Fibres Semester 2

Timothy Peters

March 2021

1 Introduction

This report is based on strand two of the Multicore Fibres project. The central theme of this strand was "optimal packing for multicore fibres".

1.1 Motivation

Optical fibres have been a source of study since the 1800's, but even more so in the current time period. With information and data being essential new resources, having the infrastructure to transfer this information on a microscopic and macroscopic scale has only become more important.

This has led to an interest in multicore fibres (MCF). Used in medicine, for example in endoscopic imaging, these MCFs have 1000s (or even millions for endoscopy) of cores packed closely together. A larger number of cores has led to an increase in information per area but it also has drawbacks such as crosstalk between cores. As a signal travels down the fibre, power transfer occurs between cores leading to noisy data and a loss of information. Thus, it is no wonder that being able to extract a clean, uncorrupted image is hugely important in most applications.

In this strand we seek to examine a MCF over a short length scale and examine how we can create tiling patterns that reduce this effect and how we can minimise the loss between cores over the minimum area. We use the insight gathered from strand one regarding coupling affects between identical and similar tiles based on distance. From this foundation we proceed in order to create an optimal tiling pattern which ideally results in minimal amount of loss of power in the cores.

1.2 Structure of Report

The rest of the report is structured as follows:

In Section 2 we initially give an overview of our system of equations. We define key concepts such as *loss*, *tile* and *sub-tile* thus being essential for understanding later sections.

In Section 3 we look to *optimally design fibres* by optimising over the propagation constants β which fill up our sub-tile. We explore this using different inbuilt Matlab functions.

In Section 4 we firstly examine *robustness* of solutions in the presence of noise. We then proceed to study the robustness with regards to the length of the fibre and we look to minimise the expectation value of the power loss.

In Section 5 we explore the same questions introduced in Section 4 but now we now look at the effect of *distance between cores* on these problems.

Finally we provide our final outlook in Section 6.

2 Our System

2.1 Set up of system

We consider one of the core arrangements shown in Figure 1. Each core in the fibre has it's own refractive index and propagation constant β which determines how light propagates down the fibre. In addition, as light propagates down a core, there is a coupling effect with neighbouring cores called CrossTalk (XT), where the strength is determined by the distance between the cores.

This gives us our system of N ODEs [2]:

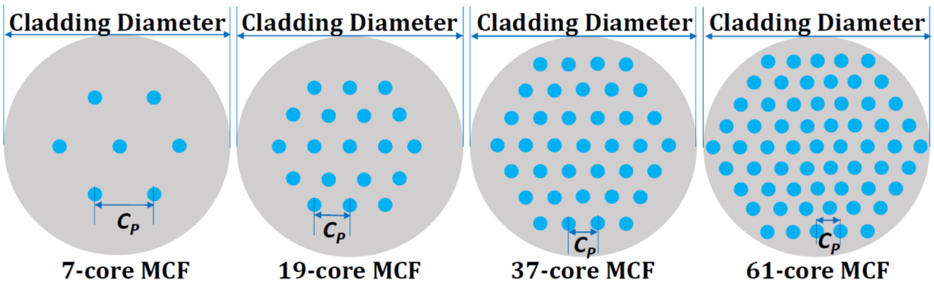

Figure 1: Types of multicore fibres [1], where C_p relates the distance between the neighbouring cores. We denotes this as d for the rest of this report.

$$u(z)' = -iAu(z) \qquad u(0) = u_0 \in \mathbb{C}^N \tag{1}$$

where $A \in \mathbb{R}^{N\times N}$ is the assembled coupling matrix with the propagation constants β_i along the diagonal. The matrix $A = A^T$ is symmetric and as such the off diagonal components are the coupling coefficient $A_{ij} = A_{ji} = \kappa_{ij}$ which are non zero if cores i and j are close enough; z corresponds to length along the fibre; $u(z) \in \mathbb{C}^N$ is the vector showing the light in each core at length z.

The solution to Equation 1 is given by :

$$u(z) = \exp(-izA)u(0), \tag{2}$$

where we have used the expm function in Matlab to calculate this.

2.2 "Tile and sub-tile"

As mentioned in Section 1.1, one of the main objectives of this strand was to explore optimal ways of packing cores into a MCF. We wanted to test different tiling configurations, looking at both the macro-scale and micro-scale, and gained many insights:

1 Some repetitive patterns consisting of $M < 20$ cores in a 19-MCF performed significantly better than others.
This repetitive pattern or *sub-tile* was tested for $M = 3, 4, 5, 6, 7, 9$ different cores. Here there is a trade off between wanting the number of distinct cores to be as small as possible while yielding good results. This was tested in two distinct ways:

  i Starting with core 1 in the center, calculate the final intensity in the core where power was initialised. Then shift the entire pattern so that core 2 was in the center while maintaining the structure and compute. Proceed for each core $i \in (1, \ldots M)$. Initial power is always in the center core, just the center core is permutated.

  ii Without shifting the structure, we initialise power to be in each core $j \in (1, \ldots, N)$ for the N-MCF and compute the final intensity in the j^{th} core.

From both of these cases, we examined the loss as given by equation 6 or 7. The best result came from the *9 core sub-tile.*

2 The same sub-tiles performed much worse in a 37-MCF.
Due to the increased number of identical cores in the MCF, the overall loss $L_1(\beta)$ and $L_2(\beta)$ for the 37-MCF increased disproportionately over the 19-MCF.

3 Small increases to distance between cores can cause large reductions in loss.
As the distance d between two neighbouring cores increased from $5 \to 6\mu m$ there was a significant reduction in loss, as shown in Figure 2-left. This raised the question of trade off between *minimal loss vs area.* In addition, as the distance between cores increases, we can see from Figure 2-right, the clear emergence of a "3-tier" structure which leads to a possible random tile generation strategy. We discuss the effects of increased distance between cores in our Loss function in Section 5.

4 By adding an extra layer of cladding between each N-MCF the structure allows more local optimal designs, instead of sub-optimal global ones.
From 1 and 2, a clear distinction in power retention between the different 19 and 37 MCF was seen. In 3, the loss could be reduced from increasing the overall distance between cores. Examining the question

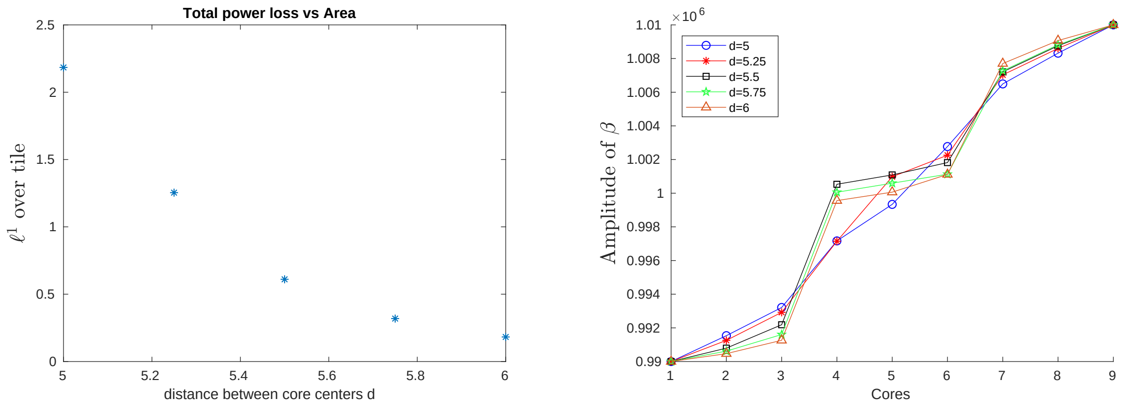

Figure 2: Left: Loss over the 19-cores as distance between cores increases; Right: Set of βs for the different distances. Both using fmincon.

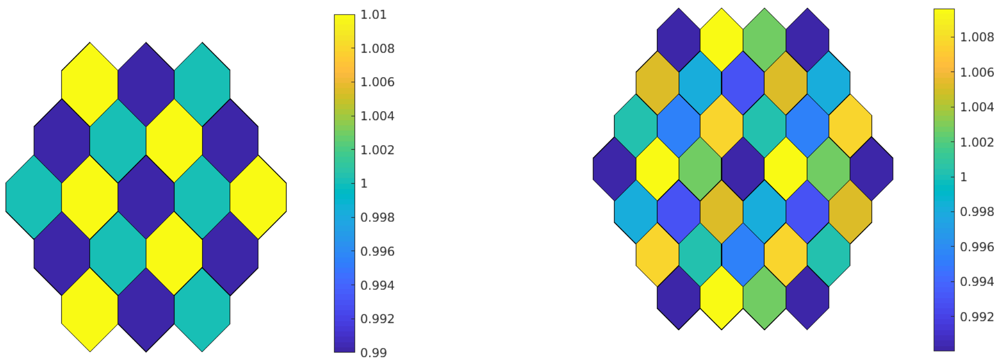

Figure 3: Left: 3 Sub-tile filling out a 19-tile $(N, M) = (19, 3)$; Right: 9 Sub-tile filling out a 37-tile $(N, M) = (37, 9)$. Also shows the structure of the relative placement for the β coefficients.

raised in 3, we proceed to look at the 19-MCF under the assumption that when constructing the full 10000+ MCF there will be separation or cladding between adjacent 19-MCFs. These 19-MCF are what will be referred to as the *tile*. Given this set up, we look at the loss as generated by (1ii).

From considering the above insights, this allowed us to narrow our search. Thus moving forward, we refer to the tile as a 19-MCF ($N = 19$) and the sub-tile is a repetitive pattern of 9 cores ($M = 9$) which fills up the tile as these preformed best in the experiments. Examples are shown in Figure 3.

2.3 Problem Formulation

Our objective was to find an optimal tiling pattern and propagation constants which resulted in the best power retention. There are two cases which we considered:

- Optimise over a specified length of fibre,
- Optimise over the whole fibre.

In the first case, we only care about a given length of the fibre, T and want to make sure it is robust over slight discrepancies - $T + \Delta T$.

In the second case, we want to minimise power transfer over the full length of the cable, so if we cut a random length of fibre, it will have minimum transfer as defined by our "loss function".

2.3.1 Specified Length

Here we consider a cable of length T. Taking a tile of size N, we would like to determine optimal values of the propagation constants, $\beta = (\beta_1, \beta_2, \dots, \beta_M)$ so that we have minimal loss in the k^{th} core when we initialise power in the k^{th} core. Here M is the size of the sub-tile we choose. We rewrite equation 2 as

$$\underbrace{u_k(\beta)}_{\in\mathbb{C}^N} = \underbrace{\exp(iTA(\beta))}_{\in C^{N\times N}} \underbrace{\hat{e}_k}_{\in\mathbb{R}^N}, \tag{3}$$

where we initialise the system so that all the power starts in the k^{th} core. We now want to extract the power output from the k^{th} core after the system evolves so we look at the intensity given by

$$I_k(\beta) = |u_k(\beta) \cdot \hat{e}_k|^2 \in [0, 1]. \tag{4}$$

We now want to minimise the loss over the entire tile. Examining this for each core in the tile:

$$I(\beta) = (I_1(\beta), I_2(\beta), \ldots, I_N(\beta))^T. \tag{5}$$

Now we define our loss function. In this course of our work we have considered the two different loss functions:

$$L_1(\boldsymbol{\beta}) = \sum_k (1 - I_k(\beta)) = \|\mathbf{1} - I(\beta)\|_1, \tag{6}$$

and

$$L_2(\boldsymbol{\beta}) = \sum_k (1 - I_k(\beta))^2 = \|\mathbf{1} - I(\beta)\|_2^2, \tag{7}$$

where $\mathbf{1} = (1, 1, 1, \ldots 1)^T \in \mathbb{R}^N$. To understand why we write our loss functions in this manner, we first look at $\mathbf{1} - I(\beta) \in R^N$. If we look at the k^{th} row in this vector, this corresponds to the difference between initial and final output in the k^{th} core when power is initiated in the k^{th} core. Thus, this column vector takes all the differences between initial and final power in a core when power is initiated in that core.

Moving to our loss functions, we then have the ℓ^1 loss given in Equation 6, whereas Equation 7 gives the ℓ^2 loss. We know that by minimising in the ℓ^1 norm, we end up minimising the sum of absolute differences. By using this norm we just measure the distance between initial and final power and make sure that over all the cores, this is minimised. The issue with this method, is that it doesn't penalise outliers. Thus a solution in which a few cores have poor final output would not be unexpected. Since one of our main concerns is to have good power retention over each core, this creates a problem which the ℓ^2 loss proves quite effective at fixing. By using this loss we avoid large deviation in single cores as this is punished more harshly. Thus, for this function we can expect less cores with low power retention and the majority of cores will have slight power loss.

2.3.2 Unspecified length

Here we follow the pattern of above, however we now consider over the entire length of the cable with length T. Our output vector now depends on both the β values and the output at each point in the fibre. We write this as:

$$u_k(t, \beta) = \exp(itA(\beta))\hat{e}_k, \tag{8}$$

for $t \in (0, T]$ Similarly,

$$I_k(t, \beta) = |u_k(t, \beta)|^2 \in [0, 1]. \tag{9}$$

In Section 4.4, we will consider the case where $t \in \mathcal{U}(0, T]$ and we look for the expectation value over an interval. Finally, our form of the loss functions is identical to Equations 6 and 7.

3 Optimal Pattern

3.1 Propagation constants β

From experimental data the propagation constants were estimated to be of order 10^6. These values are dependent on the refractive index of the material being used. There is also a practical limit in the materials with acceptable refractive index so we wish to limit our possible propagation constants to within 1% error.

Let

$$\Omega = [0.99, 1.01] \times 10^6,$$

be the set of acceptable values of βs. As we aim to maximise power retention within the cores, we take special note of the β parameters which achieve this. Our thought was that by examining the structure of above parameters, this will help highlight the form that efficient tiling patterns will take. An example of this is shown in Figure 2-(right), where we can see a three tier structure emerging for the β coefficients as the distance increases.

3.2 Optimal set of βs

We now search for the sub-tile that yields the minimum loss when filling in the tile:

$$\beta^* = \arg\min_{\beta \in \Omega^M} \mathbf{L}_j(\beta), \tag{10}$$

with associated loss:

$$L^* = \mathbf{L}_j(\beta^*), \tag{11}$$

for $j = 1, 2$.

This was explored via 3 methods in the following order:

- Direct Search,
- fmincon,
- Patternsearch (Direct Search version 2),

where fmincon and patternsearch are built in Matlab optimisation functions.

Direct search was initially used to try find the optimal set of propagation constants in the method given by Section 2.2 (1i) with the unspecified length given by Section 2.3.2. Although with a slow rate of convergence to a local minimum, the approach is simple to implement and one of the main advantages is there is no need to compute the gradient of our loss function. In our case, we chose a basic algorithm (Algorithm 1) based on the following idea:

1 Generate initial condition.

2 Perturb one of the parameter values.

3 Check if $f(x) > f(x_{new})$.

4 If perturbed value performs better, keep it.

5 Otherwise, go back to the old value.

There are various methods to improve the speed and performance of this basic algorithm as laid out in [3]. For example, adapting the perturbation factor λ by decreasing it in case of rejection, or increasing it in case of acceptance.

Algorithm 1: Direct Search Algorithm

Result: Some set of optimal βs
initialization;
$\beta^0 \in \Omega$ be given
λ perturbation factor be given
$i \leftarrow 1$
while *Enough points not generated* **do**
  Generate $j \in (1, \dots M)$
  $r \sim \mathcal{N}(0, 1)$
  $\beta^{i^*} = (\beta_1, \dots, \beta_j + \lambda r, \dots, \beta_M)$
  if $L(\beta^{i*}) < L(\beta^{i-1})$ **then**
    $\beta^i \leftarrow \beta^{i*}$
  else
    $\beta^i \leftarrow \beta^{i-1}$;
  end
  $i \leftarrow$i + 1
end

As can be seen in Figure 4, the simple implementation of Direct Search's results are nearly identical to those obtained with fmincon. Given the impossibility of producing cores with propagation constants as exactly calculated, the difference between the two is negligible.

When compared with PatternSearch, both methods failed to find an optimal solution when examining only the endpoint of the fibre. However, they did find an optimised solution for the unspecified length problem. We

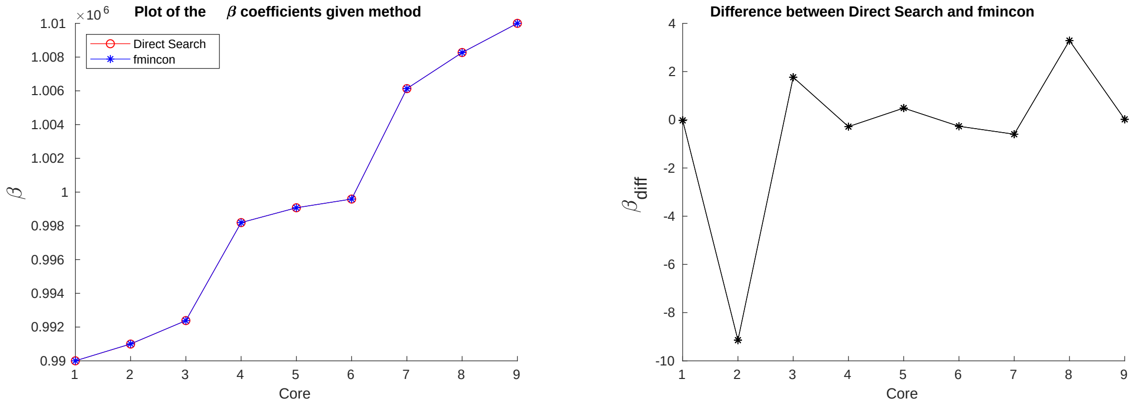

Figure 4: Left: β values calculated via direct search and fmincon. Difference between the two is almost negligible and is shown on the Right: Difference in values between the two methods.

Optimal loss in different tile sizes for different methods			
Method	Tile size	$\mathbf{L}^*_1$	$\mathbf{L}^*_2$
fmincon/Direct Search	19	1.6630	0.1913
	37	10.5514	5.9343
PatternSearch	19	1.3663	0.1159
	37	5.4961	1.1814

Table 1: Loss for given tile size and method used.

show a comparison between fmincon and PatternSearch – a Direct search Matlab function – in ℓ^1 and ℓ^2 for a 9 sub-tile in a 19-tile and 37-tile in Table 1.

PatternSearch found more optimal solutions than fmincon in both tile sizes for each type of loss function in the case of specific length. We also observe the huge increase in loss in working with a 37-tile rather than a 19-tile. For the rest of this section we will consider the **specified length** problem with the loss function given by Equation 10 along with the tile and sub-tile from Section 2.2-(4) using the PatternSearch toolbox. Later in Section 4 we will consider the different questions that arise; such as robustness for given length, unspecified length, etc.

Initially we find "good" solutions for fixed length of cable with no noise in the propagation constants. We take the interval between $0.195 \leq T \leq 0.205$ and calculate the solution at fixed points. The β set for each length are shown in Figure 5–(left). The loss versus length is shown in Figure 5–(right). An example of light travelling through the fibre for a randomly generated tile is shown in Figure 6 compared with an optimised one in Figure 7. The difference in power transfer between cores is blatant. This really highlights the need to come up with a tile of cores which reduces this effect and the problem with total random generation. If cores are too similar then power transfer will occur. One of the interesting things to notes is the difference in the distribution in the β set when considering a finite length versus an unspecified length. In the finite length – Figure 5 – we can see they take a more linear trend in the set Ω whereas for the unspecified length – Figure 4 – they form more clusters – 3 sets of 3.

4 Robustness

4.1 Noise

In an ideal setting, optical fibres would be constructed with the exact properties of what we are simulating. However, that is an unreasonable assumption. In the manufacturing process, there will be defects in constructing the cores, there will be slight deviation in the glass used, discrepancies in length, etc. Therefore, we want to be able to ensure that the system is robust. Instead of examining the propagation constants and assuming they are exact, now we assume they are normally distributed about a mean $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_M)$ given by,

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I_N).$$

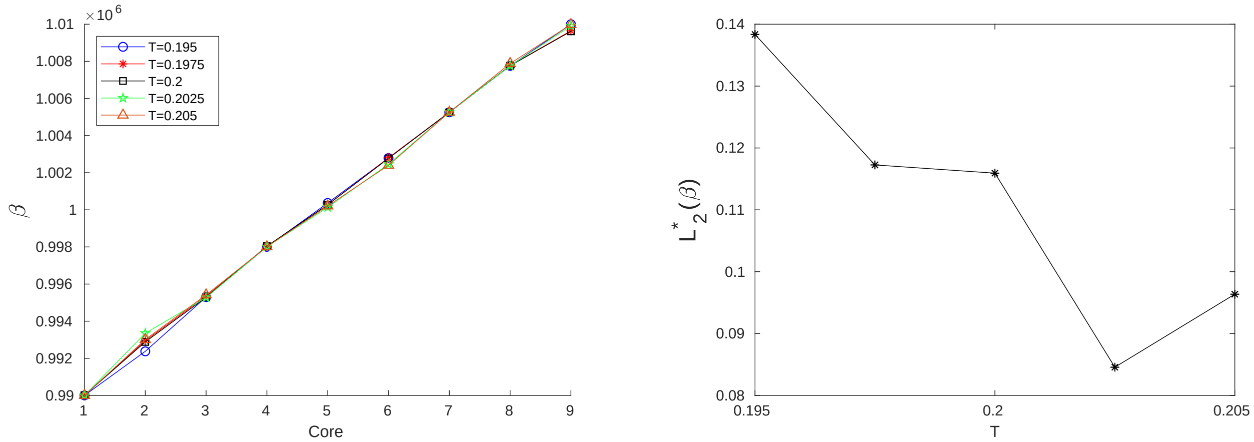

Figure 5: Left: Set of optimised β for given length using PatternSearch. Right: $L^*(\beta)$ for specified length.

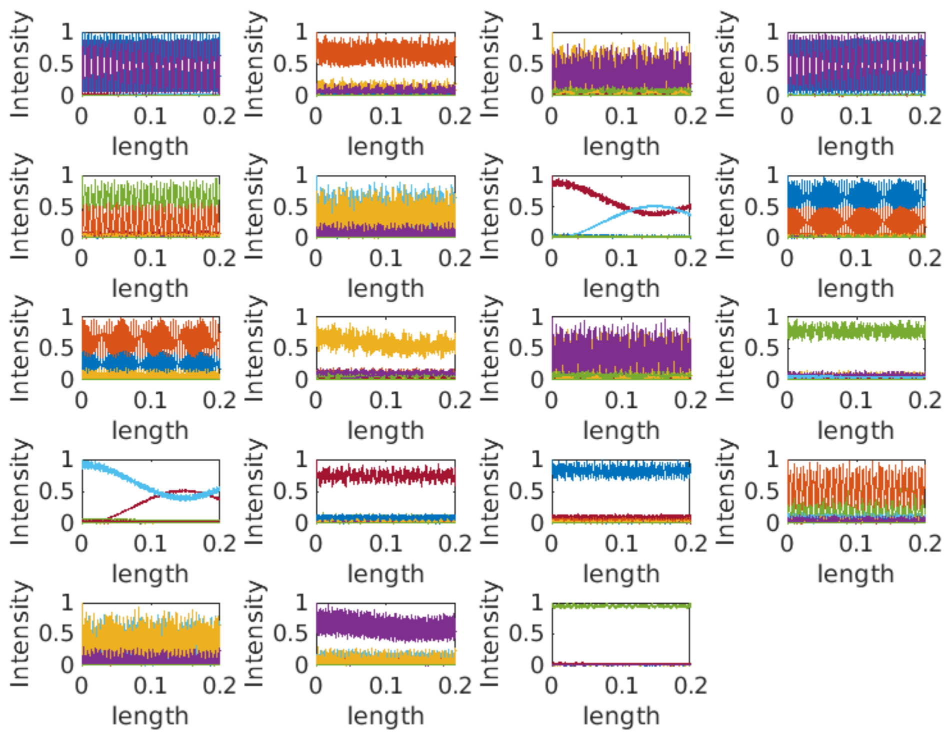

Figure 6: An example of light travelling through each core in a 19-tile for a randomly generated subtile. We can see the effects of power transfer between cores and the problem this creates when we wish to have good power retention.

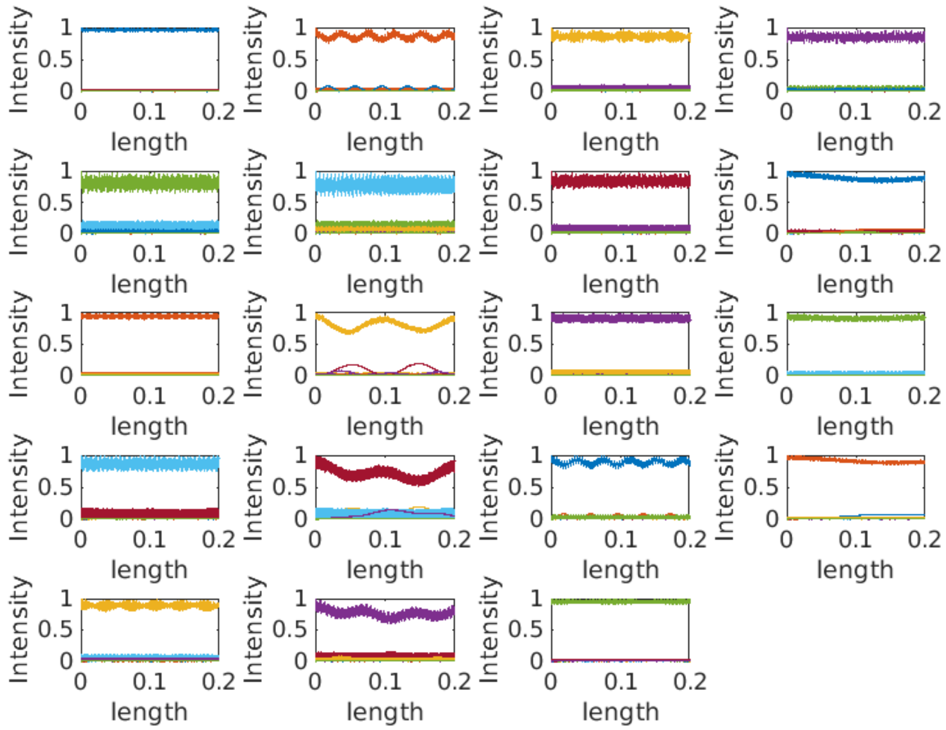

Figure 7: Signal travelling through each core for optimised tile for a given length. For column 2 especially, we observe power transfer in the middle length of the fibre but nearing full power for the final output. By looking at a specific length we don't care what happens throughout the fibre as long as final output is good.

Although we can't have precision over exact values for length, glass, etc, in practise we should have some control over the mean. However, we know that there will be some variance in the quantities we are interested in, but we may be not have control over this. This then leads us to optimise the mean value for different values of noise given by σ. Thus, for fixed $\mu \in \mathbb{R}^N$, as we increase the variance and add noise, we denote the expected loss as:

$$J(\sigma) = \mathop{\mathbb{E}}_{\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I_N)} [\boldsymbol{L}_j(\boldsymbol{\beta})] \tag{12}$$

Here, for a fixed value of σ – i.e a given level of noise – and μ we are able to determine what kind of power retention we should expect. What we are searching for is to find a good mean value so that when examined with different noise levels, it proves robust. What we do is break this into two components given by:

1 Fix σ and optimise μ.
For this first step, we firstly want to find an optimal solution under some σ, where in the previous section we did this for $\sigma = 0$. This gives us insight into the power retention and power loss which we should expect in a more idealistic setting.

2 Test σ over an optimised μ.
In this second step, we are checking whether our solution is stable. As mentioned earlier, while we may have control over our mean values, we don't necessarily have control in the level of variance. Thus, we need to check whether our mean value we compute still performs over different levels of noise.

By looking at both steps, we first find an optimal solution over trivial conditions, then we see how it performs over stress from the system.

4.2 Fix σ and optimise μ

The search for a solution where perturbations in one of the system variables – length of fibre, propagation constant, etc – has a very limited impact on the expected power loss is what motivates this question. There are a variety of ways in which we can constrain the system. We consider two ways, one probabilistic and one which penalises the variance. In this second case, we want to ensure that outliers in power loss are punished. We want to ensure our expected power loss is in a narrow interval. In terms of application, if one set of cores has perfect power retention but another set has 70% power retention, this is less useful – but with the same expected value – as two sets with 84% and 86%. We want to minimise large deviations from the expected value when adjusting system variables.

4.2.1 Probabilistic constraint

There realistically will be noise in the manufacturing process with glass, defects, etc. The constraint applied here is one that gives a high probability that the loss is small.

$$\mu^* = \arg\min_{\mu \in \Omega} \mathbf{L}_j(\mu)$$

under constraint:

$$\mathbb{P}(\boldsymbol{L}_j(\boldsymbol{\beta}) \leq S) \geq T, \qquad \boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I_N)$$

In other words,

$$\Omega = \{\boldsymbol{\mu} \in \mathbb{R}^N : \mathbb{P}(\boldsymbol{L}_j(\boldsymbol{\beta}) \leq S) \geq T, \quad \boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I_N)\} \cap \Omega_0$$

where

$$\Omega_0 = [0.99, 1.01] \times 10^6.$$

for $S \in [0, N]$ the threshold loss being accepted and $T \in [0, 1]$ the probability over which it will be accepted. This is comprised of two elements. Firstly we want to minimise our mean values similarly to Equation 10. Our constraint then demands that our noisy propagation constants have a high probability of being below a threshold value. One of the primary issues with this method is that it requires knowledge of the system which we might not have and we may not even have control over it. To be able to implement this, we need to know what kind of losses we can expect from our system under fluctuation. Thus, in practise, implementation is impractical. This lead to the second method of the penalising the variance.

4.2.2 Variance penalty

In this case, we want to minimise the expected value for fixed length but adding a penalty for large variance. We search for:

$$\mu^* = \arg\min_{\mu \in \Omega} \left(\mathop{\mathbb{E}}_{\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I_N)} [\mathbf{L}_j(\beta)] + \lambda \mathbb{E}[(\mathbf{L}_j(\beta) - \mathbb{E}[\mathbf{L}_j(\beta)])^2] \right)$$

This is comprised of two components. The first part is only checking what the expectation value of power output in our cores will be for a given noise. The second component looks at the spread of the computed outputs for a given noise and penalises extreme deviation. Thus by adding the two components together, we look for a solution that not only performs well, but is not subject to extremes.

This corresponds to one of the big questions of how stable a configuration is with regard to propagation coefficients. Thus we search for good solutions where there is noise and check if this makes optimal solutions unstable. In our test case of $T = 0.2$, we observe that our solution found in the previous section with no noise, is still a good solution. In fact, as is shown in the Table 2, for small values of σ it is possible to achieve a better expectation value since the β parameters may be outside the given interval due to noise.

4.3 Fix μ and test how it performs under varied σ

In this case, we assume that μ has already been optimised under a fixed σ_0 as above and we are testing the solution's robustness as we change the noise levels given by σ.

We know from strand one, that if the propagation constants β are too close in value this leads to power transfer between cores. Initially our cores have a set mean value for each core which is distinct enough to have low power transfer and consequently high power retention. As noise increases and we vary these βs, the more likely cores are to become close enough for this to become a problem and thus the system is more likely to break down. In the situation where the system breaks down, we can expect more power transfer between cores and

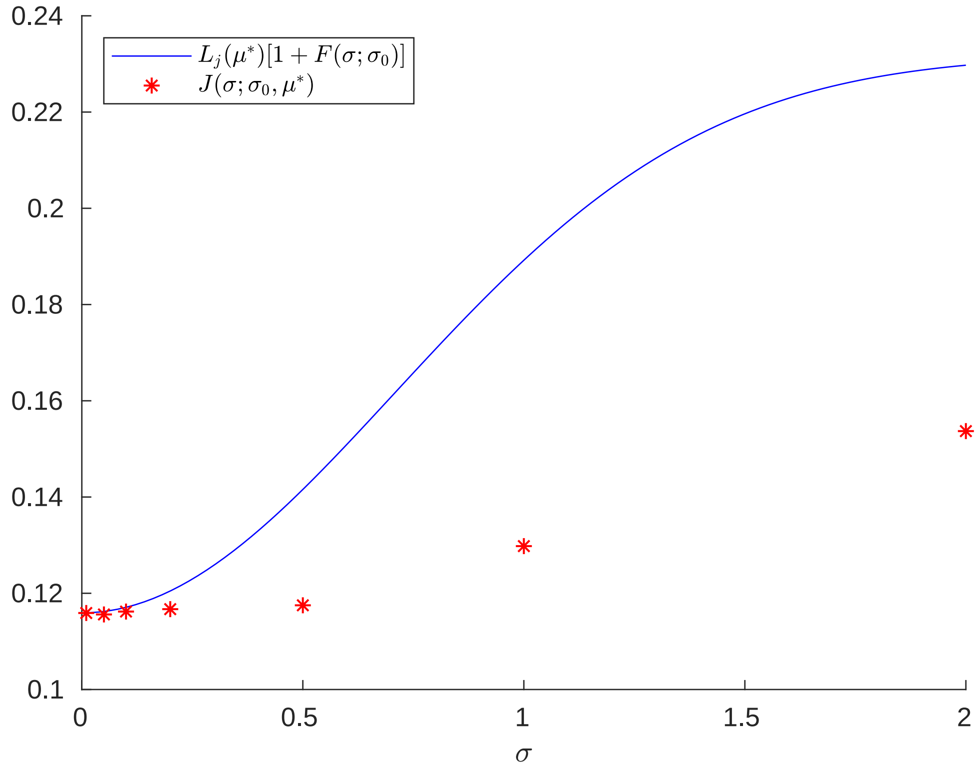

Figure 8: A test example of the inequality given by equation 13.

thus our final output will have more power loss which is precisely what we want to avoid! For testing we would want ideally want some bound to hold which limits the difference in expectation value between the noise level under which μ^* was initially optimised – σ_0 – and the new noise level σ which we are testing it. We would want something like the following equality to hold as we test different values of σ

$$J(\sigma;\sigma_0,\mu^*) \leq \mathbf{L}_j(\mu^*)[1+F(\sigma;\sigma_0)] \tag{13}$$

here $F(\sigma;\sigma_0) \sim 1-\exp(-(\sigma-\sigma_0)^2)$ is some increasing bounded function which we use to contain J (as defined in Equation 12) and bound the accepted loss.

If we look at the form of our example function for F, this maps any noise to the interval $[0,1)$, or $F:\mathbb{R}\to[0,1)$. In the trivial case, where $\sigma=\sigma_0$, we get $F(\sigma;\sigma_0)=0$ and thus $J(\sigma_0;\mu^*)=\mathbf{L}_j(\mu^*)$. For $\sigma\to\infty$, $F(\sigma;\sigma_0)\to 1$ and we get $J(\sigma_0;\mu^*)\leq 2\mathbf{L}_j(\mu^*)$.

Firstly, it is unreasonable to expect σ to tend to infinity. However, for noise levels different from the value where we initially computed μ^* – σ_0 – we would hope that our expected value would be contained within some small multiple of the base case. It would be nice in the case of our optical fibres knowing that if we were manufacturing cores with a given mean, that even if there was noise, our power loss wouldn't go out of control.

We examine this inequality for the case where $T=0.2$. For $\sigma_0=0$, we have $L^*=0.1159$. We calculate the expectation value as we increase σ, shown in table 2. We plot our test $\mathbf{L}_j(\mu^*)[1+F(\sigma;\sigma_0)]$ along with our calculated values of $J(\sigma;\sigma_0,\mu^*)$ in figure 8.

Table 2: Testing the robustness of calculated mean μ^* by varying σ.

Quantities	Catchy title that no one cares about						
σ	0.01	0.05	0.1	0.2	0.5	1	2
$J(\sigma;\sigma_0,\mu^*)$	0.1159	0.1156	0.1162	0.1167	0.1175	0.1298	0.1537

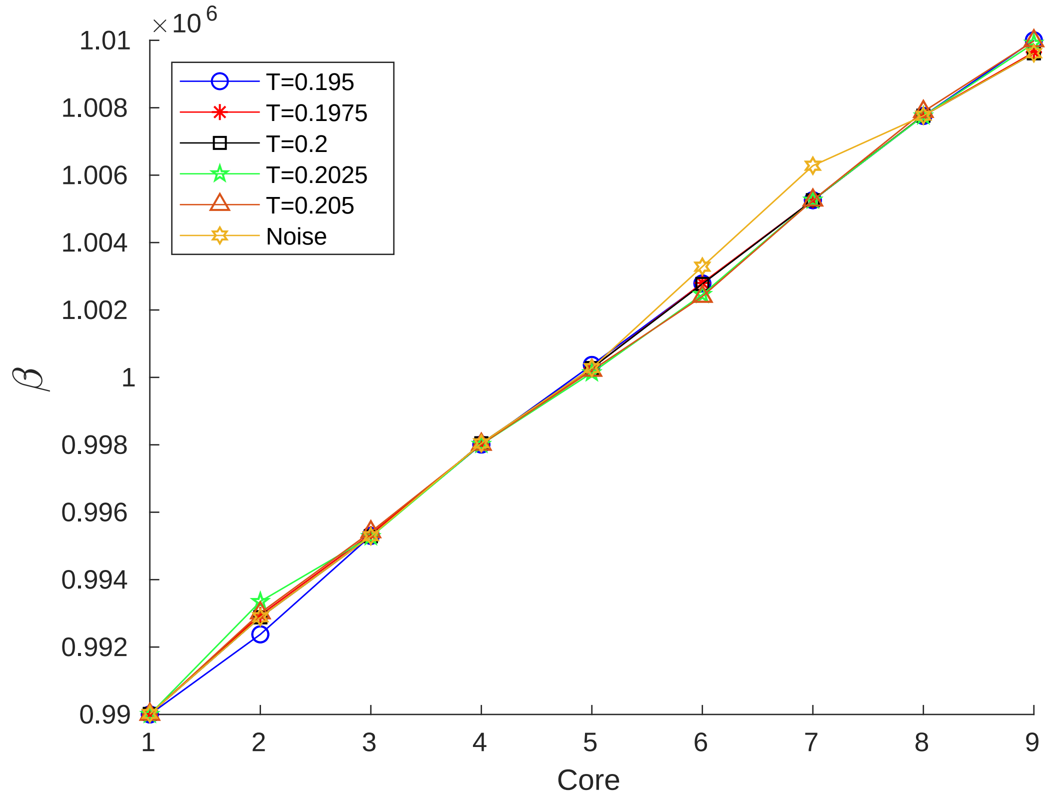

Figure 9: Set of βs for $T \in (0.195, 0.1975, 0.2, 0.2025, 0.205)$ and for the noisy case.

4.4 Robustness over unspecified length

Just as there will inevitably be defects in glass, we can expect similar discrepancies in the length of an optical fibre. We want as little deviation in the final intensity of a core as possible as we vary the length at which we are examining the final output. Here, we consider two important cases.

- Variance about a fixed length,
- Random length in an interval.

In the first case, we want to know how robust a configuration is with regard to length. This can take the form of optimising over $T + \Delta T$ where ΔT is small. Unlike perturbations in the β parameters, changing the length has a much higher impact on the expected loss. While stable from a β point of view, the solution is not stable from a length point of view. By this we mean that our set of mean values μ^* remain fairly constant as we increase σ as discussed in the previous section. If we have a length of optical fibre and we seek to maximise the core intensity at a specific length, this would mean that we are tuning our system so that over each core the power would just have been maximised. If we seek to look at how power output looks, adding or subtracting a section, then naturally the system will be leading to or moving away from that maximum. Thus it makes sense that the structure of our μ^* values changes as we vary length. In Figure 2, we see $L^* < 0.14$ for $T \in (0.195, 0.1975, 0.2, 0.2025, 0.205)$. However, as we add noise to the length, we observe a shift in the β set and we get an expected value of $L^* \approx 0.35$. We plot the β set for the difference lengths given above in addition to noisy length in Figure 9.

We see a shift in the structure, with the main difference being seen in the third largest value where there is a clear distinction.

In this second case, we examine what tile emerges when finding the expectation value of a given length T when taking a random length given by $t \in (0, T]$ uniformly distributed. In this case, we search for:

$$\mu^* = \underset{\mu \in \Omega^M}{\arg\min} \left(\underset{t \sim \mathcal{U}(0,T]}{\mathbb{E}} [\mathbf{L}_j(t, \beta)] + \lambda \mathbb{E}[(\mathbf{L}_j(t, \beta) - \mathbb{E}[\mathbf{L}_j(t, \beta)])^2] \right).$$

Thus, instead of being able to optimise over just the endpoints, the results here, need to be stable throughout the length of the entire cable. What is interesting, is that this is the same solution that the above case tends

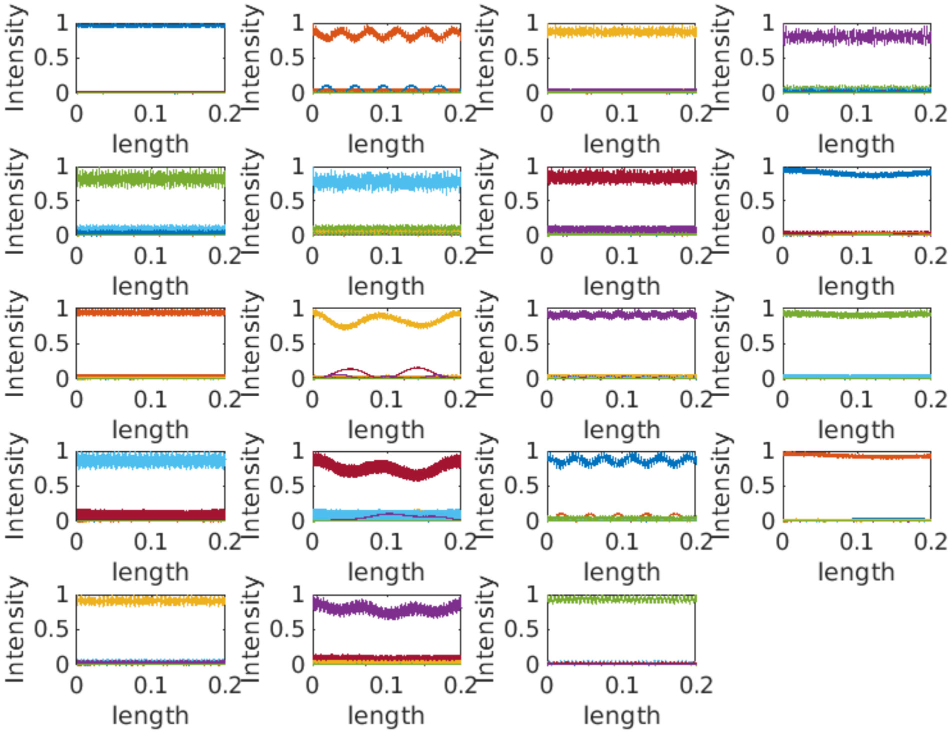

Figure 10: Power output for the uniform solution.

to. When noise above a threshold is added to a specific length, it changes state, moving to a more global stable state. We plot the power output for this uniform solution for length $T = 0.2$ in Figure 10. While it appears at first glance almost identical to the power plot in Figure 7, it actually has $4\times$ the loss at that specific length. However, it still behaves better a slighter longer or shorter cable.

5 Incorporating distance between cores in loss function

The distance between cores plays a crucial role in the coupling coefficients when examining the system of ODEs. However, increasing the distance between cores also increases the area of each core resulting in less cores in a given area. This poses a question regarding the relative importance between achieving minimum power loss and occupying the minimum area and where the emphasis lies. When we compare the results having increased distance between cores d from $5 \to 5.25$ it shows the importance of asking this question, examining the two difference cases:

5.1 Specified Length

The first observation is that the loss does indeed decrease for any given length. Firstly, in the interval between $0.195 \leq T \leq 0.205$, there is roughly a 25% decrease in power loss. Then, examining the expected value for $T = 0.2$ for given σ_0 we observe

5.2 Unspecified Length

Whether looking at a short length of cable $T = 0.2$ or a longer stretch $T = 2$, the difference that adding an extra bit of distance between two cores, makes a substantial difference. We compare the set of βs that were computed in 4.4 for the uniform case, applying it to the longer length of cable. A table showing in the computed values for the end points and expectation value for the interval are shown in Table 3. The two difference cases are also plotted in Figure 11 where we look at $d = 5$ and Figure 12 where $d = 5.25$. Although similar, from looking at the figures and Table 3 we can see that for the increased distance, the intensity is both larger and has less deviation.

Distance between Cores	Loss at $T = 2$	Expect Loss for $t \in U(0, 2]$
5	0.2699	0.3584
5.25	0.1410	0.1584

Table 3: Difference in expected loss for a length of $T = 2$ for different distances between cores.

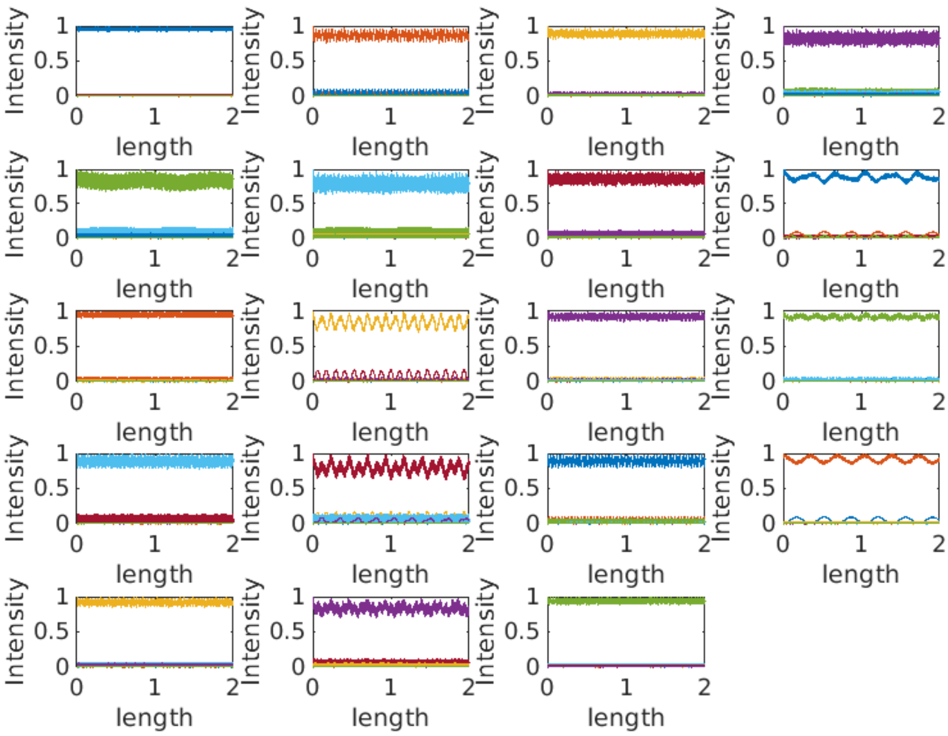

Figure 11: Power output for all 19 cores for $T = 2, d = 5$.The propagation constants β have been optimised for an unspecified length as in Section 4.4 for the given value of d.

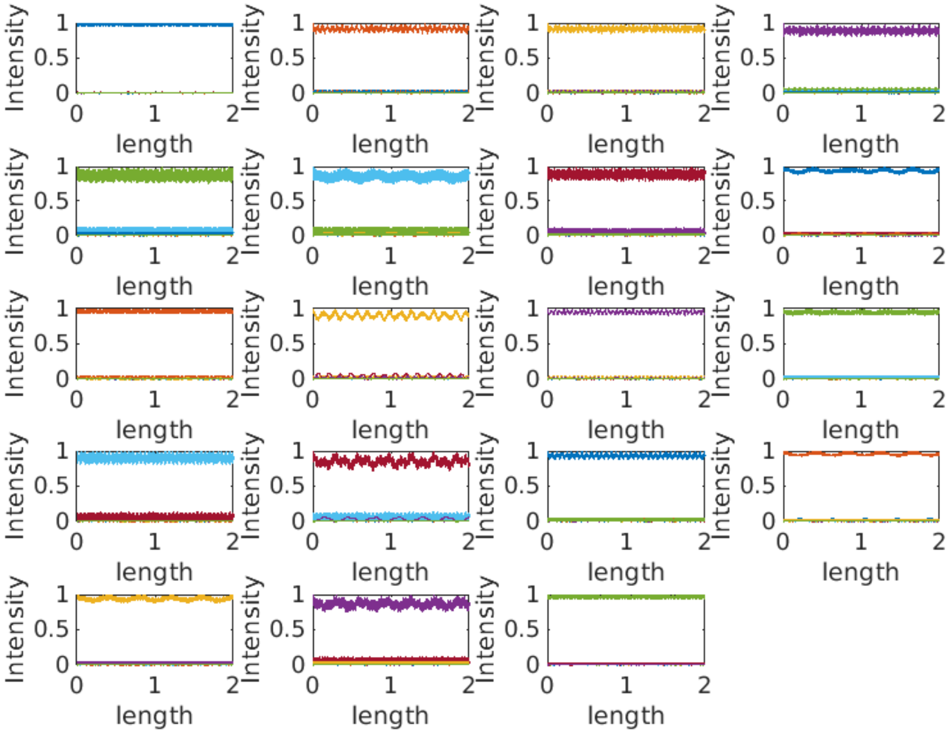

Figure 12: Power output for all 19 cores for $T = 2, d = 5.25$. The propagation constants β have been optimised for an unspecified length as in Section 4.4 but for the increased value of d.

6 Conclusions & Future work

There is rich detail in considering how to optimise a multicore fibre. We started off by considering how we would even approach construction in terms of the tile and sub-tile. Following this we examined key parameters in how the system evolves such as the propagation constants as well as the fibre length. Now we finish with our final outlook.

6.1 Conclusions

From our initial experimentation on construction of a tile, using a sub-tile of $M = 9$ works nicely. It results in a nice balance between power transfer and number of cores. In addition considering the problem in terms of 19-MCF tiles instead of a single 10000-MCF is hugely important. Experimentation between the 19-MCF and 37-MCF showed huge benefits for the 19-MCF with a layer of cladding.

Thinking of the problem as now trying to optimise a 9 sub-tile in a 19-tile, we then examined how length and propagation noise effected power output. What we noticed was that an optimised solution for a given length was not hugely impacted by noise to the βs. What was interesting though, is the effect that length variation caused, changing the structure of the solutions. Solutions that were specified for a given length performed poorly over fluctuations in length. As such, it may be more valuable to consider solutions which perform well over the full length of the cable.

Finally we consider the change that distance between cores has on the power transfer. It is remarkable that small increases have such a disproportional effect in reducing power loss. This is really highlighted in Table 3.

6.2 Future Work

Moving forward there are many questions which are raised from examining the system. One of the clearer ones is studying how power loss decreases when the parameter d is adjusted in the interval $[5, 5.25]$. Given a power reduction of over 50% in the expected value over the optical fibre, what is the best distance between cores?

From studying the effect that length has on the solution structure this raises the question of what makes some solutions more stable/unstable than others and why some can deal with noise in the length/propagation constants. In addition can we predict how the specified length problem evolves? If we look at Figure 5 we can see all solutions are close together, although differences are noticed in the second, sixth, and last values. Can we figure out how the β set evolves as length changes?

References

[1] Hui Yuan, Marija Furdek, Ajmal Muhammad, Arsalan Saljoghei, Lena Wosinska, and Georgios Zervas. Space-division multiplexing in data center networks: on multi-core fiber solutions and crosstalk-suppressed resource allocation. *Journal of Optical Communications and Networking*, 10(4):272–288, 2018.

[2] Allan W Snyder and John Love. *Optical waveguide theory.* Springer Science & Business Media, 2012.

[3] Tamara G Kolda, Robert Michael Lewis, and Virginia Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482, 2003.