under-calculation most of the time and will converge slowly to higher pump values by slow correction. This solution is convenient because every other part of the algorithm can be kept the same, the only difference lies in the initial guess, where the pumps are scaled down, an example is reported below for factor$_{pump}$ = 3:

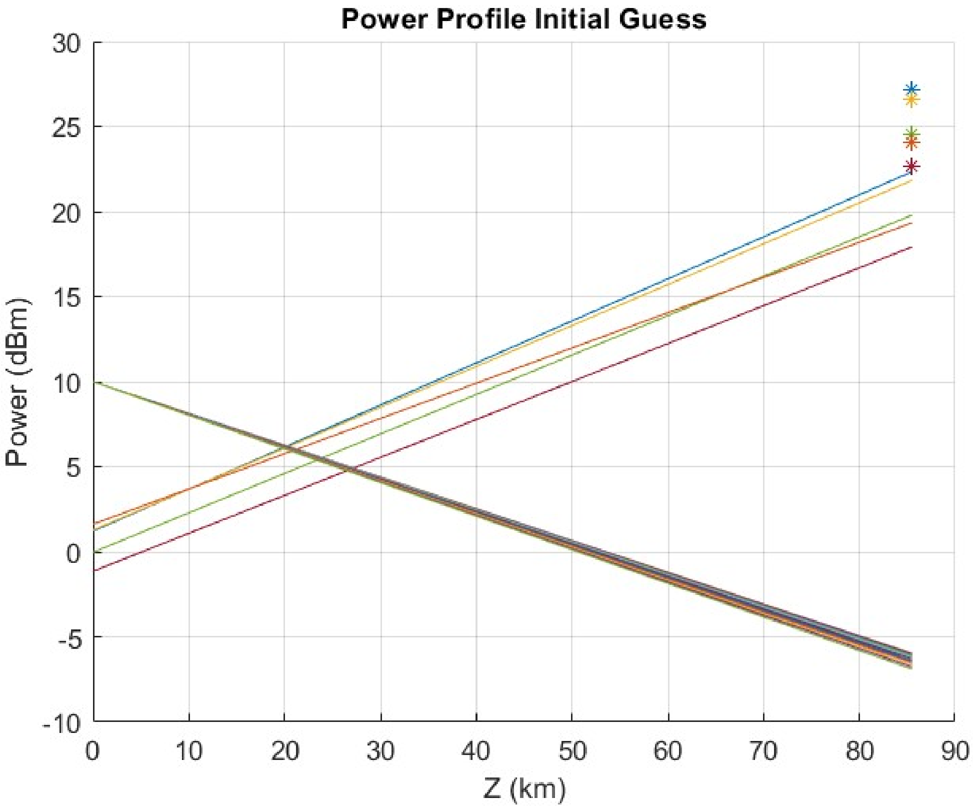

Figure 2.14: Power profile scaled down initial guess for a C+L system

The exact value of factor$_{pump}$ depends on the amount of power present in the equations, until this point it is set manually for each case. However, a method to automate the choice of this factor will be put into place in the final version of the algorithm which will be described in later sections.

To understand the impact of this technique, the highest realistic signal power used in practice is considered, which typically does not exceed 10 dBm. With this high signal power, the algorithm can only converge when the *Adjustment* value is approximately 1.1. This corresponds to a total pump power of less than 1.08 Watts. This is problematic because it limits the applicability of the algorithm, meaning it could not be used on systems with higher pump power requirements.

However, by employing pump power scaling, the algorithm is able to converge even with *Adjustment* values as low as 0.1, resulting in total pump powers exceeding 10 Watts—well beyond the normal industry range. This ensures that the algorithm can converge for virtually any real-world scenario, solving the problem of divergence. Below is the output of the algorithm for an extreme case of *Adjustment* $= 0.1$, using factor$_{pump} = 470$, $CH = 0.1$ and $CL = 0.001$.

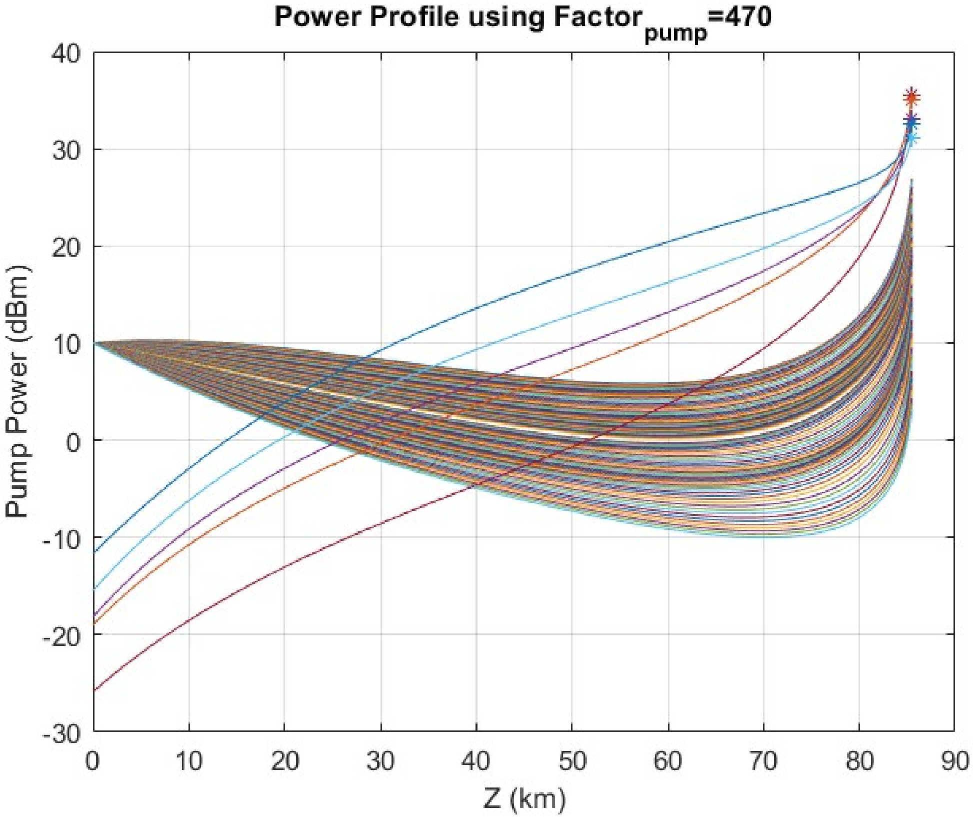

Figure 2.15: Algorithm solution for a case with *Adjustment*=0.1

Oscillation

A second issue encountered after divergence was resolved is oscillation. It happens when the power is not big enough to cause divergence, but the correction phase is not precise enough to allow the algorithm to converge to a solution whose error is within the tolerance. Of course higher tolerance values will result in less oscillating systems, but the value of $2e-3$ used in the algorithm is small enough to produce accurate solutions.

To study it, below is reported the Pump_Error curves of a case where oscillation happens. This time the C+L system is used with 5dBm of signal power, *Adjustment* = 1 while the algorithm parameters are CH=10 and CL=2 (While treating oscillation, factor$_{pump}$ is considered big enough to not cause divergence):

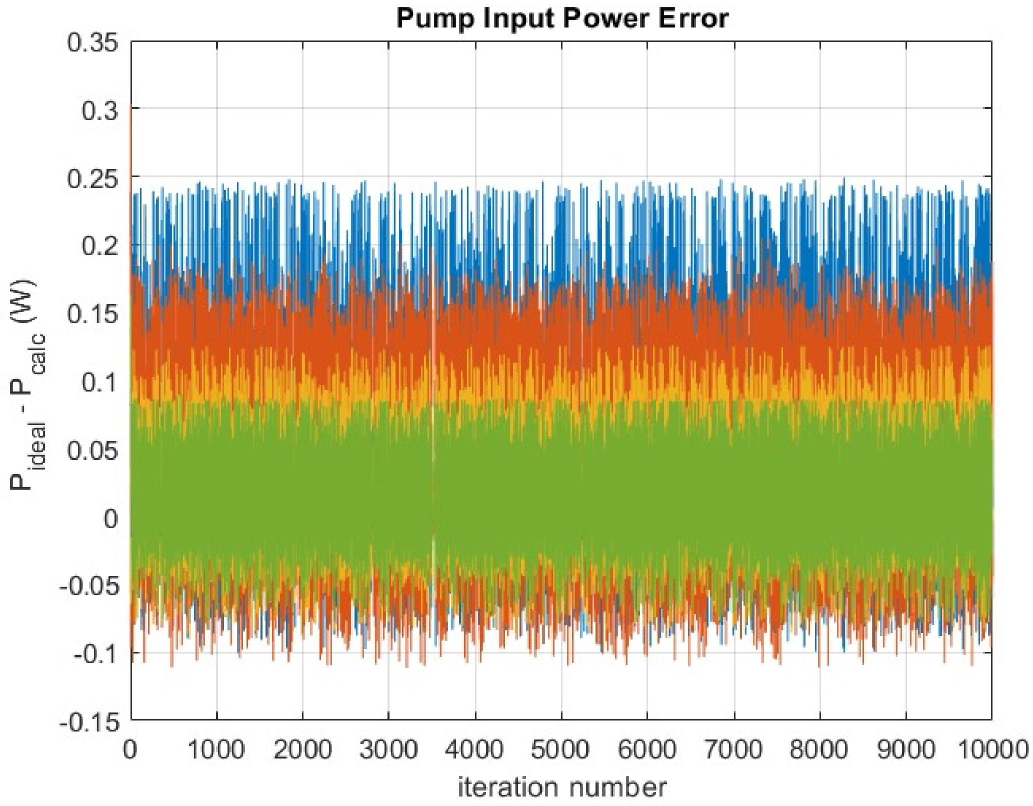

Figure 2.16: Pump_Error showcasing oscillation

Even after running for 10000 iterations, the algorithm could not converge to a satisfactory solution, instead it kept oscillating between error values of approximately -0.2 and 0.2.
To solve this issue, the same case is run again through the algorithm, this time with lower correction factors such as CH=4 and CL=0.4, the Pump_Error curves are reported below:

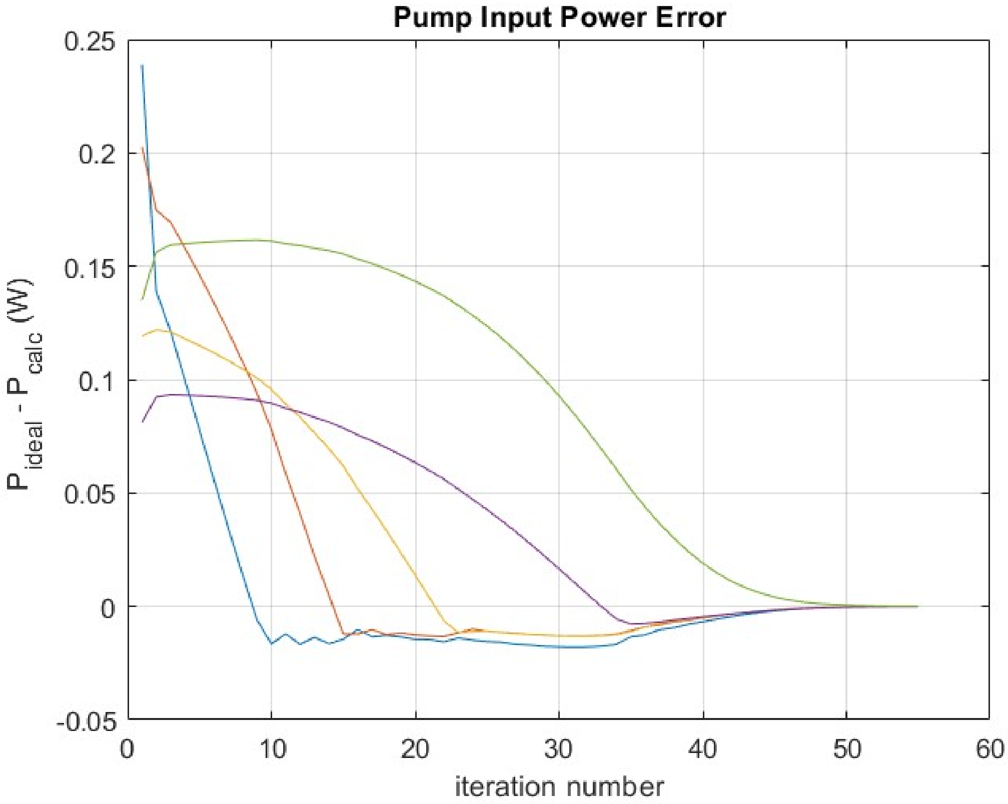

Figure 2.17: Pump_Error when oscillation is solved

It can be then concluded that lower correction factors will cause the system to converge. However, if very low correction factors are used, while the system will still converge, it will do so after taking many iterations, defeating the point of the method proposed by the thesis of being computationally efficient. Here is the Pump_Error curves for the same system with CH=0.04 and CL=0.004:

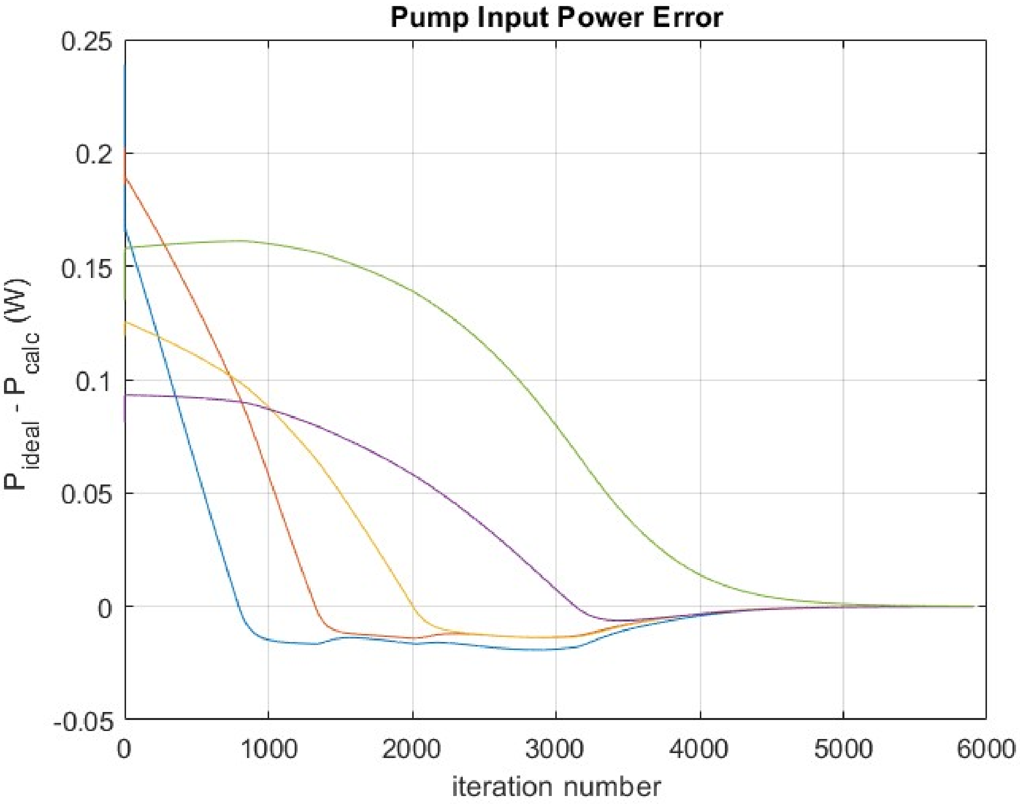

Figure 2.18: Pump_Error when oscillation is solved but with slow parameters

The curves seem similar to the ones before, however it takes 100 times more iterations for the algorithm to converge. Consequently, a trade-off between convergence and efficiency exists in the values of the correction factors. And a big issue of this method is to find the optimal values to use.
To further explore this trade-off, the values for CH and CL that produced convergence in the least amount of iteration were determined manually for 28 cases of varying signal and pump power (defined by the *Adjustment* factor). The findings are reported in the tables below:

$P_{\textbf{signal,in}}$ **(dBm) \Adjustment Factor**	**1**	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**
-10	7	6	4	3	7	6	3
0	6	6	5	4	5	4	1
5	4	5	5.5	5	5	2	4
10	5	7	6	6	4	1	2

Table 2.1: Optimal CH values for varying boundary conditions

$P_{\textbf{signal,in}}$ **(dBm) \Adjustment Factor**	**1**	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**
-10	0.7	0.6	0.2	0.04	0.035	0.015	0.009
0	0.6	0.6	0.3	0.15	0.05	0.035	0.015
5	0.4	0.5	0.35	0.2	0.1	0.05	0.025
10	0.5	0.7	0.5	0.3	0.2	0.1	0.09

Table 2.2: Optimal CL values for varying boundary conditions

From the tables it is seen that CH is not very critical for convergence as the values seem to be somewhat constant across all the cases. It is noted that those values seem to be relatively high with respect to CL. this is because CH is responsible of correcting pumps in cases of over-calculation that are divergence prone, so keeping a high CH value is beneficial for avoiding divergence.

On the other hand, CL seems to be much more crucial as values bigger than the one mentioned in the tables lead to oscillating systems. What is worse is that the range of values seem to be very wide across the different case scenarios with a min-max ratio of $r = \frac{0.7}{0.009} = 77.7$.
Moreover, the relationship between the optimal CL values and pump and signal power is complex; for increasing pump power, optimal CL values tend to go lower however for increasing signal power these values tend to go higher. Table 2.2 shows the complex dynamics of the CL values and the difficulty of using a single value that leads to convergence every time. One could think to exploit these values to interpolate a surface $S(P_{signal,in}, Adjustment)$ that would give a good value of CL to use for some given boundary conditions but this approach could lead to over fitting the data and the surface might not generalize for other multi-band systems. Because of this analysis, even if the Dynamic Pump Calibration Method showed promising results with very fast computation of the power profile, it fails to generalize for a wide range of case scenarios. For this reason, another method will be developed that does not depend on correction factors: The Progressive Signal Injection Method.

2.3 Progressive Signal Injection Method

The Progressive Signal Injection (PSI) Method was thought of because of the need of a different algorithm that exploits equation 2.4 to compute the power profile without having to rely on very dynamic factors. Since the method will be used for a wide range of systems, its parameters should be ideally constant for all cases or at least slowly changing in an easily traceable way.

2.3.1 Loop description

The general logic of this method stems from the idea that instability in the algorithm seems to be due to the interaction between signals and pumps, the more power is exchanged between the two, the more the equations are unstable.

For this reason, the method starts under the assumption of undepleted pumps, meaning pumps that have not yet transferred power to the signals. Since the power transfer between amplifying channels is proportional to the power of both channels, this is accomplished by injecting signals with negligible power, achieved by scaling down the signals by factor$_{signal}$. For the pumps, instead of starting with pumps propagated in the fibre subject only to attenuation like in the DPC method, they are also subjected to ISRS. For doing so a system with only pumps is considered, forming an initial value problem, which can be efficiently solved using MATLAB's ode45 function. The scaled-down signals subject to attenuation are then added to the pumps to form the initial guess of the power profile. An example of such a guess is reported below for signal input power of 5 dBm and factor$_{signal}$ = 30:

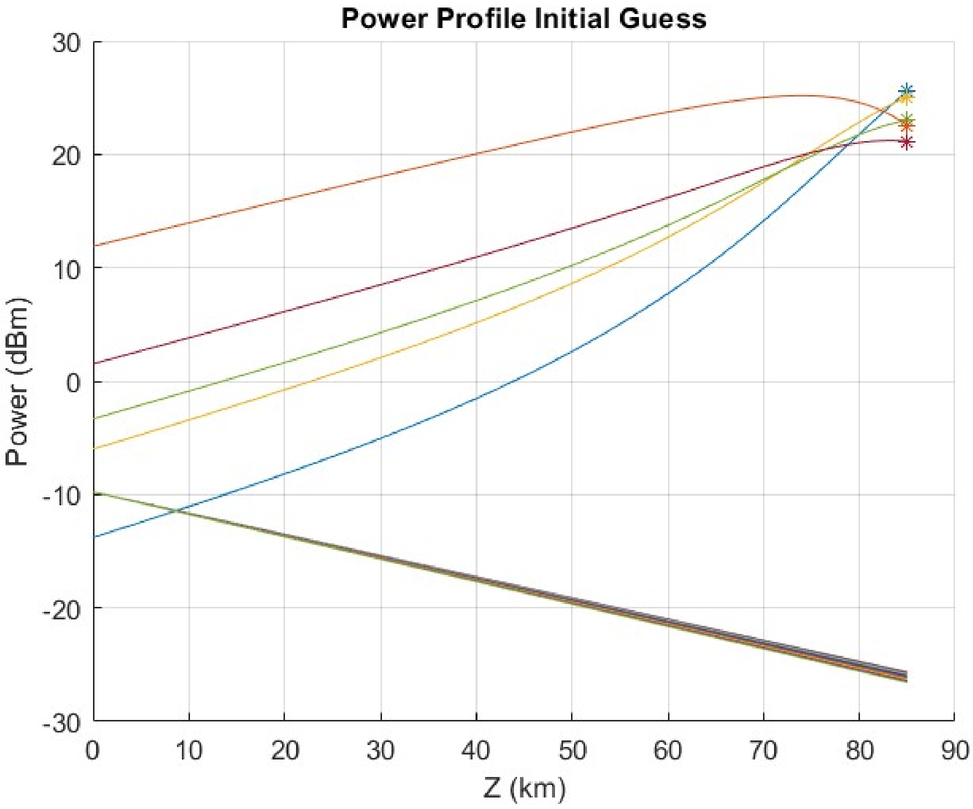

Figure 2.19: PSI initial guess

After propagating the guess through equation 2.4, the following power profile is achieved:

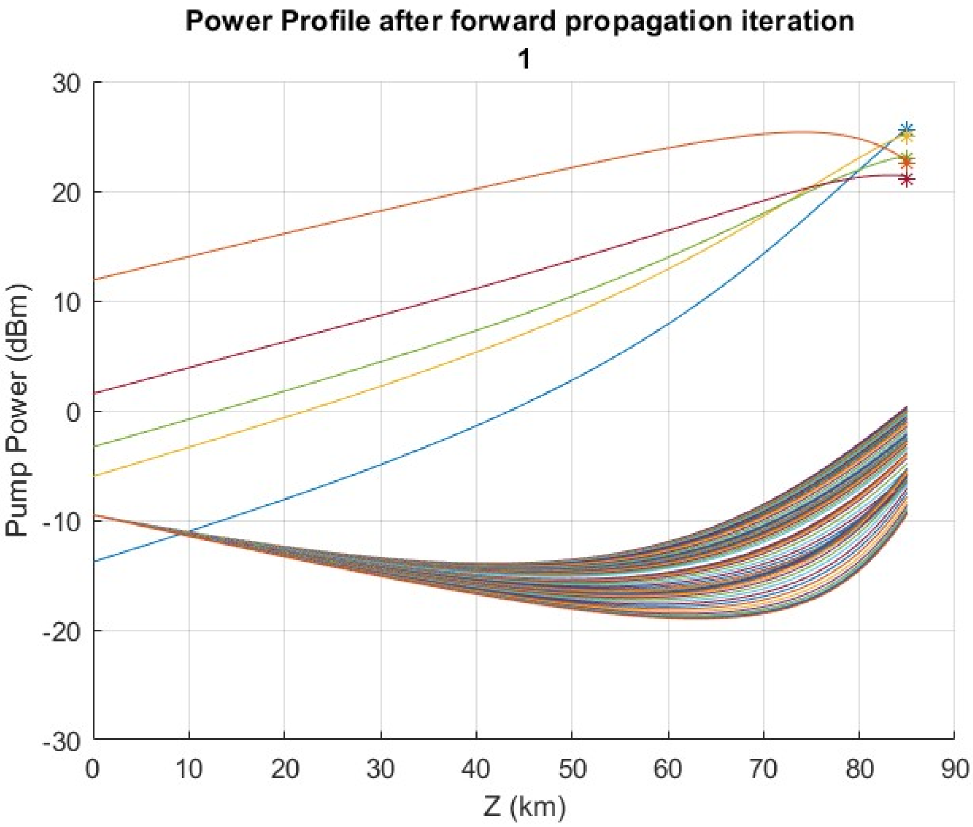

Figure 2.20: PSI Power profile output after 1 iteration

The solution seems to be immediately a good one, for a system with a signal input power of -9.52 dBm that is. The method was able to achieve an almost satisfactory solution from one single iteration. In fact, If the pumps can be considered almost undepleted because signal power is minimal, something that becomes exactly true for $\text{factor}_{\text{signal}} \to \infty$, then the method is calculating the power profile of pumps and signals in a separate fashion. Which is dividing the boundary value problem into two initial value problems. The method then proceeds by moving the signal power up by steps of step_{dBm}, a parameter of the method. In each iteration, the signal power is moved up, and the guess is propagated through the vector equation.
Moreover, because some interaction is anyway happening between the pumps and the signals, the pump power profiles do get affected by the propagation. In each iteration this should be corrected by scaling the pumps so they return to their boundary conditions exactly, like the following:

$$P(N_{ch_{signal+i}}, :) = P(N_{ch_{signal+i}}, :) \cdot \frac{P_{\text{pump,in}}(i)}{P(N_{ch_{signal+j}}, end)} \tag{2.7}$$

The algorithm keeps going until the signal power rejoins the initial condition exactly. By increasing signal power slowly and correcting the pump profiles each time, the system is moving from one converged solution of the system to the other, allowing to finally reach a correct solution with the given boundary conditions.

An algorithm describing the method fully in pseudo-code is presented below.

Algorithm 2 Progressive Signal Injection (PSI) Method

```
**Initialize:** Scale down $P_{\text{signal,in}}$ by factor$_{\text{signal}}$), set step size for increasing signal power step$_{dBm}$, and tolerance $tol$.
Solve the initial pump-only power profile as an initial value problem (IVP) using ode45.
Add the scaled-down signals subject to attenuation to form the initial guess for the power profile.
**while** $Any(|P_{\text{signal}}(0) - P_{\text{signal,in}}| > tol)$ **do**
    Correct pump profile:
    **for** each pump $i$ **do**
        Scale pump profile to match boundary condition:
        $P(N_{ch_{signal}} + i, :) = P(N_{ch_{signal}} + i, :) \cdot \frac{P_{\text{pump,in}}(i)}{P(N_{ch_{signal}}+i, L)}$
    **end for**
    Increase the signal profile $P_{\text{signal}}$ by step$_{dBm}$ to form the new guess.
    Propagate the guess using the vector equation:
    $P = P(: ,0) \cdot \exp\left(-\text{Direction} \cdot \alpha_0 \cdot Zk + \text{Direction} \cdot G \times P \times T_{trig} \cdot \Delta Z\right)$
**end while**
```

To show the convergence of the PSI method, a C+L system is considered with $P_{signal,in} = 5$ dBm, Adjustment factor=1 and step$_{dBm}$ is set to 0.05 dBm. The output of PSI is the presented in the following figure:

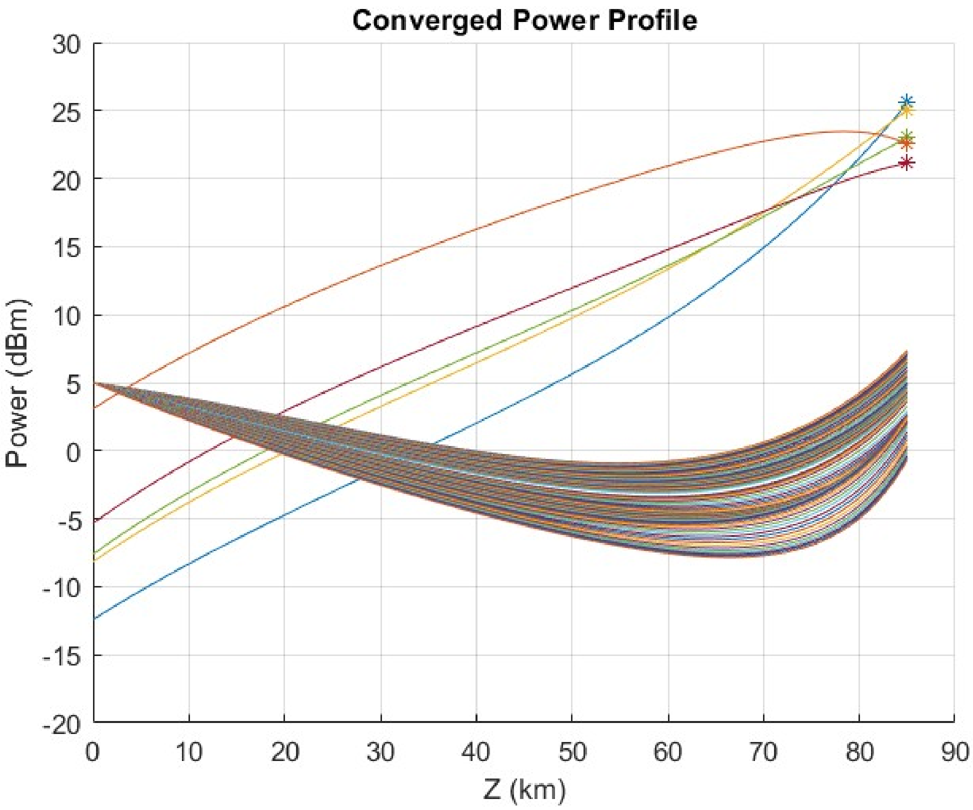

Figure 2.21: PSI Power profile converged output

Convergence was achieved in the above case in 297 iterations leading to the above power profile in 3.123 seconds. For reference, the same system is run using the conventional method which took 12.254 seconds. A negligible maximum signal error of 0.0349 dBm between the 2 methods is seen. A time gain factor of approximately 4 is achieved in this case.

It seems like this method is quite slower than DPC. However, this is largely influenced by the value of the method's parameters, namely step$_{dBm}$ and factor$_{signal}$ that provide a better solution at the expense of efficiency. To explore this trade-off, the same system is solved multiple times with varying parameters, the number of iterations and maximum signal error with respect to the conventional method's solution are reported in the table below:

Table 2.3: Iterations and Accuracy for Different Step Sizes and Signal Scaling Factors

$step_{dBm}$	**factor**$_{signal}$ **= 50**	**factor**$_{signal}$ **= 30**	**factor**$_{signal}$ **= 10**	**factor**$_{signal}$ **= 5**
0.05	Iterations: 341 Error: 0.0276 dBm	297 0.0349 dBm	201 0.0529 dBm	Diverged
0.1	Iterations: 171 Error: 0.0391 dBm	149 0.0459 dBm	101 0.619 dBm	Diverged
0.25	Iterations: 69 Error: 0.0864 dBm	61 0.169 dBm	41 1.751 dBm	29 20.633 dBm
0.5	Iterations: 35 Error: 0.164 dBm	31 0.191 dBm	21 3.391 dBm	15 11.261 dBm

Table 2.4: Table showing the performance of PSI for different values of step$_{dBm}$ and factor$_{signal}$

The relationship between the method's parameters and its performance can be easily understood. Higher step$_{dBm}$ lead to less iterations since the signal profile goes up to the initial condition faster. This however produces less accurate solutions, since the systems has less iterations to stabilize to the additional power injected inside of it. On the other hand, higher values of factor$_{signal}$ ensure that the initial no pump-signal interaction assumption of the method is respected, producing more accurate solutions. In some extreme cases, when factor$_{signal}$ is not big enough, considerable depletion of the pumps happens during the first iterations causing the algorithm to be too unstable and the solution to diverge.

If a maximum error of 0.05 dBm is accepted, then the fastest couple of parameters able to be considered are (step$_{dBm}$=0.1, factor$_{signal}$ = 30) achieving convergence for this system in 149 iterations, offering a time gain factor of approximately 8 over the conventional method.

In conclusion, another method for calculating the power profile has been presented, the Progressive Signal Injection method. On its own, it is slower than DPC but only depends on 2 parameters; step$_{dBm}$ and factor$_{signal}$ that are not crucial to the convergence of the method (given that factor$_{signal}$ is big enough), but are rather concerned with the trade-off of accuracy and efficiency. This method solved the problem of the DPC method having too dynamic parameters such as the correction factors. In the following, a discussion of some technical issues faced by PSI is presented along with how the method was modified to overcome them.

2.3.2 Technical issues

Similarly to the previous DPC method, even if for them cases it is working perfectly fine, PSI also faces some technical issues when subjected to certain extreme boundary conditions. In other words, the method is also divergence-prone for high power systems. Moreover, because of its easy for it to meet its stopping condition, it does not have problems in oscillation but rather in accuracy, as the latter cannot be imposed. In the following, both issues and their resolution will be discussed

Divergence

Divergence in PSI occurs for the same reasons of DPC mentioned in 2.2.2. Very high power values can cause the exponential nature of 2.4 to give out increasing values in output with no bound, leading to infinite values and eventually NANs. This happens even though pumps are corrected exactly to the profile they should be on using equation 2.7, simply because signals are amplified by these high power pumps and they are the ones who end up causing divergence, since no correction can be applied on them.

The approach to resolving divergence in PSI is inspired by the method used for DPC, specifically through the scaling down of the pumps by the parameter factor_{pump} in the initialisation phase. Additionally, because the correction in each iteration involves scaling the pumps to match their boundary conditions $P_{\text{pump,in}}$, this boundary condition itself must be scaled by factor_{pump}, resulting in the scaled boundary condition $P_{\text{pump,in,scaled}}$. At this stage, the method proceeds as before: scaling the signals gradually in each iteration while the pumps remain at their reduced values.

Once the signals have been fully scaled up, the pumps are then incrementally restored to their true boundary condition. Both the pumps and $P_{\text{pump,in,scaled}}$ are scaled back in a step-by-step manner over several iterations using a parameter $\text{step}_{dBm,pump}$ after which the profile is propagated through equation 2.4. Moreover, in each iteration, the pumps are corrected to align with the scaled boundary condition at that stage, $P_{\text{pump,in,scaled}}$. Once the pumps have been fully scaled back to their actual boundary condition, the entire profile is propagated through equation 2.4, and the method is considered to have converged.

Accordingly, the initial guess of the method, is one where both signals and pump are scaled down by different factors, both subject to attenuation with the pumps also subject to ISRS. The guess for a case where $Adjustment = 0.2$ and $P_{signal,in} = 5$ while $\text{factor}_{pump} = 10$ and $\text{factor}_{signal} = 30$ dBm is represented by the figure below:

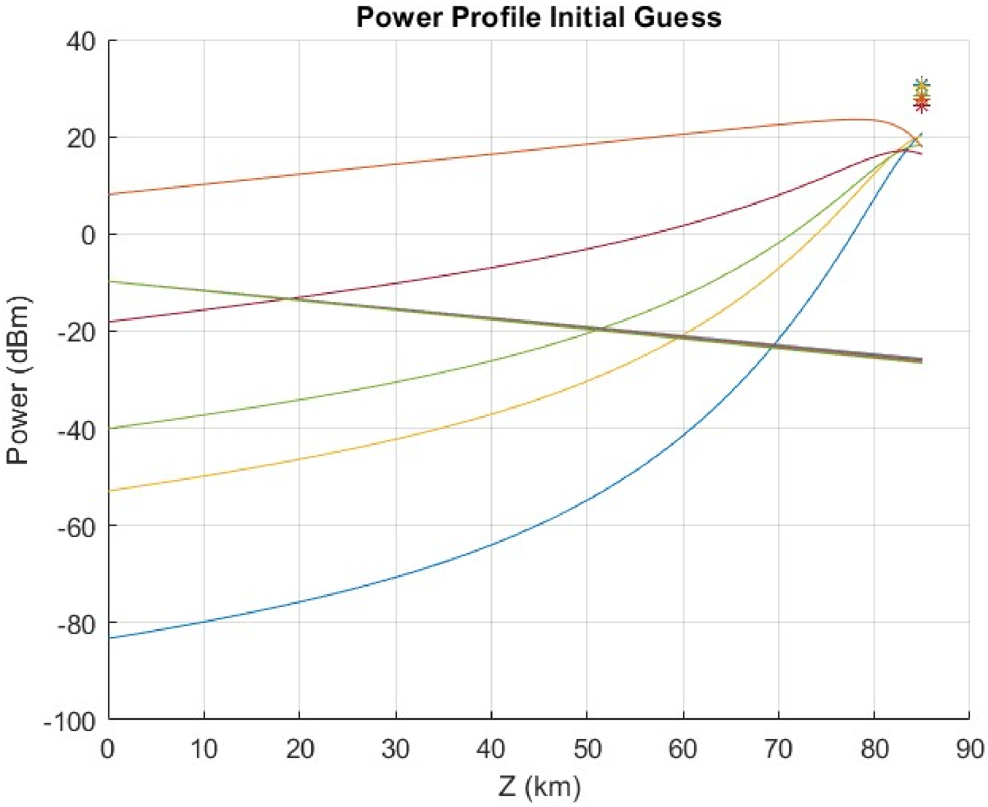

Figure 2.22: Modified PSI initial guess

The modified version of the PSI method is depicted in the algorithm below:

Algorithm 3 Modified Progressive Signal Injection (PSI) Method

```
Initialize: Scale down signal profile by factor$_{\text{signal}}$. Set step sizes step$_{dBm,signal}$ and step$_{dBm,pump}$, and tolerance tol.
Scale down pump profile and pump boundary condition ($P_{\text{pump,in}}$) by factor$_{\text{pump}}$
$P_{\text{pump,in,scaled}} = \frac{P_{\text{pump,in}}}{\text{factor}_{\text{pump}}}$
Solve the initial pump-only power profile using ode45.
Initialize flags signal_scaled_up and converged_flag to 0.
while $Any(|P_{\text{signal}}(0) - P_{\text{signal,in}}| > tol)$ or $converged\_flag == 0$ do
    Correct pump profile:
    for each pump $i$ do
        Scale pump profile to match scaled down boundary condition:
        $P(N_{ch_{signal}} + i, :) = P(N_{ch_{signal}} + i, :) \cdot \frac{P_{\text{pump,in,scaled}}(i)}{P(N_{ch_{signal}}+i, L)}$
    end for
    if signal_scaled_up $== 1$ then
        if $Any(|P_{\text{pump}}(L) - P_{\text{pump,in}}|) > tol$ then
            Increase pump power by step$_{dBm,pump}$
            Increase $P_{\text{pump,in,scaled}}$ by step$_{dBm,pump}$
        else
            signaling to get out of the loop
            converged_flag = 1
        end if
    else
        Increase signal power by step$_{dBm}$ to form the new guess
    end if
    Propagate the guess using the vector equation:
    $P = P(: ,0) \cdot \exp\left(-\text{Direction} \cdot \alpha_0 \cdot Zk + \text{Direction} \cdot G \times P \times T_{trig} \cdot \Delta Z\right)$
    if $P_{\text{signal}}(0)$ has reached its target then
        signal_scaled_up $= 1$
    end if
end while
```

It is mentioned that signal_scaled_up is a flag that becomes true when the method has finished scaling up the signals signaling the initiation of the pump scaling up process. While converged_flag is initialised to 0 and set to 1, to exit the loop in the next iteration, when the pumps are scaled back up all the way to their initial condition $P_{pump,in}$.

To understand the impact of this mechanism, the original PSI method described in 2.3.1 is stress tested with a high realistic signal power of 10 dBm and varying *Adjustment* factors. The minimum value of Adjustment factor for which convergence is achievable is 1.05 translating to a total pump power of 1.13 Watts. This makes the method as of 2.3.1, valid for a small subset of systems of low pump power.
Instead with the modified PSI, the achievable adjusted factor goes down to 0.25 (corresponding to 4 Watts of total pump input power) for which no divergence occurs. For this factor$_{signal}$ = 100 and factor$_{pump}$ = 100 were used. The converged solution for $P_{signal,in}$ = 10 dBm is presented below:

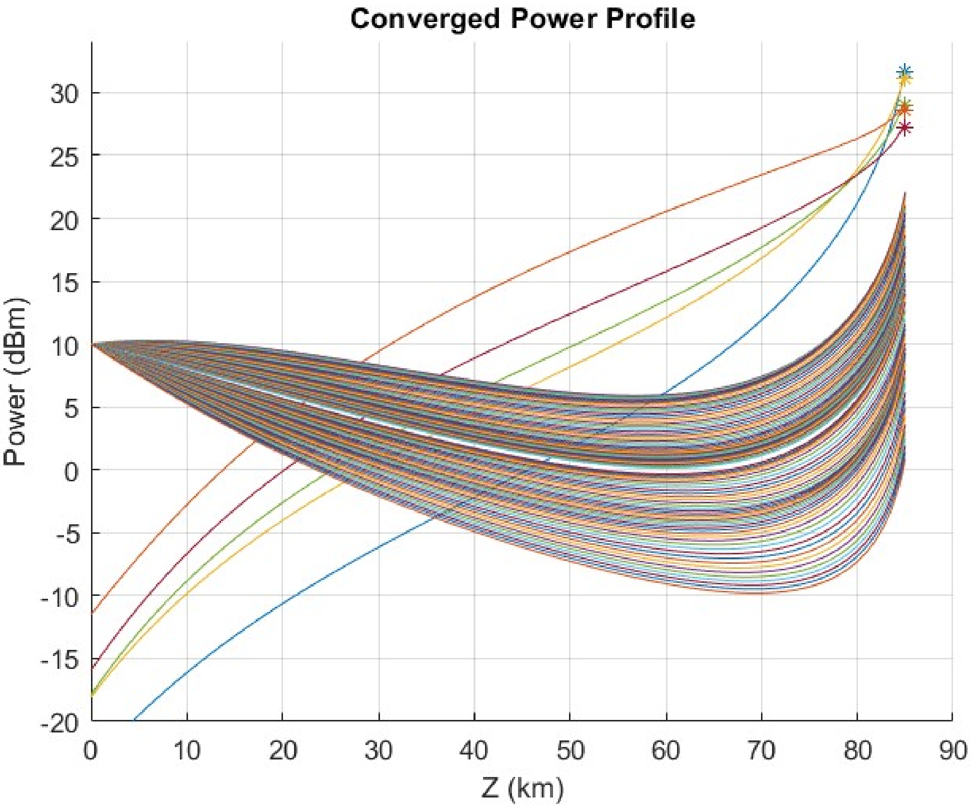

Figure 2.23: Converged solution with modified PSI method for *Adjustment* = 0.25

In conclusion, this allowed to virtually remove divergence for all realistic scenarios.
Unfortunately, this approach introduces more inaccuracy to the solution. This will be discussed more in detail below.

Inaccuracy

As shown by the table 2.4, the accuracy of the PSI method depends on its parameters and is traded-off with efficiency. For low pump powers such as the one used in the system represented in the table, the method is able to provide a fairly accurate solution with a low enough complexity. However, when using lower Adjustment factors representing higher pump powers, the scale down of the pumps as per algorithm 3 becomes necessary for convergence. Furthermore, because of the scale down and up of the pump power, the system is solved only once with the correct, scaled up pump power (in the last iteration), after-which no correction is happening. This means that there is no control onto where the pump power falls. In some cases it is far away from the boundary condition, which in turn is reflected in the signal powers affecting the whole accuracy of the power profile.

To illustrate this inaccuracy, the parameters found after the analysis of table 2.4, are used to solve systems with 5 dBm of signal power and varying Adjustment factors by the modified PSI method. Moreover, factor$_{pump}$ is set to 10 and step$_{dBm,pump}$ to 0.05 dBm. The different maximum signal errors with respect to the conventional method are shown in the table below:

Adjustment Factor	**Max Signal Error (dBm)**
1.0	0.139
0.9	0.119
0.8	0.123
0.7	0.119
0.6	0.121
0.5	0.109

Table 2.5: Maximum signal error for different adjustment factors using step$_{dBm,signal}$ = 0.1 dBm and step$_{dBm,pump}$ = 0.05 dBm

Each one of these runs took 348 iterations, which still offers a significant time gain over the conventional method, since these can be computed in around 3 seconds. On the other hand, the maximum signal error is higher than what is normally requested (at least under 0.1 dBm of error). Additionally, for some other cases of signal power, these errors can also be higher.

In an effort to lower this error, drastically small step$_{dBm,signal}$ and step$_{dBm,pump}$ are needed, which however will render the method too slow and still not achieve sub 0.1 dBm of error in some cases. An example with $\text{step}_{dBm,signal} = 0.01$ dBm and $\text{step}_{dBm,pump} = 0.02$ dBm is reported in the table below:

Adjustment Factor	Max Signal Error (dBm)
1.0	0.0725
0.9	0.0465
0.8	0.0471
0.7	0.0472
0.6	0.0805
0.5	0.109

Table 2.6: Maximum signal error for different adjustment factors using $\text{step}_{dBm,signal} = 0.01$ dBm and $\text{step}_{dBm,pump} = 0.02$ dBm

Even if in most cases, with these parameters the PSI method was able to achieve a sufficiently low error, each one of these runs took 1978 iterations to complete which takes roughly 16 seconds, no longer offering any efficiency advantage over the conventional method.

This limitation is mainly due to a non existing control over the quality of the solution in the PSI method. Instead of stopping according to some metric of solution quality as done when *Pump_Error* becomes low enough in DPC, this method stops when it is done scaling up signals and pumps. Moreover, after it is done, the method cannot be run multiple times without any modification to the guess, as its correction phase is imposing the values of the pumps rather than nudging them in the correct direction. This causes instability and eventually will cause the solution to diverge after a given number of iterations.

In conclusion, the PSI method is a good method for efficient solving of the Raman differential equations, since it depends on four easily tunable parameters that do not vary in a very dynamic way across the different system boundary conditions. These parameters are: $\text{step}_{dBm,signal}$, $\text{step}_{dBm,pump}$, factor_{signal} and factor_{pump}. However, it may sometimes output solutions with low accuracy, given the limited control the method has over the quality of the solution.

2.4 The Hybrid Method

Based on the two methods described in section 2.2.1 and 2.3.1 and the discussion of their limitations, there is the need for a final method that combines both DPC's accuracy control and PSI's slow parameter dynamic providing a fast, reliable and easily automated method for efficient power profile computation. In this section, this hybrid method will be described along with how it resolved the issues of both the methods.

2.4.1 Loop description

The merged algorithm works by allowing the modified PSI method to give an intermediate low accuracy solution. This solution is then passed as a guess to the original version of the DPC method which then corrects it in an iterative fashion to finally converge to the accurate solution.

In this way, instead of providing a stand-alone solution, the Progressive Signal Injection method provides the best guess possible for the Dynamic Pump Calibration method, thus allowing the latter to operate in a much narrower region around the true solution in the solution space. This approach exploits PSI's autonomous parameters and DPC's convergence power. Furthermore, because of the guess's closeness to the true solution, the correcting factors of DPC become much less sensitive to the boundary conditions, and a simple value of 1 for CH and 0.1 for CL is enough for convergence in most cases. In section 2.5, a method will be implemented to automate the values of CL when they are too big to achieve convergence, allowing the parameter to become fully automatic.

The exact way the algorithm works is actually very similar to the modified PSI algorithm presented in 3, except after both signals and pumps are done scaling up, the output of the method is fed to the original DPC algorithm presented in 1. The hybrid method's algorithm is reported below:

Algorithm 4 Hybrid Method

```
**Initialize:** Scale down signal profile by $\text{factor}_{\text{signal}}$. Set step sizes $\text{step}_{dBm,signal}$ and $\text{step}_{dBm,pump}$, and tolerance tol.
Scale down pump profile and pump boundary condition ($P_{\text{pump,in}}$) by $\text{factor}_{\text{pump}}$
$P_{\text{pump,in,scaled}} = \frac{P_{\text{pump,in}}}{\text{factor}_{\text{pump}}}$
Solve the initial pump-only power profile using ode45.
Initialize flag signal_scaled_up to 0.
**while** true **do**
    Correct pump profile:
    **for** each pump $i$ **do**
        Scale pump profile to match scaled down boundary condition:
        $P(N_{ch_{signal}} + i, :) = P(N_{ch_{signal}} + i, :) \cdot \frac{P_{\text{pump,in,scaled}}(i)}{P(N_{ch_{signal}} + i, L)}$
    **end for**
    **if** signal_scaled_up $==$ 1 **then**
        **if** $Any(|P_{\text{pump}}(L) - P_{\text{pump,in}}|) > tol$ **then**
            Increase pump power by $\text{step}_{dBm,pump}$
            Increase boundary condition $P_{\text{pump,in,scaled}}$ by $\text{step}_{dBm,pump}$
        **end if**
        **if** $All(|P_{\text{pump}}(L) - P_{\text{pump,in}}|) < tol$ **then**
            Propagate using the vector equation:
            $P = P(:,0) \cdot \exp\left(-\text{Direction} \cdot \alpha_0 \cdot Zk + \text{Direction} \cdot G \times P \times T_{trig} \cdot \Delta Z\right)$
            Feed PSI's guess to DPC method described in 1
            Break out of while loop when DPC converges
        **end if**
    **else**
        Increase signal power by $\text{step}_{dBm}$ to form the new guess
    **end if**
    Propagate the guess using the vector equation:
    $P = P(:,0) \cdot \exp\left(-\text{Direction} \cdot \alpha_0 \cdot Zk + \text{Direction} \cdot G \times P \times T_{trig} \cdot \Delta Z\right)$
    **if** $P_{\text{signal}}(0)$ has reached its target **then**
        signal_scaled_up $=$ 1
    **end if**
**end while**
```

2.4.2 Validation

To appreciate the effect of this hybrid approach, a previous case from table 2.5 is taken, where the adjustment factor is 0.5 and signal input power is 5 dBm. Previously, this case was solved by the modified PSI method, with an error of 0.109 dBm, a somewhat significant error. Moreover, The method converged in 348 iterations.

The same system is now solved using the Hybrid method, all the parameters of PSI remain the same as those of table 2.5, CH is set to 1 and CL to 0.1 as mentioned before. The method achieves convergence in 438 iterations with a maximum signal error of 0.0166 dBm. Almost a 10 fold increase in accuracy for the cost of 90 iterations.

Below is reported a figure representing the Pump_Error curves produced in DPC after PSI's solution was fed to it:

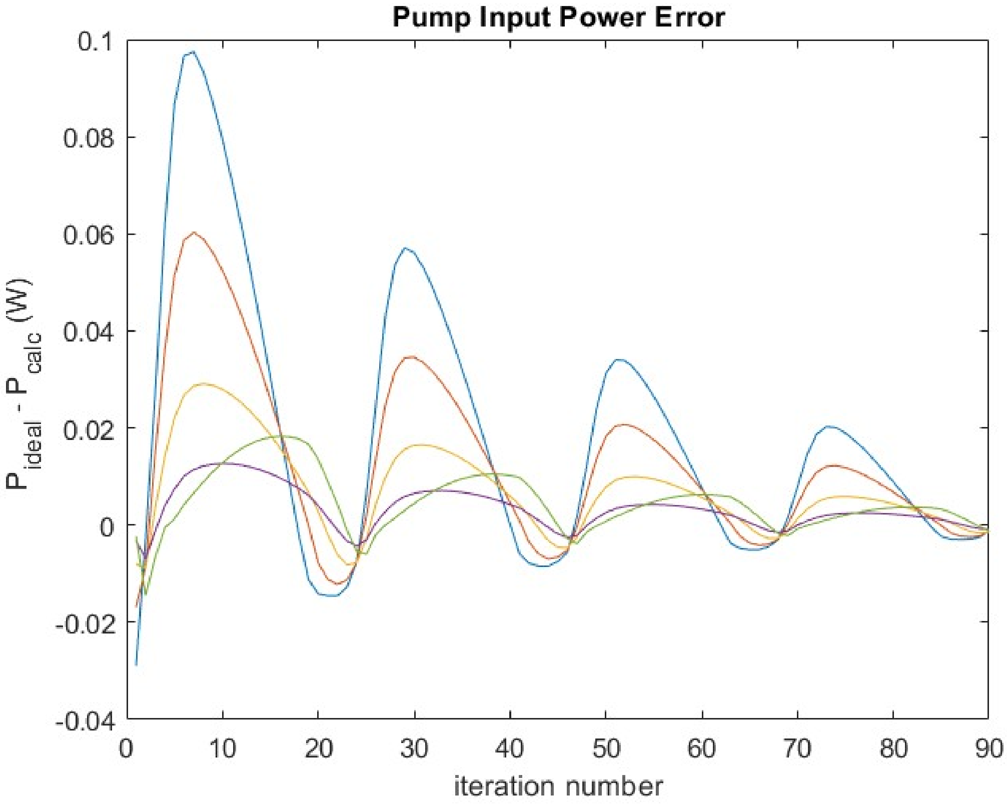

Figure 2.24: Pump_Error curves in the DPC phase of the hybrid method

As seen, in just 90 iterations the DPC method is able to impose an error on the input values of the pumps (with respect to the boundary conditions) smaller

than *tol*. Something that is reflected positively on the accuracy of the whole power profile.
As a validation for the Hybrid method, table 2.5 is repeated using the hybrid method with the same parameters described above and the results are shown below:

Adjustment Factor	Max Signal Error (dBm)	Iterations
1.0	0.0554	421
0.9	0.0626	411
0.8	0.0541	401
0.7	0.0396	392
0.6	0.0324	411
0.5	0.0166	438

Table 2.7: Maximum signal error and iterations for different adjustment factors using the Hybrid method

Out of all the cases, the maximum additional cost is for an Adjustment factor of 0.5 where 90 additional iterations were needed. For all the cases, the Hybrid method produced solutions with a sub 0.1 dBm error with respect to the conventional method. And in all of them, the same correction factors CH=1 and CL=0.1 were used. This simple test proves that the Hybrid method is able to solve both DPC and PSI's problems of highly dynamic parameters and low accuracy respectively.

Additionally, another benefit can be obtained from this hybrid approach. In fact, because the output of PSI is no longer used on its own as a solution but rather as a guess for DPC to improve, previous constraints put on the parameters in an attempt to produce accurate solutions can be loosened. The parameters in question are $\text{step}_{dBm,pump}$, $\text{step}_{dBm,singal}$ and factor_{signal}, previously set to 0.05, 0.1 and 30 respectively. Loosening these parameters would influence the efficiency of the algorithm allowing for less iterations and a significant time gain at the cost of accuracy, which however is no longer a big concern of PSI in the context of the hybrid method. To loosen the constraints on the parameters, the dBm steps can be increased and the signal scale down factor can be decreased. For example, the values 0.5 dBm, 2 dBm and 4 are used for the respective parameters and the test of table 2.5 is repeated, the following table is obtained:

Adjustment Factor	**Max Signal Error (dBm)**	**Iterations**
1.0	0.0451	143
0.9	0.0531	120
0.8	0.0386	137
0.7	0.0401	125
0.6	0.0440	158
0.5	0.0385	187

Table 2.8: Maximum signal error and iterations for different adjustment factors using the Hybrid method with loosened parameters and P_{in}=5dBm

From the table, it can be concluded that loosening the parameters, not only rendered the method much faster, saving more than 200 iterations and giving an output in less than a second, but did not have a negative effect on the accuracy, this is because DPC is taking care of that aspect.

To conclude, the approach of the hybrid method is a big step forward in the efficient solving of the power profile. It produces accurate solutions in a little number of iterations, and can ensure convergence in a big subset of boundary conditions. The only thing remaining is to ensure all of its parameters work well for different scenarios through parameter automation.

2.5 Parameter Automation

As mentioned in the theoretical part of the thesis, the main objective is to develop a method for efficient power profile calculation that can be inserted in the closed form GN model, to render it faster and enable it to be used in real-time application and perform efficient complex optimisation. Those objectives require the developed method to solve different UWB systems in a vast space of signal and pump power. For this reason, the method's parameters should all be set to some value that works in most realistic cases or at least have an adaptive mechanism that adapts them to an acceptable value for each case scenario.
The Hybrid method described in 4, has the following tunable parameters:

- CH
- CL
- step$_{dBm,signal}$
- step$_{dBm,pump}$
- factor$_{signal}$

- factor$_{pump}$
- Stopping number

All of the mentioned parameters have been seen before throughout chapter 2 except for Stopping number which is just a threshold on the number of total iterations, after which the method stops and declares it is not able to converge over a solution. This is useful to avoid infinite oscillating systems.

Most of these parameters can be given one value that works fairly well for all the cases. The values are assigned as the following:

- CH = 5, as seen in table 2.1, this parameter is not very dynamic and is mostly useful to avoid divergence. An average value of those figuring in the table is taken, as it seems to be working well.
- step$_{dBm,signal}$ = 2. This value allows the method to be efficient by scaling up the signals quickly while maintaining a good accuracy.
- step$_{dBm,pump}$ = 0.5. This value allows the method to be efficient by scaling up the pumps quickly while maintaining a good accuracy.
- factor$_{signal}$ = 4. Fast enough with no real effect on convergence or accuracy.
- Stopping number = 3000. As most systems converge before this amount of iterations.

The given values may not be the optimal ones. Some optimisation could be done in future research to determine the optimal values, which would definitely have an impact on both efficiency and robustness. However, heavily tested values have been given that seem to be working well in most cases.

The 2 remaining parameters that on the other hand do not seem to have one specific value that can be reused every time are CL and factor$_{pump}$. This was seen by the vastly different values taken by CL in table 2.2, and by the fact that the highest percentage of power inside the system is injected through the pumps, meaning factor$_{pump}$ needs to be dependent on it.
Consequently, 2 simple algorithms are put into place to make these 2 parameters adaptive to the specific case the Hybrid method is solving. Each of which will be discussed below.

2.5.1 Pump Factor

A simple way to automate the pump factor can be done by exploiting the speed at which divergence happens. As seen in figure 2.13, when the method diverges

because of too high pump power, it does so in a couple of iterations after which the loop is exited. Meaning that no significant loss of time happens if the system diverges and is rerun with a different value of factor$_{pump}$. The idea is to start the algorithm with no pump scale down (a value of 1), then if divergence happens, the system is rerun with a new value of factor$_{pump}$ picked from a vector of Factors containing increasing values, for example $Factors = [151015]$.
Divergence is detected by checking the P matrix for NANs. The algorithm is as follows:

Algorithm 5 Automated Pump Factor Adjustment Algorithm

```
Initialize: Define the Factors vector, k=1, factor_pump = $Factors(k)$, nanflag=1
while nanflag == 1 do
    Run algorithm in 4 with current $\text{factor}_{\text{pump}}$.
    Check for divergence:
    if any(isnan($P$)) then
        nanflag = 0;                         ▷ No NANs, algorithm converged
    else
        $k = k + 1$;
        if $k >$ length($\text{factor}_{\text{pumps}}$) then
            divergence_flag = 1;     ▷ Algorithm diverged for all pump scales
            Break;
        else
            Update pump factor: $\text{factor}_{\text{pump}} = \text{factor}_{\text{pumps}}(k)$;
        end if
    end if
end while
```

This way, the factors in Factors are tried one by one. If the method could not converge for any of those it assumes that power is too high and no convergence can be achieved. It breaks and signals what happened through the divergence flag. Otherwise, if no NANs are found in the converged P, $nanflag$ is set to 0 and the loop is exited.

2.5.2 Lower Correction Factor

As discussed in 2.2.2, the lower correction factor (CL) addresses the trade-off between oscillation and efficiency. If the value is too high, it can cause infinite oscillation with frequency proportional to the value of CL, while a very low value may result in slow convergence. The challenge lies in finding the right balance every time, as this trade-off shifts across different boundary conditions, as shown in table 2.2. In this table it is also seen that bigger values of CL work well in low pump power scenarios. As power increases the system becomes more sensitive and a finer CL should be used.

A good way to address this issue relies on early oscillation detection. The method can start with a big enough CL that works well for low pump powers, then a mechanism is put in place to detect oscillation as it is happening. The method can then lower the value of CL every time oscillation is detected.

To detect oscillation, a MATLAB function called "findpeaks" can be used on one of the Pump_Error curves, returning the values of the peaks and their position. Of course, not all peaks should be considered, but only those with significant magnitude. Oscillation is assumed when the number of peaks with a value bigger than $magnitude_{thresh}$ is bigger than a certain threshold $Peaks_{thresh}$. As as response, the CL value is reduced proportionally to the frequency of the oscillations as follows:

$$CL = \frac{CL}{c_0 \cdot length(peaks)}$$

where c_0 is a proportionality coefficient that determines how sensitive to the oscillation frequency the CL reduction should be. It is initially set to 1 and left for future optimisation.

Oscillation is checked every 100 iterations. At each check, MATLAB's findpeaks function is applied to one of the Pump_Error curves, examining the section from the last recorded CL adjustment (last_change) to the current iteration. Initially, last_change is set to 1 and updated whenever the CL is reduced. This approach ensures that oscillations that might emerge over a longer period can still be detected, while the peaks that triggered previous CL reductions are no longer considered in subsequent checks

The algorithm works as follows:

Algorithm 6 Lower Correction Factor Adjustment Algorithm

```
Initialize: Set initial correction factor CL, last_change = 1, oscillation check every 100 iterations, threshold values $\text{magnitude}_{thresh} = 0.3$ and $\text{Peaks}_{thresh} = 2$, and proportionality constant $c_0 = 1$.
while algorithm is running do
    if iter_number ∈ oscillation_check_values then
        Extract pump error:
        Pump_Error_1 = Pump_Error(last_change : iter_number, 1)
        Find peaks in the Pump Error curve:
        peaks = findpeaks(Pump_Error_1)
        Filter insignificant peaks:
        peaks = peaks(peaks > $\text{magnitude}_{thresh}$ × max(Pump_Error_1))
        if length(peaks) > $\text{Peaks}_{thresh}$ then
            Adjust correction factor:
            $\text{CL} = \frac{\text{CL}}{c_0 \times \text{length(peaks)}}$
            Update last change iteration:
            last_change = iter_number
        end if
    end if
    Continue with the rest of the algorithm.
end while
```

This algorithm is inserted at the end of the while loop of the DPC method described in 1, right after the propagation through equation 2.4. The initial CL value should be determined, some test showed that a value of 0.1 seems to be working well for low pump power, thus the method starts with it and then reduces it as it sees fit.

To show the utility of this mechanism, a case is considered where $P_{signal,in} = -5dBm$ and $Adjustment = 0.5$, the initial CL value is set to 0.1. The Pump_Error curves are reported below:

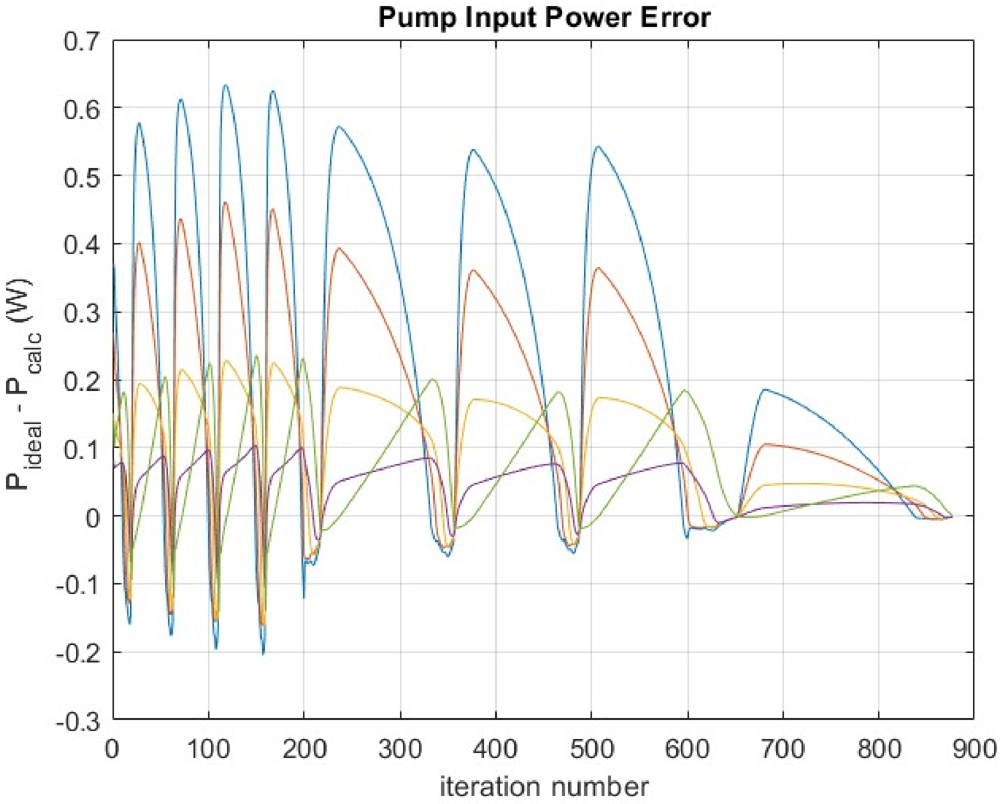

Figure 2.25: Pump_Error after the implementation of CL reduction

It is clear in the figure how at the 200^{th} and the 600^{th} iterations, the CL factor was reduced and consequently, the oscillation frequency reduced with it until a good value of CL=0.0083 was found allowing the algorithm to converge.

In this chapter, various methods exploiting equation 2.4 to efficiently solve the Raman-coupled differential equations have been thoroughly explored, able to provide accurate power profiles for the closed form EGN model. From the limitations of the DPC and PSI methods to the introduction of the hybrid method with adaptive parameters, a strong foundation has been established for solving these complex systems. The hybrid method is promising in addressing the challenges faced, particularly through its flexibility and reliability. The next chapter will present a comprehensive performance analysis, comparing this method with conventional approaches, and testing its applicability to UWB systems with non-uniform signal power. This testing is crucial to establishing the working conditions where the algorithm delivers optimal performance.

Chapter 3

Performance Evaluation and Comparison

Following the complete description of the final hybrid method developed throughout this thesis, along with all its parameters and the values applied, this chapter focuses on assessing its performance. For the proper use of the method, it is important to understand its reliability across different scenarios, and its time-saving potential in comparison to traditional power profile computation methods. The analysis includes extensive testing under extreme signal and pump power conditions and various ultra-wide band systems not previously explored during the method's development, along with a detailed comparison of speed, accuracy and convergence. Additionally, an overview of the average time gain achieved by the hybrid method in an optimization scenario, relative to the conventional approach, will be presented.

3.1 Stress Test

The hybrid method showed great potential in the efficient resolution of the coupled Raman differential equations. It seems to be converging on accurate solutions in a small number of iterations, this is demonstrated by table 2.8 which shows accurate convergence within less than 200 iterations for the C+L system with 5 dBm of uniform signal input power and varying pump adjustment factors. It is important to understand if this performance will remain the same for more extensive cases of signal and pump powers and eventually for wider UWB systems.

For this purpose, the hybrid method is used to solve C+L systems with adjustment factors going from 1 to 0.1 with steps of 0.1. It should be noted that the adjustment factor divides the original pump powers as shown in 2.2.2 to scale them up or down, meaning that a low Adjustment factor indicates high pump powers. Furthermore, the systems have uniform signal input powers ranging from -10 to

+10 dBm with steps of 1 dBm. For each case, the number of iterations is reported in the table below, where divergence and infinite oscillation are indicated with *Div* and *Osc* respectively:

$P_{\mathbf{signal,in}}$**(dBm) \Adjustment**	**1**	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
-10	55	187	155	430	181	840	*Div*	*Div*	*Div*	*Div*
-9	58	179	150	298	171	1211	*Div*	*Div*	*Div*	*Div*
-8	61	172	142	169	453	586	*Div*	*Div*	*Div*	*Div*
-7	66	160	145	169	540	980	*Div*	*Div*	*Div*	*Div*
-6	73	215	197	676	348	751	*Div*	*Div*	*Div*	*Div*
-5	108	59	192	361	270	702	1631	*Osc*	*Div*	*Div*
-4	228	48	186	155	280	360	1507	*Osc*	*Div*	*Div*
-3	219	85	180	146	254	470	868	*Div*	*Div*	*Div*
-2	89	143	175	226	370	423	1457	*Div*	*Div*	*Div*
-1	64	203	169	189	159	304	421	*Div*	*Div*	*Div*
0	99	195	164	138	366	186	510	*Div*	*Div*	*Div*
1	123	190	159	131	221	291	1069	*Osc*	*Div*	*Div*
2	135	183	155	145	138	255	581	*Osc*	*Div*	*Div*
3	123	177	151	135	169	251	376	2329	*Div*	*Div*
4	201	173	147	147	238	201	403	1467	*Div*	*Div*
5	197	170	145	117	171	335	553	877	*Div*	*Div*
6	194	168	143	115	159	177	320	*Osc*	*Div*	*Div*
7	192	167	142	113	180	154	581	1047	*Div*	*Div*
8	191	167	143	116	185	270	368	777	*Div*	*Div*
9	193	167	143	112	184	177	*Osc*	693	*Div*	*Div*
10	233	159	145	116	186	170	*Osc*	445	*Div*	*Div*

Table 3.1: Number of iterations until convergence for different signal and pump powers for CH=5 and CL=0.1

Below is a surface plot visualizing the same data, in which a value of -100 is considered to signal divergence while a value of 0 is considered if oscillation happens.

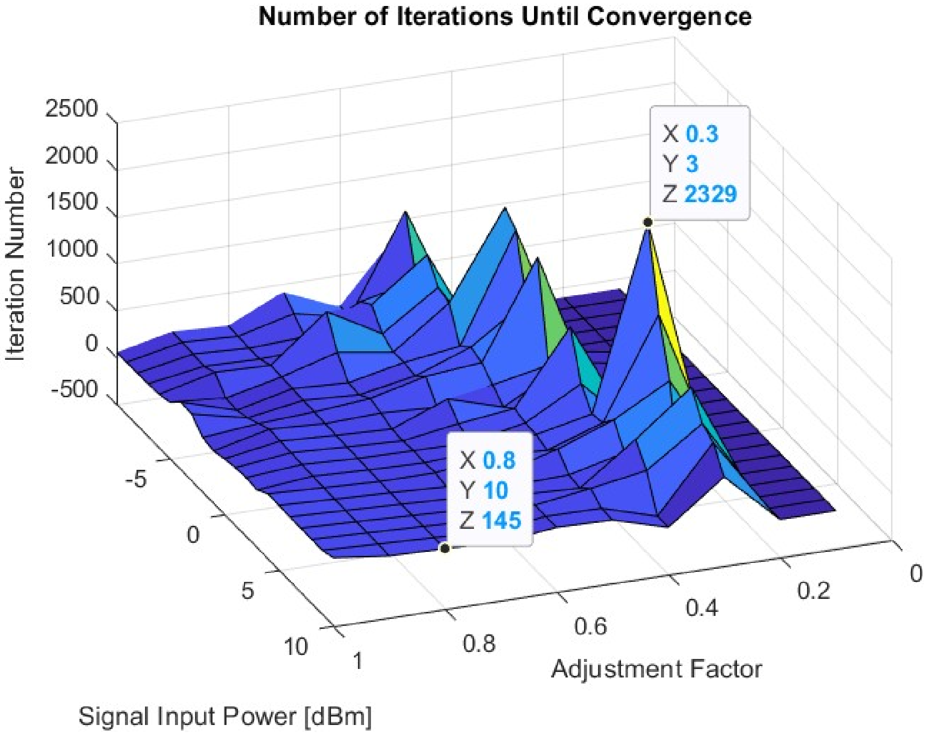

Figure 3.1: Iterations until convergence for the hybrid method with CH=5, CL=0.1, $step_{dBm,signal} = 2$, $step_{dBm,pump} = 0.5$ and $factor_{signal} = 4$ for various signal and pump power

From this data it can be seen that the method is able to achieve convergence for most cases of $Adjustment > 0.2$.
If a maximum total pump power of 3 Watts is considered, corresponding to $Adjustment_{min} = 0.4$, and signal input power is bounded between -5 and 10 dBm, the hybrid method converges in all but 2 cases; namely the $(P_{signal,in}, Adjustment)$ couples (9 , 0.4) and (10 , 0.4) which can be considered extreme cases.
Furthermore, the average number of iterations taken by the method to achieve convergence in the filtered case mentioned is 263. For reference, 263 iterations are computed in less than two seconds using an Intel Core i7-1255U processor (1.7 GHz).

To search for better CH and CL parameters, the above test is repeated six times for values of CH in [1 3 5] and CL in [0.1 0.05]. Six matrices, similar to the one above, are obtained. The average number of iterations for convergence in each case, along with the number of converged cases, is reported in the table below:

CH	CL = 0.1	CL = 0.05
1	395.6733	367.7152
3	313.6776	352.7417
5	321.5197	395.6340

Table 3.2: Average iterations for different CH and CL settings

CH	CL = 0.1	CL = 0.05
1	150	151
3	152	151
5	152	153

Table 3.3: Number of converged cases for different CH and CL settings

From the above tables, it can be understood that the values of the correction factors CH and CL do not have a big effect on convergence as it is achieved for almost the same number of cases across all values considered. Moreover, the pair $CH = 3, CL = 0.1$ seem to achieve a slight gain in efficiency (through average iteration number) with respect to the case $CH = 5, CL = 0.1$. Consequently it is beneficial to switch to $CH = 3$ in the method parameters, which will be applied in the following tests.

This study shows how the method is able to provide a solution for the Raman coupled differential equations in most cases of a C+L system with uniform input power. It also shows that the method does so in a relatively low average number of iterations, completing the initial objective.

3.2 Comparison With Conventional Method

In this section, a comparison between the developed hybrid method and the conventional method based on MATLAB's bvp4c function is presented. The focus will be on the speed of each method in different UWB systems, such as the C+L one used in chapter 2, but also C+L+S and C+L+S+E systems. the relationship between increasing bandwidth and the efficiency difference between the two methods will be explored. Moreover, convergence will be compared, highlighting cases where the hybrid method successfully converges in situations where the conventional method fails

3.2.1 Speed

C+L systems with uniform signal input power

In the previous section, table 3.1 showed that the hybrid method takes around 260 iterations on average to converge for a C+L system. To see how much this method improves over the conventional one a study similar to the one reported in table 3.1 is done, with reduced cases because, as preliminary results have already shown, the conventional method is computationally expensive. The *Adjustment* values considered are [1 0.7 0.4] while those of $P_{signal,in}$ are [-5 0 5 10]. The conventional bvp4c method is run three times for each case and the elapsed time is averaged to reduce the impact of measurement noise. The same is done with the hybrid method for comparison.

The results are presented in the following tables:

$P_{signal,in}$ (dBm)	**Adjustment = 1**	**Adjustment = 0.7**	**Adjustment = 0.4**
-5	0.4388	2.7875	4.9895
0	0.7254	1.1965	2.9472
5	1.5989	0.5950	1.9007
10	1.5121	0.8363	osc

Table 3.4: Average time elapsed by the hybrid method for varying signal and pump power

$P_{signal,in}$(dBm)	**Adjustment = 1**	**Adjustment = 0.7**	**Adjustment = 0.4**
-5	11.8595	13.4503	17.2416
0	18.2212	26.9112	23.5033
5	13.4306	15.9686	24.7477
10	22.5177	16.5569	25.8477

Table 3.5: Average time elapsed by the conventional bvp4c method for varying signal and pump power

The error between the 2 methods is reported below. The case where complex numbers appear are due to the conventional mehtod not converging. This will be further discussed in the convergence section 3.2.2.

$P_{signal,in}$ (dBm)	**Adjustment = 1**	**Adjustment = 0.7**	**Adjustment = 0.4**
-5	0.0884	0.0392	0.0188
0	0.0625	0.0144	-23.2393 + 13.6438i
5	0.0484	0.0386	0.8974
10	0.0472	0.0574	NA

Table 3.6: Error Matrix for Different Adjustment Factors and Signal Powers

The amount of gain in time between the 2 methods is seen to be substantial. To quantify it, the time gain is defined as:

$$TimeGain = \frac{Time_{conventional}}{Time_{Hybrid}}$$

It is computed using all the data in tables 3.4 and 3.5. The following table is obtained:

$P_{signal,in}$ **(dBm)**	**Adjustment = 1**	**Adjustment = 0.7**	**Adjustment = 0.4**
-5	27.03	4.83	3.46
0	25.12	22.49	7.97
5	8.40	26.84	13.02
10	14.89	19.80	NaN (osc)

Table 3.7: Time gain between the conventional and hybrid methods for varying signal and pump power

The gain is substantial, it seems to get lower for increasing pump power (decreasing *Adjustment* factors) and for lower signal input power. This can be explained by the fact that the hybrid method would need to scale down the pumps in cases of high pump power, loosing efficiency.

The values of table 3.7 are averaged for all the cases to obtain the average time gain achieved by the hybrid method over all C+L cases considered. That value is : 15.80

C+L systems with non uniform signal input power

The developed method's performance is now compared to the conventional one in the case of more realistic C+L systems where the signal power is not uniform but varying according to the following figure:

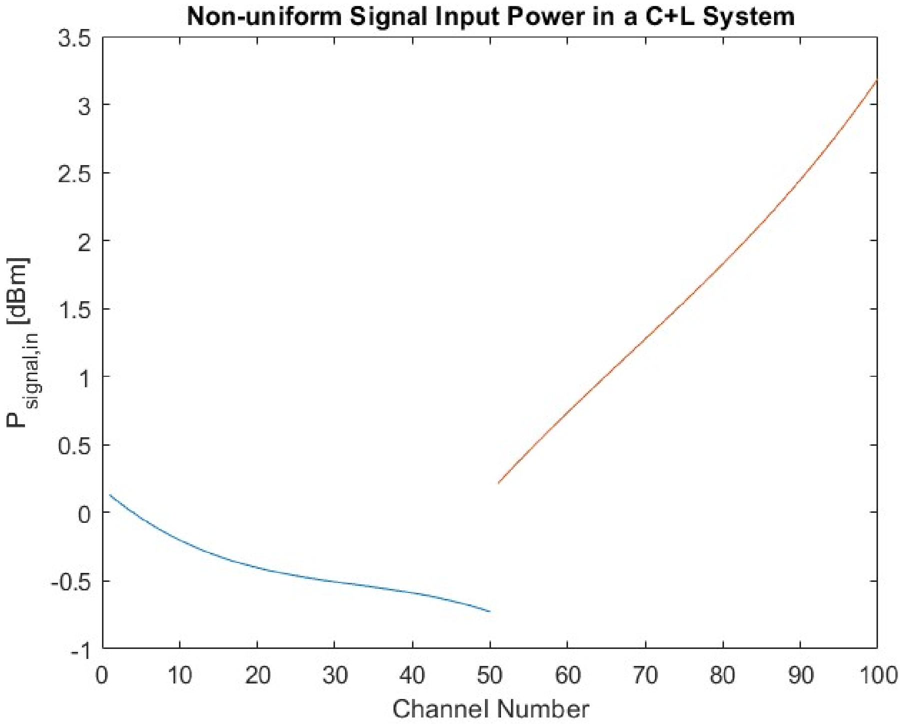

Figure 3.2: Non-uniform signal input power C+L system

The varying signal input power illustrated above is applied to the same C+L system used before. The pumps used are the same as the previous sections with $Adjustemnt = 1$. The output of both methods is shown below, along with the time elapsed averaged over five tries and the error:

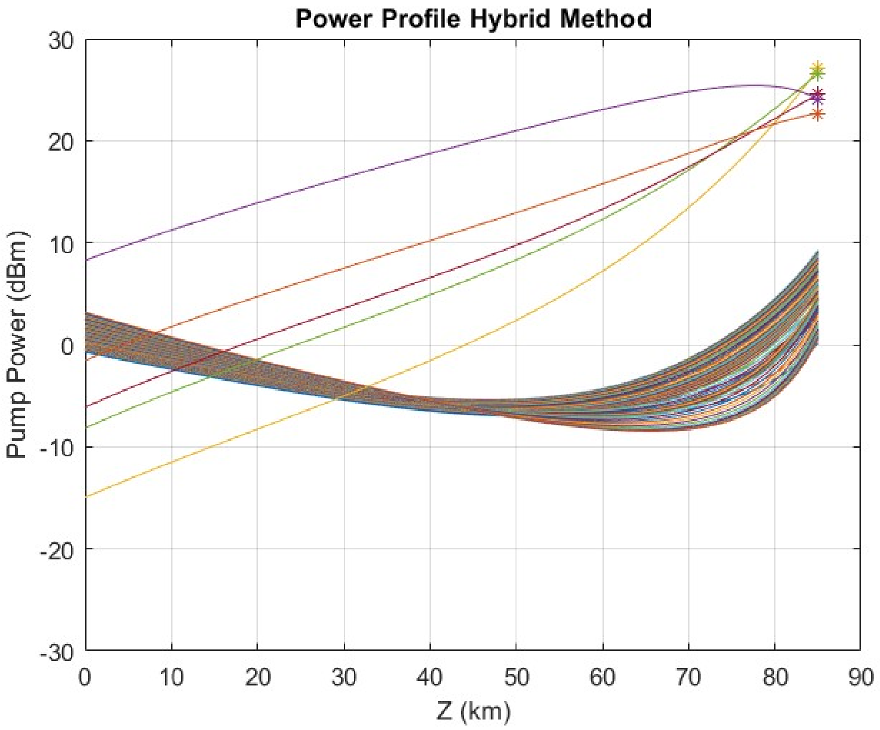

Figure 3.3: Hybrid method output for non-uniform signal input power C+L system

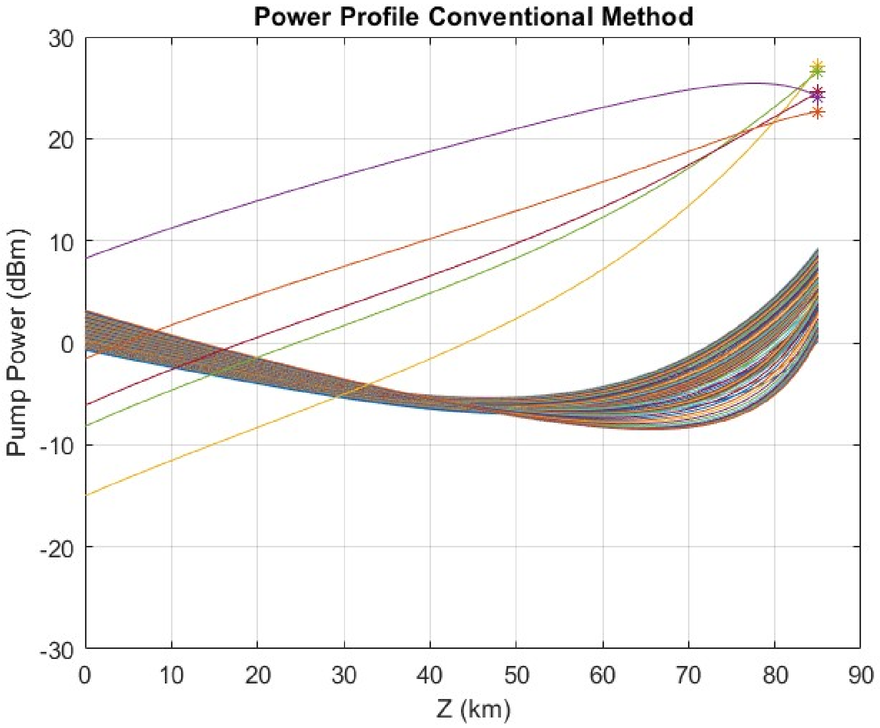

Figure 3.4: Conventional method output for non-uniform signal input power C+L system

The two figures look similar, in fact the error between the two is 0.0352 dBm, an accepted one. More importantly, the average time for convergence of the hybrid method is 0.910 seconds while the conventional one takes 23.307 seconds to converge, representing an average time gain of approximately 25.

To further test this time gain, the experiment is repeated for different scenarios of non uniform signal input power. The input powers shown in 3.2 are scaled by a factor k where it takes values in [1 2 3] while the Adjustment factor is varied between the values [1 0.7 0.4]. The average times taken by both methods after 3 runs of the 9 scenarios are reported below:

k Factor	**Adjustment = 1**	**Adjustment = 0.7**	**Adjustment = 0.4**
1	0.5873	1.3039	4.9541
2	0.7754	1.0539	2.0868
3	1.0778	1.3084	2.3356

Table 3.8: Average timing of the hybrid method for non-uniformly powered C+L systems

k Factor	**Adjustment = 1**	**Adjustment = 0.7**	**Adjustment = 0.4**
1	30.1953	46.2053	48.8202
2	35.2815	40.0672	42.4700
3	25.4631	54.1620	39.7435

Table 3.9: Average timing of the conventional method for non-uniformly powered C+L systems

The error between the 2 methods is reported below, where complex values indicate the non convergence of the conventional "bvp4c" method. This will be further studied in 3.2.2.

k Factor	**Adjustment = 1**	**Adjustment = 0.7**	**Adjustment = 0.4**
1	0.0611	0.0304	-27.9443 + 13.6438i
2	0.0620	0.0364	-27.3306 + 13.6438i
3	0.0591	0.0395	-26.3177 + 13.6438i

Table 3.10: Error Matrix for non-uniformly powered C+L systems

Over all the scenarios, the hybrid method is able to outperform the conventional one in non-uniform signal input power C+L systems with an overall average time gain of 31.40

C+L+S systems

At this stage, the performance of the method is compared for wider UWB systems, namely ones with the added S band. This band takes the range of frequency between 196 and 201 THz. Along with the C+L channels considered before, 38 channels are added in the range of the S band with a channel spacing of 125 GHz and a symbol rate of 100 GBaud. The signal channels are powered in input in a realistic non-uniform fashion as illustrated by the figure below.

Moreover the pumps are similar to the ones used before with the same power scalable by the *Adjustment* factor, however their frequencies are shifted by 5 THz to make room for the S band. The pump frequencies thus look like this: [215.56, 213.87, 211.72, 209.51, 205.55] THz.

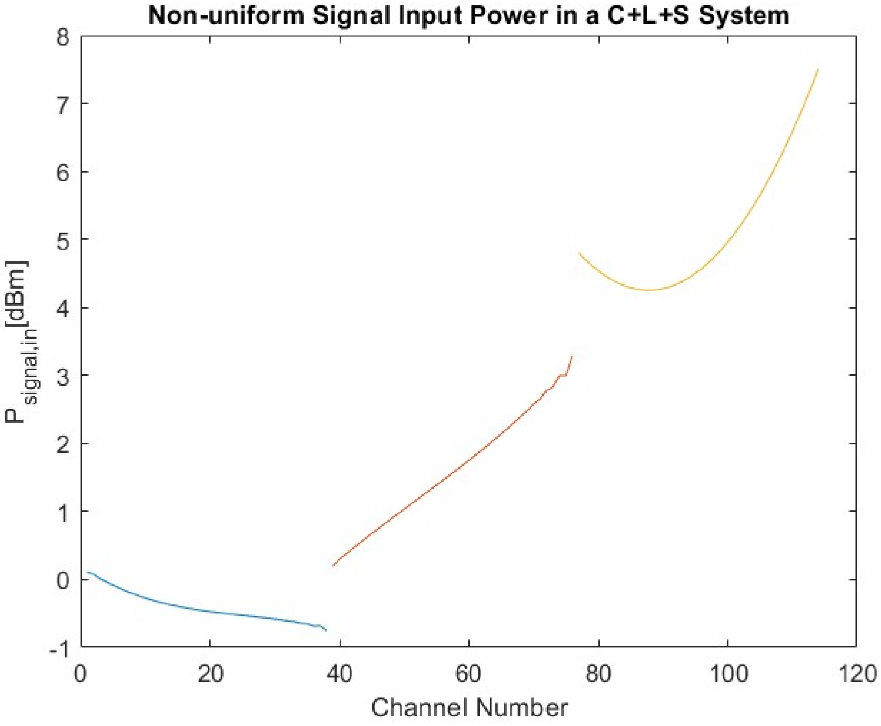

Figure 3.5: Non-uniform signal input power C+L+S system

This system's power profile throughout the span of the fibre is solved four times with different boundary conditions. Namely by varying the *Adjustment* factor between the values [1 0.5] and the k factor, introduced in 3.5, between [1 2]. The time elapsed by both methods, averaged over 3 runs is reported below, along with the error that will be discussed in the final section.

k Factor	**Adjustment = 1**	**Adjustment = 0.5**
1	0.6933	1.3664
2	0.9243	1.5358

Table 3.11: Timing of the hybrid method for non-uniformly powered C+L+S systems

k Factor	**Adjustment = 1**	**Adjustment = 0.5**
1	35.9251	52.1826
2	27.8826	48.9343

Table 3.12: Timing of the conventional method for non-uniformly powered C+L+S systems

k Factor	**Adjustment = 1**	**Adjustment = 0.5**
1	0.0603	0.4079
2	0.0482	0.0323

Table 3.13: Error Matrix for non-uniformly powered C+L+S systems

Consequently, the average time gain achieved over all runs of C+L+S systems, is 38.009

Performance In The Main Software

The original purpose of the efficient calculation of the power profile is to eventually use it to estimate the NLI noise and the GSNR over the link using the closed form EGN model, allowing for optimisation of throughput, among other things. The previous studies of this section 3.2.1, have shown the average time gain the hybrid method was achieving over the conventional one in increasingly complex scenarios. Here, an optimisation is done to get a practical estimation of the average time gain achieved by the method. The optimisation is done on a 10 span, 1000 Km long C+L+S+E system with 125 GHz channel spacing and 100 Gbaud symbol rate. The cost function is one that maximises throughput over the entirety of the link. Below is a table showing the amount of times the cost function was calculated (loops) by the CFM using both methods along with the time elapsed accordingly.

Table 3.14: Comparison of time elapsed by the optimisation using the conventional and hybrid Methods

Method	Number of Loops	Time Elapsed (s)
Reference Method	835	192764
Hybrid Method	1736	14979

A small proportionality computation can be done to compare the two times. Since each method performed a different amount of loops, the time gain is calculated as follows :

$$TimeGain = \frac{\frac{192764 \cdot 1736}{835}}{14979} = 27.8$$

.

Conclusions On The Efficiency

To conclude, the method for power profile computation developed throughout this thesis, referred to as the hybrid method, achieves substantial time gain over the bvp4c method of MATLAB in all of the possible cases. A summary of all the time gains is reported below:

Scenario	Average Time Gain
C+L Uniform	15.80
C+L Non-Uniform	31.40
C+L+S Non-Uniform	38.01
C+L+S+E Optimization	27.8

Table 3.15: Average Time Gain in Different Scenarios

It is thus concluded that the efficiency of the hybrid method increases for increasing signal input power complexity and increasing bands. This is hypothesised to be due to the fact that with increasing bands, the only thing different for the developed method is the size of the matrices in equation 2.4. Whereas other methods that try to solve the differential equations directly, would have an increasing number of coupled equations which would be increasingly less efficient to solve. This makes the hybrid method an even more important step forward towards the efficient power profile computation in the context of ultra-wide band systems

Finally, an explanation of the reason why the hybrid method seems to be so much faster than the bvp4c method is given. For solving the coupled Raman differential equations, the conventional method relies on MATLAB's bvp4c function, which addresses the boundary value problem by solving two initial value problems, one forward and one backward from opposite ends of the fiber. The method iterates by adjusting initial guesses until the solutions from both directions converge. According to the solver's output, the solution is reached after 148068 calls to the ODE function and 494 calls to the boundary condition function.

Each iteration of the conventional method involves complex numerical calculations, which accumulate over a large number of iterations. In contrast, the hybrid method developed in this work achieves the same results through efficient matrix multiplications and requires only a few hundred iterations on average. This significant efficiency difference arises from the hybrid method's ability to reduce the computational complexity per iteration, and number of iterations, thus offering substantial time savings while maintaining accuracy.

3.2.2 Convergence

The efficiency study done in section 3.2.1 has already shown that both methods converge for most of the systems presented. The error tables 3.6, 3.10 and 3.13 showed the maximum error between the power profiles computed by each of the methods to be negligible in most cases indicating that both are accurate enough. However, in some cases the error is a complex number. This indicates that one of the methods did not converge.

Consequently, to compare the two methods in terms convergence, an example is taken where the error is complex.

A case from table 3.10 is taken where $k = 2$ and $Adjustment = 0.4$. The error was shown to be complex with high magnitude. In fact the power profile in output of the conventional method is the following:

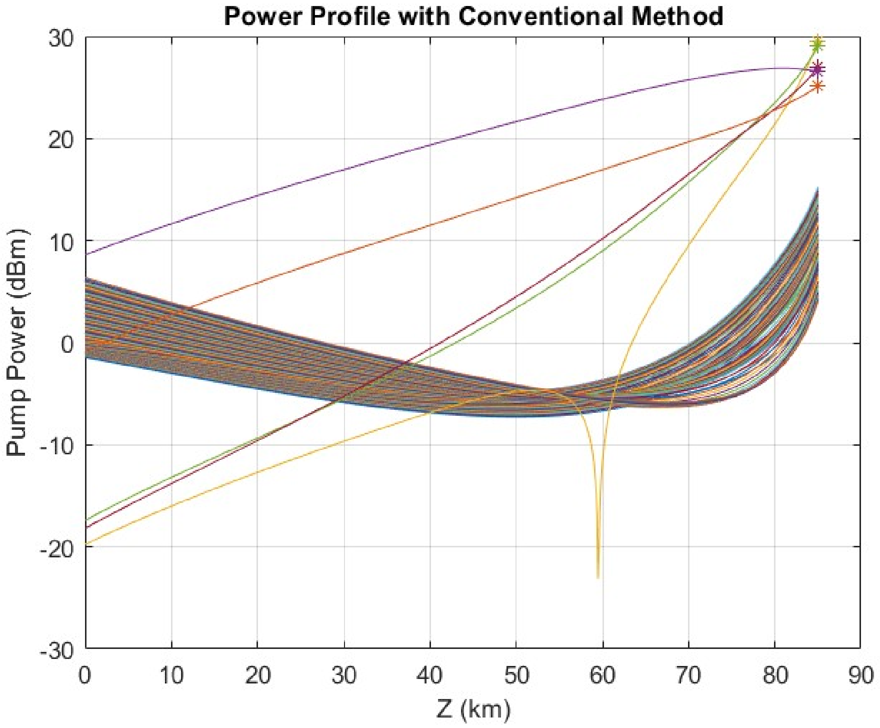

Figure 3.6: Non-uniform signal input power C+L system with conventional method

While the solution provided by the hybrid method is reported below:

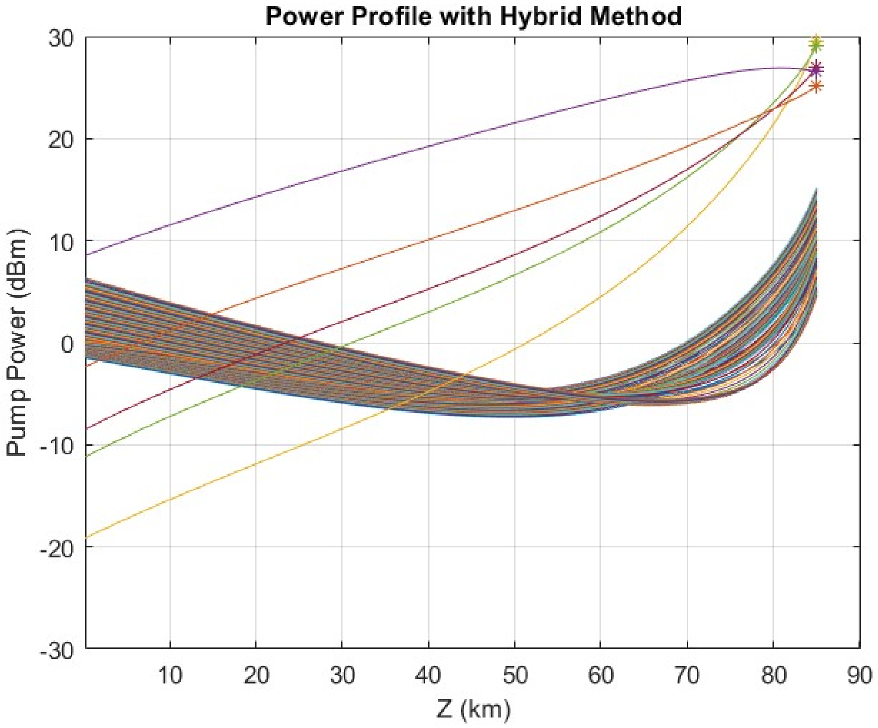

Figure 3.7: Non-uniform signal input power C+L system with hybrid method

It is immediately seen that the output of the conventional method is erroneous, because of the dip in the power of one of the pumps. In fact this pump has a complex profile inside the matrix P. This in turn affects the signals and the whole reliability of this method. On the other hand the hybrid method provided a real solution.

As seen in the different error tables, this scenario happens multiple times. In all of which the hybrid method is able to provide real solutions while the complex values arise in "bvp4c".
It is concluded that the hybrid method not only outperforms bvp4c in efficiency but is also applicable to a wider range of systems.

Through extensive testing, it has been demonstrated that the method developed in this work is significantly more efficient than the conventional approach previously used. On average, it achieves a time gain factor of around 30, depending on the specific scenario. Additionally, the hybrid method has shown robust convergence across a wide variety of cases, even in situations where the conventional method fails. This makes the developed approach superior in all critical aspects, particularly in terms of efficiency and reliability.

Conclusion

In this thesis, advanced technologies for modeling and optimizing Ultra-Wide Band long-haul Raman-amplified coherent optical transmission systems were studied, with the aim of facilitating future commercial implementation. Such systems are critical for achieving higher throughput in optical links, addressing the increasing demands of users and the complexity of modern applications. Specifically, the thesis explored the Enhanced Gaussian Noise (EGN) model, a closed-form physical layer model that efficiently estimates non-linear noise on optical links given the power profile of signals. However, calculating this power profile, governed by non-linear coupled differential equations, is computationally demanding. The primary objective was the development of a more efficient, reliable, and adaptive method for power profile calculation, leading to significant speedup of the closed-form model, thus paving the way for real-time application in commercial systems.

During the course of this thesis, a detailed study of the coupled Raman differential equations governing the variation of signal and pump powers across the fibers was conducted. A new approach was taken in solving these equation that led to the development of two novel methods: Dynamic Pump Calibration (DPC) and Progressive Signal Injection (PSI). Both methods were based on the integral form of the equations but followed different strategies to ensure convergence to a solution. While both methods performed well, each faced specific limitations. DPC struggled with highly dynamic parameters that were difficult to adapt to every scenario, whereas PSI encountered accuracy issues, lacking control over precision metrics. The core contribution of this thesis was the development of a hybrid method that combined these two approaches, transitioning from PSI to DPC in a serial manner, where PSI's final solution is passed to DPC as an initial guess. This integration allowed the hybrid method to retain the strengths of both while overcoming their respective shortcomings. Through extensive testing, the hybrid method demonstrated robustness by achieving convergence in all practical scenarios, even in cases where conventional methods failed. Moreover, it proved to be significantly more efficient with no loss in accuracy, delivering an average time factor improvement of 30 compared to the existing method. Furthermore, it has been shown that the efficiency of the developed method increases with

the complexity and the bandwidth of the system. This advancement enabled optimizations on 1000 km multi-band systems to be performed in a fraction of the time required by conventional approaches. Such improvements will make it possible to conduct complex optimizations on increasingly larger UWB systems much faster, accelerating research in this field and driving the commercial viability of these technologies.
Although the developed method has proven to be robust, there is still room for improvement in future research. Enhancements could focus on better adaptability of the parameters across a wider range of scenarios through the use of more advanced algorithms. Additionally, optimizing these parameters could further improve both efficiency and robustness. Finally, further testing should be conducted to evaluate the method's performance in systems with co-propagating pumps.

Ultimately, the method developed in this thesis paves the way for more efficient resolutions of Raman differential equations. This advancement in computational efficiency will significantly contribute to the research and deployment of UWB systems, enabling complex optimizations across multiple bands, ones that were previously hardly feasible due to the extended simulation times required. Finally, beyond the field of optical communications, it is intriguing to wonder whether these methods might find broader applications in the efficient resolution of general coupled differential equations, potentially unlocking new possibilities in other areas of science and engineering.

Bibliography

[1] Y. Jiang, J. Sarkis, A. Nespola, F. Forghieri, S. Piciaccia, A. Tanzi, M. Ranjbar Zefreh, and P. Poggiolini. «Performance Enhancement of Long-Haul C+L+S Systems by means of CFM-Assisted Optimization». In: *Journal of Lightwave Technology* 42.10 (May 2024), pp. 2331–2339 (cit. on pp. 1, 7).

[2] P. Poggiolini, G. Bosco, A. Carena, V. Curri, Y. Jiang, and F. Forghieri. «A Detailed Analytical Derivation of the GN Model of Non-Linear Interference in Coherent Optical Transmission Systems». In: *IEEE Photonics Technology Letters* 23.11 (June 2011), pp. 742–744 (cit. on pp. 1, 9).

[3] A. Carena, G. Bosco, V. Curri, Y. Jiang, P. Poggiolini, and F. Forghieri. «EGN Model of Non-Linear Fiber Propagation». In: *Optical Society of America* 060.1660 (2014), Coherent communications, Nonlinear optics, fibers (cit. on pp. 1, 8).

[4] M. Ranjbar Zefreh and P. Poggiolini. «A real-time closed-form model for nonlinearity modeling in ultra-wide-band optical fiber links accounting for inter-channel stimulated Raman scattering and co-propagating Raman amplification». In: 20 (Nov. 1999), pp. 569–571 (cit. on pp. 2, 8).

[5] M. Ranjbar Zefreh, F. Forghieri, S. Piciaccia, and P. Poggiolini. «Accurate closed-form real-time EGN model formula leveraging machine learning over 8500 thoroughly randomized full C-band systems». In: 20 (Nov. 1999), pp. 569–571 (cit. on p. 2).

[6] Y. Jiang et al. «Experimental Test of Closed-Form EGN Model over C+L Bands». In: *OFC* (2024) (cit. on p. 2).

[7] G. S. Hurst. *Spectroscopy*. Encyclopedia Britannica. Accessed: Sep. 02, 2024. URL: https://www.britannica.com/science/spectroscopy (cit. on p. 6).

[8] Pierluigi Poggiolini. «A generalized GN-model closed-form formula». In: *arXiv preprint arXiv:1810.06545* (Sept. 2018). URL: https://arxiv.org/abs/1810.06545v1 (cit. on p. 9).

[9] L. Kwon Choi, P. Kim, Ja. Park, J. Park, and N. Park. «Adiabatic, closed-form approach to the highly efficient analysis of a fiber Raman amplifier problem». In: *OPTICS LETTERS* 30 (Jan. 2005) (cit. on pp. 15, 16).