Summer Research Internship Report

Jake Denton

June-August 2021

1 Introduction

1.1 Background

Study and analysis of techniques used in the production of composite materials are of growing importance due to the wide range of applications these materials play an integral part in. The investigation of the properties that composites possess from the geometries that underpin them is key to cut costs and optimise effectiveness for industries such as aerospace, vehicle and sports manufacturing as well as in civil infrastructure and construction. One area that can cut costs and reduce risk of human error is automated fibre placement (i.e. use of a robot to construct the material). This project further develops the investigation of the automated placement process, with emphasis on how the errors in placement affect permeability and through this, properties like strength following the binding of the dry fibre preform in resin transfer moulding.

1.2 Outline

This report is structured as follows. First, a glossary is provided to describe various engineering terms that may not be familiar to mathematicians. Following this, there is a section describing where the data originates from, how it is converted from gap widths to permeabilities and how the information is stored in matrices that can be used in MATLAB, which is recommended for studying this data (since all the files available are written in this language). Previous work is discussed after this, with emphasis on the censoring of data, the form of the marginal distribution, and analysis in the time and frequency domains. Moving on, Gaussian processes are motivated and linked to a model which incorporates the dependencies between data points at fixed distances apart. The features of the sample autocorrelation are described before potential candidates for the covariance functions of the processes involved in the model are analysed. Finally, the findings are discussed and there are some suggestions for further work.

1.3 Glossary

This glossary is included in order for readers to acquaint themselves with certain technical and non-technical terms which will be used throughout this report.

Yarn - A spun thread
Tow - A yarn of carbon fibre, made of 24000/24K strands in our case, giving an overall width of 6.35mm.
Tape - 8 tows placed alongside each other with no programmed gap (width 6.35*8=50.8mm)
Gap - The programmed gap (programmed to be 1mm in width) between two tapes. Adjacent gaps refers to gaps separated by a single tape.
Gap Width - The shortest distance between the edges of two tapes.
Layer - One slice of the preform, of dimensions 1m x 1m, consisting of 18-19 tapes arranged alongside each other by the robot with 1mm gaps between. Any discrepancy from 1m left over by these tapes is filled with further tows to ensure any given layer is 1m x 1m. Adjacent layers (layers above or below the given layer) have tows arranged perpendicular to the given layer (a cross-ply design).
Preform - The whole dry fibre structure (16 layers).
Pixel - One way to specify positions on an image, 1 pixel=$35.3*10^{-6}$m.
Permeability - The ability of a material to be penetrated or passed through.
Porosity - The proportion of empty space in a material.

2 Data

2.1 Structure Of The Preform

Before describing the form of the data given, it is worthwhile describing the structure of the preform itself. This can be inferred from the relevant definitions in the glossary above but will be reiterated for clarity here. The preform (dry fibre structure) is comprised of 16 layers. Each layer is formed by a robot which places the tows that make up each tape onto the 1m x 1m panel. 8 tows are placed alongside each other to make a tape, then there is a programmed gap of 1mm before the next tow is placed and the next tape begins. This process continues all the way along the panel. After each layer is completed, its surface is split into a 12x7 grid and an image is taken by a camera at each cell of the grid. The tapes on each layer are placed perpendicular to the tapes on layers before, so in this way the direction of the gaps alternate. This creates two distinguished classes of layers, namely odd and even layers. There are 8 of each (since there are 16 layers). There are relative shifts of 3.5 tow widths in each layer relative to the previous layer of the same direction, which shifts back for the following layer of the same direction (more easy to see via the diagram). For more information on the exact dimensions of parts of the preform (e.g. layer height), see Matveev et al.[2]

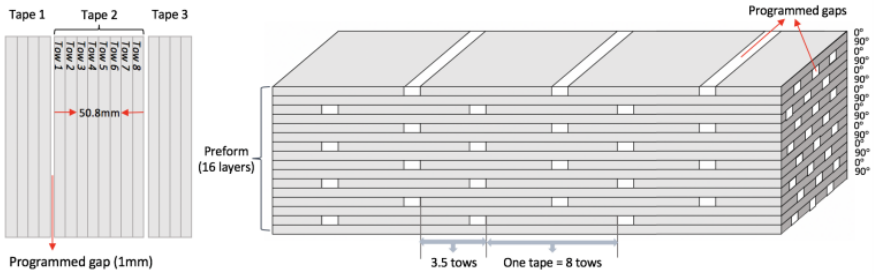


Figure 1: Structure Of The Preform

A great visualisation of this is given in the diagram above from the dissertation by Fay [1]. The diagram shows a chunk of the preform.

2.2 Gap Data Acquisition

The permeability of greatest interest is that along the gap, as the gap allows the resin to resonate through the preform to bind the dry fibres together. Consequently, gap width is an independent variable which is found from the gap data, obtained using an automatic image analysis system on the snapshots of each layer described above. This system is not reliable for images on the edge of the layers, so we should only consider data from images within the inner 10x5 grid.

Obviously, it is preferable to combine the data from many of these images to make analysis simpler. The method used to do this is called stitching. Stitching works by observing the direction of the gap for the layer then aligning images perpendicular to this direction (this is much easier than trying to match the ends of the gaps that the grid cuts apart). For odd layers, where the direction of the gaps is along the horizontal, the images are stitched vertically creating 5 columns which will be referred to as subsets. On the other hand, since the direction of the gaps for even layers is vertical, the images are horizontally stitched creating 10 rows (also referred to as subsets). To mitigate the need to remember which layer is in which direction, it is beneficial to rotate the even subsets so that both classes of layer share the same direction of gaps. This is done as shown in the diagram below from page 11 of Faye's dissertation [1].

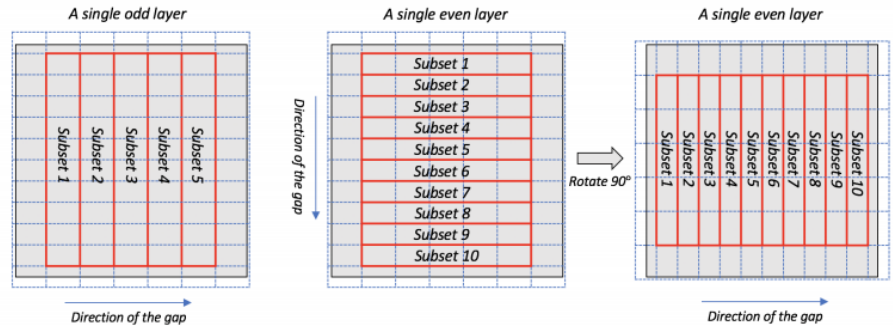


Figure 2: Subsets Of Odd/Even Layers

2.3 Structure Of The Data

It is from these subsets that the data is given. This data can be found in the By column folder, and needs to be added to the MATLAB path in order to be accessed in a script. The notation PkLl-m refers to the preform number k (there were two preforms produced, see Matveev et al.[2] for more details), the layer number l (from 1-16) and the column/row number m for odd/even layers respectively. The data in the By column folder contains information on the first preform. Note that despite there appearing to be 7 columns for odd layers and 12 rows for even layers in this folder, the edges have severe errors which is why subsets were introduced above. This means that only columns 2-6 (odd subsets 1-5) and rows 2-11 (even subsets 1-10) should be used from each layer.

To read in the data (which is available in .txt format), first use the *dir* function to list the contents of the By column folder. For example, *dir('By column/P1L*col*.txt')* picks out all the .txt files in the folder containing the wildcard expression *col* meaning the columns, so this isolates all the data from odd layers. This command defines a structure, and if it contains more than one element then it can be manipulated in a similar way to a vector in order to call individual files. To read individual files, use the *dlmread* function. For example, *dlmread(oddnames(j).name,' ',3,0)* reads the j-th file from the structure oddnames from the fourth row onward. These two examples were taken directly from code available for this project. The *permeabilities.m* script is a good place to start to get affiliated with the MATLAB commands relevant to this work.

All of the matrices obtained from reading in each file share the same structure. They have either 290 or 306 columns, but the final column contains only zeroes so should always be removed. The first column of each matrix describes the position in pixels along the subset. This position increases in increments of 5 pixels down the column from 1330 to 7060. Each column after the first

represents the position of tow edges going perpendicular to the direction of the gaps. Recall that there are 8 tows in each tape and each tow has 2 edges therefore the relevant columns corresponding to the start/endpoints of the gaps are 17/18,33/34,49/50, ... ,273/274 and (if the matrix has 306 columns) 289/290. All positions are given in terms of pixels, each of which represents a distance of $35.3*10^{-6}$m. The diagram below visualises this. It may be easier to think of the first column as an X direction (going along the gap) and for a fixed row/position, elements beyond the first column represent going along a Y direction (perpendicular to the gap).

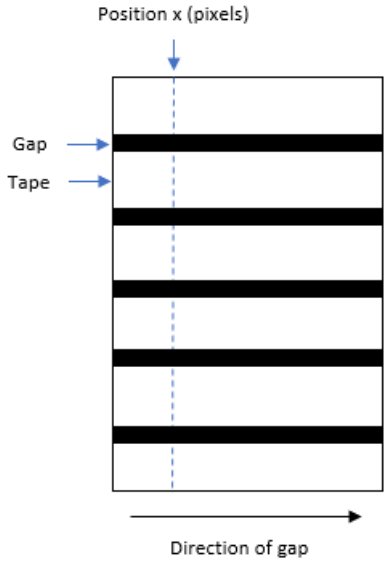


Figure 3: Diagram depicting fixed position x along gap

Besides the individual row/column data, there are also a few files available for which the information of an entire layer has been stitched together, which can be found in the New_stitched folder. For the first preform (P1) layers 1,3 and 5 are available. The position (first) columns of these matrices range from 1345 to 33325 pixels. The first and seventh parts (these are the **edges** of the layer which are normally not considered) have not been removed and there are unreliable overlapping regions where the columns have been stitched (see Laurence's report [3] for more details), so in practice these files need quite a lot of cleaning to be used for analysis. Again, to use these files, remember to add the folder to the pathway.

2.4 Conversion To Permeability Data

Permeability data is obtained as a tensor (a matrix of values in MATLAB). Each element of the tensor represents permeability along a different direction. Permeability along the gap, K_{xx} (measured in m^2), is most relevant to the investigation and so any permeability modelled or shown graphically in what follows refers to this value. This value can be found only for 3-dimensional objects (due to the script files used and the definition of permeability), so the gap is split up into many small volume elements which is necessary in order to study how K_{xx} evolves along the gap. This is shown in the diagram below:

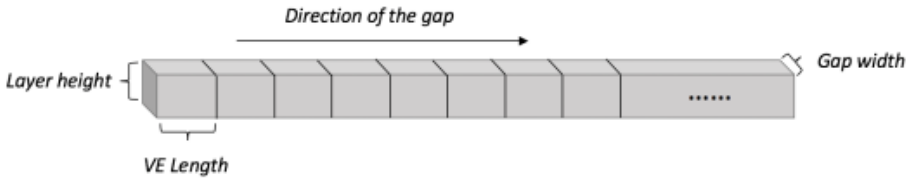


Figure 4: Gap split into individual volume elements (or boxes)

For each volume element, methods described in page 10 of [2] can be implemented to obtain K_{xx} using *calc permeability v2.m*, which calls upon *Gebart function.m* and *rectangle_permeability.m*. The inputs of this function should be in metres, and are described in detail on page 8 of [3]. The volume element length is a variable that is decided outside the functions above (the function asks for the mean gap width along the box, and also the orientation which requires the midpoints of the start/end of the box), but it will most often be set to 5 pixels for reasons set out in the next section.

3 Previous Work

3.1 Censoring Of Data

A great deal of analysis has already been done on the data this report focuses on. This section aims to summarise the main results of this previous work.

A good place to start is censoring of the row and columns. In [3], the mean gap 1 widths are plotted for the odd and even layers. The gap widths are non-uniform, with the variance of observations increasing beyond 3700/4000 pixels for rows and columns respectively. An explanation for this behaviour can be found by looking at the images. To obtain contrast between tows of current and previous layers required for the gap detection to be reliable, a powerful light was shone onto each layer before the image was taken. Unfortunately, binder veil (a substance coating the tows) is an excellent reflector of light and this distorted the images, resulting in the variance observed in the

plots of [3]. In response to this, only gap widths before 3700/4000 pixels are considered from each image in the MATLAB scripts. Due to overlaps, data is also not considered until after a certain distance in pixels, often the 30th row of the matrix. See the below image for an example of light distortion.

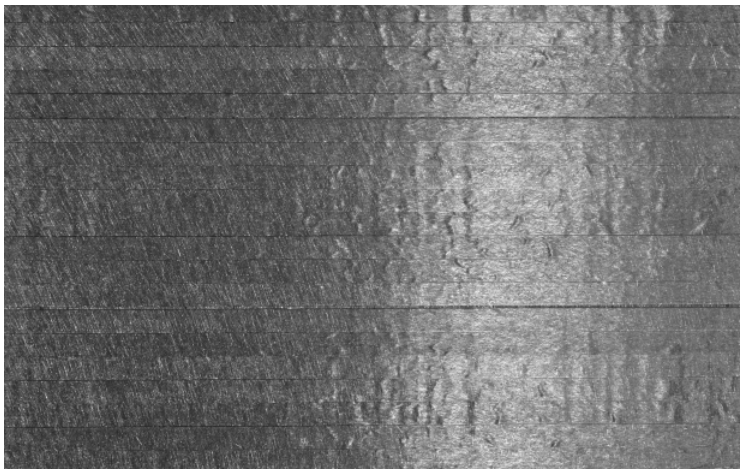


Figure 5: Image for Layer 1 Column 3

Layers 12 and 14 should also be ignored as they give erroneous results.

3.2 Fitting the Marginal PDF

The odd and even layers were both assembled the same way, so it seems reasonable to expect them to be mutually independent realisations of the process with the same distribution. This was checked in [1] using the Mann-Whitney U-test, and for box length of less than 60 pixels the test was rejected. However, since the processes were the same for both types of layers in reality, both layers will still be considered together. It is thought that the reason for this strange result is slight differences in image quality between the two types of layers.

Random independent samples of K_{xx} were then taken with the aim of fitting a model (or marginal PDF) to describe the permeability of any box along the gaps. Contrary to previous research which suggests that the log-normal distribution is the suitable model for data of this type, [1] found that after parameter estimation through likelihood maximisation, the log-normal, normal and truncated-normal distributions are not suitable for describing the samples. Skewness was found via the adjusted Fisher-Pearson coefficient which indicated a right (or positive) skew. Based upon these findings, a Gaussian mixture model was considered. This model is essentially a weighted sum of multiple normal distributions, each of which is assigned a mixing coefficient

which represents the probability that a given point x comes from that normal distribution (all of which sum to one). It was determined that there should be two components involved in the modelling of these samples.

Parameters were estimated through the expectation-maximisation (EM) algorithm. The two components are responsible for the extreme data (far from peak) and for the data at and close to the peak respectively. This model fits well for all VE lengths considered, with all p-values calculated (testing the null that the model is indeed a Gaussian mixture) exceeding 0.05. As a result, the following marginal distribution has been established for the permeability at any point along the gap for a VE length of 5 pixels:

$$p(\mathbf{x}) = 0.2649 * \mathcal{N}(\mathbf{x}_1 | 0.3714 * 10^{-10}, 0.2163 * 10^{-21}) + \\ 0.7351 * \mathcal{N}(\mathbf{x}_2 | 0.3226 * 10^{-10}, 0.0377 * 10^{-21})$$

where in each component: the multiplier is the mixing coefficient, the second number is the mean μ and the smallest number is the variance σ^2.

3.3 Random Processes

Since the marginal distribution is based upon independent observations, the next natural step in understanding the permeability of the preform is to account for the dependencies between different boxes along the same gap. The study of these dependencies is known as *time series analysis*. To begin with this analysis, a few relevant terms are defined below then a short summary of previous investigations of the data in the time and frequency domains is provided.

A random process is an ordered sequence of random variables, defined on a set of time points. If we let the distances along the gap denote the time points, then both the local permeabilities (at the boxes) and the gap widths can be considered as discrete random processes.

A random process is called strictly stationary if the joint probability density function of the variables in time (or distance in our case) remains the same even after a translation. Similarly, a random process is weakly stationary (or wide stationary) if its mean is constant and its autocovariance function (explained in the next paragraph) depends only on the distance (or lag) between points. The gap widths appear to be weakly stationary and hence it can be assumed that the permeabilities are also weakly stationary as a function of these gap widths (see *Correlation theory of stationary and related random functions* by A.M. Yaglom for more details).

The autocovariance function measures the linear dependence between two points observed at different times, and is defined as:

$$\gamma_x(s,t) = cov(x_s, x_t) = E[(x_s - \mu_s) * (x_t - \mu_t)]$$

Here, x_s and x_t are observations at times s and t. Note how this function encodes the information relating to the variances of every point in the process. Consequently, if the random process is Gaussian (or involves normal variables) then knowing the mean/autocovariance function means you know the distribution of the process for any number of points since normal variables depend only on their first and second moments.

The autocorrelation function measures the linear predictability of x_t using x_s, and is defined as:

$$\rho_x(s,t) = \frac{\gamma_x(s,t)}{\sqrt{\gamma_x(s,s) * \gamma_x(t,t)}}$$

Observations of the time series at t, X_t, are known as realisations of the variable x_t. In practice, the autocorrelation function is difficult to calculate, so it is estimated using the sample autocorrelation function. This is defined as:

$$\hat{R}(\tau) = \frac{1}{Ts^2} \sum_{t=1}^{T-\tau} (X_t - \overline{x})(X_{t+\tau} - \overline{x})$$

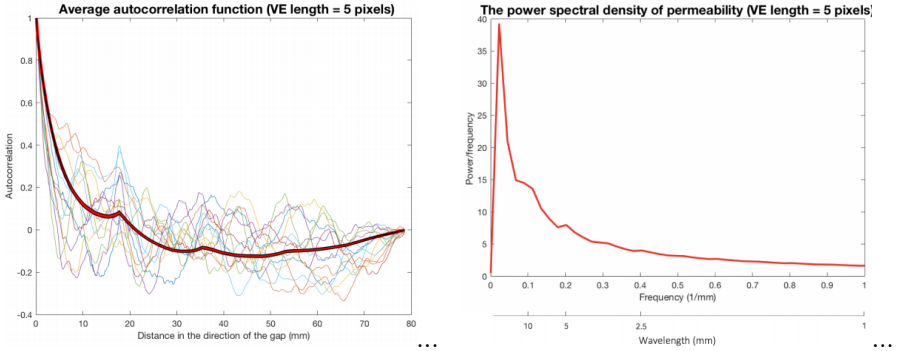
Here, T is the number of observations, τ is the lag (or distance), s^2 is the sample variance and $\overline{x}$ is the sample mean. Study of the autocorrelation function is regarded as an analysis of the *time* domain. Another way to analyse the series is through the *frequency* domain and the *power spectral density*.

The power spectral density can be obtained by taking the Fourier transform of the autocorrelation function via the Wiener-Khinchin theorem. For this discrete random process, define this density by S with frequency f (f is any real number). Recalling the discrete Fourier transform, the power spectral density (or PSD) is given by:

$$S(f) = \sum_{\tau=-\infty}^{\tau=\infty} R(\tau) exp(-i(2\pi f)\tau)$$

3.4 Autocorrelation and Power Spectral Density

The autocorrelation of the local permeabilities along each gap were considered in [1]. Since the autocorrelation function does not go near zero until large distances, the permeability of boxes along the gap are correlated. In the frequency domain, a strong peak at a wavelength (1/frequency) of 44.64mm can be observed for a volume element length of 5 pixels. This interesting peak tells us that there is a lot of power and information available for this length, so this is kept fixed for all analysis done later in this report.



(a) Autocorrelation Function (b) Power Spectral Density

Figure 6: ACF/PSD for VE length=5 pixels

Similar analysis was done on the gap widths in [3]. Investigation into autocorrelation led to a conservative estimate of 100mm for the distance at which boxes can be thought of as independent (autocorrelation is very near zero). This means that boxes along the same gap which are at least one subset apart can be considered as independent.

The hypothesis that adjacent gaps are independent was also tested, with a p-value of 0.6724 confirming that the hypothesis is reasonable. Sections of adjacent gaps at the same distance along the gap were also paired to check for independence. This time, a very weak correlation of around 0.05 was found from 500 collections of data. See *gapsection.m* for details.

3.5 Summary of Previous Work

In summary, when gaps within each subset are considered, only boxes between the distances of 1480 pixels (52.244mm) and 3700/4000 pixels (130.61 / 141.2mm) for rows/columns respectively should be considered to avoid the effects of overlaps and light distortion, and layers 12 and 14 should be ignored. The marginal distribution of the permeabilities has been established as a two-component Gaussian mixture. Lastly, sample autocorrelation and power spectral density were defined and considered. Plots of these show that

two boxes along the same gap are correlated if they are within a distance of about 100mm away from each other, and there are two significant peaks in the power spectral density for 5 pixels - the first of which being at wavelength 44.5mm. These peaks are not as powerful for larger VE lengths, so the VE length is fixed at 5 pixels.

4 Analysis of Permeability

4.1 Gaussian Process Model

As discussed in section 3.3, to develop the permeability model the dependencies between boxes along the gap need to be incorporated. Since the marginal is a Gaussian mixture model, this hints that the larger model should consist of a combination of Gaussian processes (or random fields). A Gaussian process is a random process for which any finite set of variables have a multivariate normal probability distribution function.

The model should be smooth, and the distribution of any single variable at a fixed distance along the gap should match the marginal (two component Gaussian mixture model). As there are two normal components within the marginal distribution there should be two Gaussian processes replacing these which we denote X_1 and X_2. These are assumed to be independent of each other. Within the model, the processes should have mean m_1/m_2 and variance σ_1^2/σ_2^2 coming from the marginal. However, thanks to the properties of normal variables, the two processes can be easily reconstructed into processes of mean zero and variance one via the decomposition $m + \sigma * X(x)$. Clearly the expectation of this sum is m whilst the variance is σ^2 as desired. This reconstruction simplifies later calculations. Since the permeability is assumed to be a weakly stationary process, the above two processes and any model constructed from them are all assumed to be weakly stationary.

There are also mixing coefficients present in the marginal distribution ($\pi_1 = 0.2649$ and $\pi_2 = 0.7351$). These mixing coefficients are parameters involved in latent variables that determine the identity of the data point i.e. which component of the mixture is responsible for each data point (see [1] for more details). We will denote the latent process that replaces the latent variables by X_3, which we assume to be a weakly stationary Gaussian process that is independent of X_1/X_2 with mean zero and variance one. This latent process is present as the argument of a *link* function - a function that determines which of the other two Gaussian processes is responsible for a point.

Ideally, the value of the link function should be zero or one. This ensures that any observation comes from either the term containing X_1 or X_2, and not a combination of both. In this way, the model acts similarly to the marginal distribution, where each observation comes from only one of the two Gaussian components. Of course, the obvious candidate is the indicator function which

can only take values 0 or 1. However, the indicator function is discontinuous at the point where it changes from 0 to 1 and so violates the requirement of smoothness. As a result, we should consider using a continuous approximation of the indicator function.

There are a number of functions that can act as smooth approximations of the indicator function. Examples include the sigmoid functions of the form $\frac{1}{1+exp(-\alpha x)}$, and inverse tangent functions of the form $\frac{2}{\pi}arctan(\alpha x)$ and the cumulative distribution functions (CDFs) of many random variables. One function that gives the user more control over the steepness of the shift and also has the advantage of being easily interpretable is the cumulative distribution function of normal variables. Clearly, this function is smooth and moves from very near zero to very near 1 quickly due to the bell-shape of the normal distribution. Reducing the variance of the underlying normal variable narrows the bell-shaped distribution, in turn sending the cumulative distribution function to a value near 1 from 0 over a shorter interval in x. The variance is very easily changed with MATLAB's in-built normcdf, and even using relatively large values for the variance gives a reasonable approximation. The CDF of a $\mathcal{N}(0, 0.1^2)$ variable can be seen along with the exact indicator function in the image below:

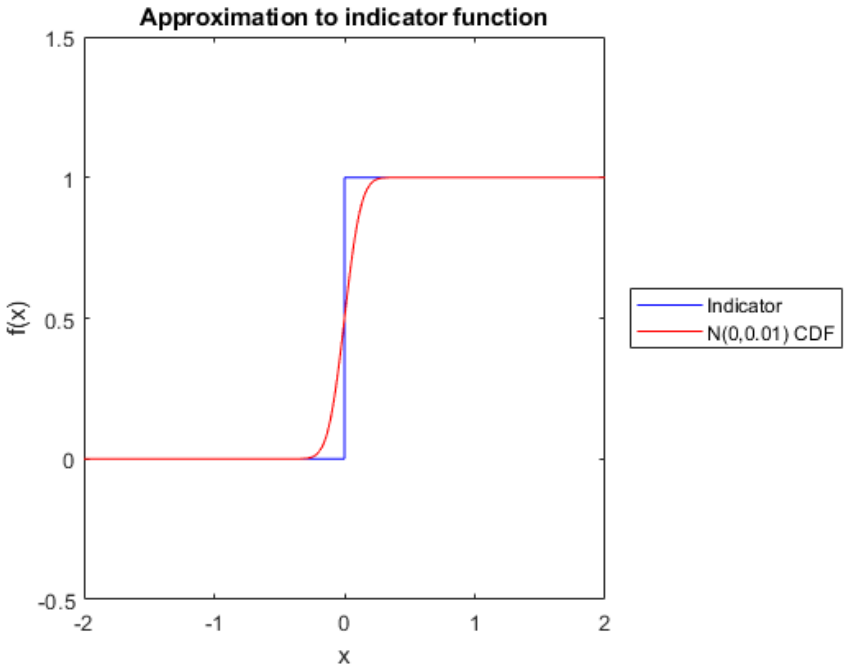


Figure 7: Normal CDF Approximation

Let the smooth CDF link function be denoted $\Phi_{m,\sigma}$, depending on the latent process X_3 which consists of individual $\mathcal{N}(0,1)$ variables. Here, m and σ are the mean and standard deviation of the underlying normal variable. Then the model proposed for the data, at distance x along the gap, is as follows:

$$\begin{aligned} Y(x) =&(m_1 + \sigma_1 * X_1(x)) * (1 - \Phi_{m,\sigma}(X_3(x)) \\ &+ (m_2 + \sigma_2 * X_2(x)) * \Phi_{m,\sigma}(X_3(x)) \end{aligned}$$

To connect the time series model with the marginal distribution and find a value of m given σ, the link function should depend in some way on the mixing coefficients. This influence can be expressed mathematically by setting $E[\Phi(X_3)] = 0.7351$ (π_2). Fixing the expectation forces approximately 73.51% of observations to come from X_2, matching the marginal distribution. The expectation can also be manipulated in order to find the mean of the underlying normal variable given σ^2. From the law of total probability:

$$E[\Phi_{m,\sigma}(X_3)] = \int_{-\infty}^{\infty} \Phi_{m,\sigma}(X_3) * \phi(X_3)\, dx_3 \qquad (*)$$

where ϕ is the standard normal probability density function.

To evaluate (*), first define A as the underlying normal variable of Φ with distribution $\mathcal{N}(m, \sigma^2)$. Then:

$$\begin{aligned} &P(A \leq X_3 | X_3 = \omega) = P(A \leq \omega) = \Phi(\frac{\omega - m}{\sigma}) && \text{(Condition on } X_3\text{)} \\ &P(A \leq X_3) = \int_{-\infty}^{\infty} \Phi(\frac{\omega - m}{\sigma}) * \phi(\omega)\, d\omega && \text{(Equal to (*))} \end{aligned}$$

In the first step, X_3 is conditioned upon then $\Phi_{m,\sigma}(\omega)$ is expressed in terms of the standard normal CDF. In the second step, the law of total probability is applied. Since A and X_3 are both normal variables, A-X_3 is normal with mean m and variance $1+\sigma^2$. Using this:

$$\begin{aligned} P(A \leq X_3) &= \int_{-\infty}^{\infty} \Phi(\frac{\omega - m}{\sigma}) * \phi(\omega)\, d\omega \\ &= P(A - X_3 \leq 0) \\ &= \Phi(\frac{-m}{\sqrt{1+\sigma^2}}) \end{aligned}$$

Therefore, $E[\Phi_{m,\sigma}(X_3)] = \Phi(\frac{-m}{\sqrt{1+\sigma^2}})$, where Φ is the standard normal CDF. This can now be equated to π_2 and rearranged for the mean m. The result

is $m = -0.6283 * \sqrt{1+\sigma^2}$ which is negative for all values of σ. This makes sense intuitively, as the link function caps off at 1. Increasing the expectation beyond 0.5 (which is the expectation of the standard normal CDF) moves the CDF left, sending the value of the link function near 1 earlier in the domain. A shift of the CDF in the negative direction beyond X=0 can only come as a result of the mean of the underlying distribution going under zero, so the bell-curve also moves towards negative infinity.

4.2 Model Autocorrelation

Testing the model defined in the last section against the data is complicated. Simulating the model directly for permeabilities along a gap and comparing with the data won't do as it is too difficult to observe visually whether or not the model does fit the data in this way. One better way in which the goodness of fit can be tested is by comparing the autocorrelations of the model and sample. The autocorrelation and it's estimator from data (the sample autocorrelation) were defined in section 3.3. If many independent realisations (or gaps) of the permeability data are considered and the corresponding sample autocorrelation is found for each, then if the average sample autocorrelation matches the autocorrelation of the model the form of the proposed model may be deemed suitable for the data. Therefore, these two quantities need to be found. This subsection deals with deriving the model autocorrelation. Recall that the proposed model is of the following form:

$$\begin{aligned} Y(x) =&(m_1 + \sigma_1 * X_1(x)) * (1 - \Phi_{m,\sigma}(X_3(x)) \\ &+ (m_2 + \sigma_2 * X_2(x)) * \Phi_{m,\sigma}(X_3(x)) \end{aligned}$$

To obtain the autocorrelation, the autocovariance must be found (the numerator of the autocorrelation) as well as the variance of the model (denominator). The autocovariance is given as:

$$\gamma_Y(x) = E[Y(0)Y(x)] - E[Y(0)]^2$$

This simplified formula comes from the fact that weakly stationary processes have constant means and autocovariance functions that depend only on the distance between two points, x. The variance is found by setting x=0 in the above.

E[Y(0)] can easily be found using the fact that the expectations of $X_1(0)$ and $X_2(0)$ are zero as standard normal variables (see previous section), and the constraint that $E[\Phi_{m,\sigma}(X_3(x))] = \pi_2 = 0.7351$:

$$E[Y(0)] = 0.2649 * m_1 + 0.7351 * m_2$$

where m_1 and m_2 are the means of the components from the marginal distribution.

The first term in the autocovariance function is more complicated. Using the model, the term can be expanded:

$$\begin{aligned} E[Y(0)Y(x)] =& (m_1^2 + \sigma_1^2 E[X_1(0)X_1(x)]) * E[(1 - \Phi_{m,\sigma}(X_3(0)))(1 - \Phi_{m,\sigma}(X_3(x)))] \\ &+ m_1 m_2 * E[\Phi_{m,\sigma}(X_3(0))(1 - \Phi_{m,\sigma}(X_3(x)))] \\ &+ m_1 m_2 * E[(1 - \Phi_{m,\sigma}(X_3(0)))\Phi_{m,\sigma}(X_3(x))] \\ &+ (m_2^2 + \sigma_2^2 E[X_2(0)X_2(x)]) * E[\Phi_{m,\sigma}(X_3(0))\Phi_{m,\sigma}(X_3(x))] \end{aligned}$$

The unknowns in this term are the covariance functions of X_1 and X_2 and the expectation in the final line (all other expectations can be expanded into sums involving this unknown).

Firstly, the covariance functions must be chosen. These functions should decay as the distance x increases, and are restricted in the sense that the covariance matrix arising from them must be *positive semi-definite*. Since all covariance matrices are symmetric and their diagonal elements (representing variance) are positive, this restriction is equivalent to the matrix being diagonally dominant. A diagonally dominant matrix is a matrix where the absolute value of each diagonal element exceeds the sum of the absolute values of the off-diagonal elements over the corresponding row.

One popular choice of covariance function (sometimes referred to as kernel) is the Matérn. The Matérn covariance between two points separated by distance x is:

$$C_\nu(x) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu}\frac{x}{l})^\nu K_v(\sqrt{2\nu}\frac{x}{l})$$

where $\nu > 0$ is the smoothing parameter, $\Gamma()$ is the Gamma function, $l > 0$ is the correlation length and K_ν is the modified Bessel function of the second kind. This formula simplifies to the product of an exponential and a polynomial order p when $\nu = p + \frac{1}{2}$ is a half-integer.

The benefit of using this model is that the smoothing parameter is chosen by the user. A Gaussian process with Matérn covariance is $\lceil\nu\rceil$-1 times differentiable in the mean-square sense (see Wikipedia page on Matérn covariance function for other properties). Initially, this covariance function will be used for the three processes with $\nu = \frac{5}{2}$, but later it will be discovered that for X_1/X_2 this kernel is insufficient to fit the autocorrelation of the model with the data.

Within this kernel and other kernels which will follow later, the correlation length l must be declared. This parameter is set equal to the reciprocal of

the frequency at the first peak in the power spectral density for X_2 (the most powerful peak), which is given as 44.5mm in [1] and 40.7mm in [2] but takes value 40.099mm in the spectral density discussed later in this report. For X_1, l is the reciprocal of the frequency at the second peak (approximately 16.3mm from [2], 10.025mm later in report). For X_3, the correlation length can be chosen freely, but for now it is set equal to the average of the above lengths.

The second unknown (the expectation involving the link function $\Phi_{m,\sigma}$) can be written as the following double integral by rearranging for the standard normal CDF then conditioning with $X_3(0) = \omega_0$ and $X_3(x) = \omega_x$ before using the law of total probability:

$$E[\Phi_{m,\sigma}(X_3(0))\Phi_{m,\sigma}(X_3(x))] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Phi(\frac{\omega_0 - m}{\sigma})\Phi(\frac{\omega_x - m}{\sigma})P_{X_3(0),X_3(x)}(\omega_0, \omega_x)\, d\omega_0\, d\omega_x$$

Here, Φ is again the standard normal CDF in the first two terms of the product. The third term is the bivariate normal PDF of two correlated standard normal variables (recall that X_3 is a Gaussian process with normal variables of mean zero and variance one). The covariance matrix associated with this PDF is:

$$\Sigma = \begin{pmatrix} 1 & E[X_3(0)X_3(x)] \\ E[X_3(0)X_3(x)] & 1 \end{pmatrix}$$

The off-diagonal elements are known assuming the Matérn $\frac{5}{2}$ covariance function with correlation length discussed above. This integral is clearly very difficult to evaluate thanks to the non-zero covariance in the matrix. As a result, this integral is computed numerically using the *integral2* function in MATLAB.

The variance of the proposed model Y also needs to be computed before the autocorrelation can be calculated. The standard formula for variance is:

$$\sigma^2 = E[Y(0)^2] - E[Y(0)]^2$$

Substituting x=0 into Y(x), squaring, taking the expectation and inserting $E[\Phi_{m,\sigma}(X_3(0)] = 0.7351$ and $E[X_1(0)^2] = 1$ (since $Var(X_1) = 1$) gives:

$$\begin{aligned} E[Y(0)^2] =& (m_1^2 + \sigma_1^2)(1 - 1.4702 + E[(\Phi_{m,\sigma}(X_3(0))^2] \\ &+ m_1 m_2(1.4702 - 2 * E[(\Phi_{m,\sigma}(X_3(0))^2]) \\ &+ (m_2^2 + \sigma_2^2)E[(\Phi_{m,\sigma}(X_3(0))^2] \end{aligned}$$

The only unknown here is the expectation on the right-hand side, which can be expressed as the following integral:

$$E[(\Phi_{m,\sigma}(X_3(0))^2] = \int_{-\infty}^{\infty} (\Phi(\frac{\omega_0 - m}{\sigma}))^2 P_{X_3(0)}(\omega_0)\, d\omega_0$$

To evaluate this integral, define two standard normal variables u and v, then:

$$\begin{aligned}\int_{-\infty}^{\infty} (\Phi(\frac{\omega_0 - m}{\sigma}))^2 P_{X_3(0)}(\omega_0)\, d\omega_0 &= E[P(u \leq \frac{\omega_0 - m}{\sigma}, v \leq \frac{\omega_0 - m}{\sigma})] \\ &= P(\sigma u - \omega_0 \leq -m, \sigma v - \omega_0 \leq -m)\end{aligned}$$

Now consider the transformed variables $\sigma u - \omega_0$ and $\sigma v - \omega_0$.

$$\begin{aligned}E[\sigma u - \omega_0] &= E[\sigma v - \omega_0] = 0 \\ Var[\sigma u - \omega_0] &= Var[\sigma v - \omega_0] = \sigma^2 + 1\end{aligned}$$

The integral is equal to the value of a bivariate normal CDF evaluated at *(-m,-m)* with:

$$\mu = (0, 0) \quad \text{and} \quad \Sigma = \begin{pmatrix} 1+\sigma^2 & 1 \\ 1 & 1+\sigma^2 \end{pmatrix}$$

When $\sigma = 0.1$, performing the above gives 0.7167. This has been confirmed numerically using the *integral* function in MATLAB.

All of the relevant terms to calculate the model autocorrelation have now been discussed. The function *YCovariance.m* calculates the model autocovariance from the following inputs: a vector of lags, correlation lengths and other parameters used within the covariance functions. There is also an input q which allows one to skip the time-costly numerical integration when the user has called the function once and does not wish to change the parameters of the X_3 process (calling this function results in a run time of about 30 minutes!). There are two supplementary scripts that complement the function and check the output, namely *SimMultiProbit.m* and *TestIndicatorCov.m*. The first of these produces simulations of the model Y, finds the sample autocorrelation function using *sampleacf.m* and plots the result. The second is a function which finds the autocovariance of the model when the exact indicator is used as the link function. The autocorrelations arising from use of the exact indicator and the simulations of Y should match that of the smooth approximation of the link function closely.

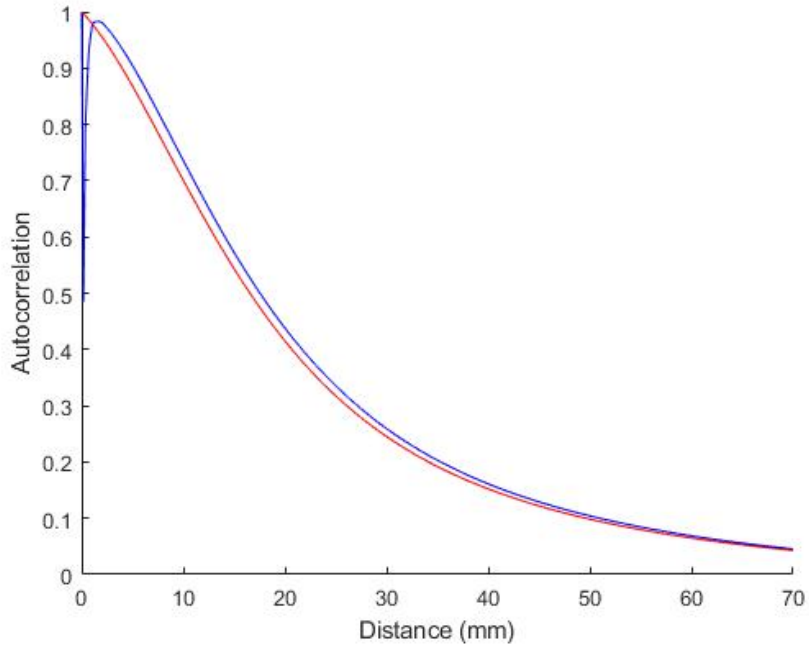


Figure 8: Smooth Indicator Model Approx. ACF (blue)/Exact Indicator Model ACF (red)

These autocorrelations relying on Matérn processes remain above zero for all lags. This is an issue as it will be seen in the next section that the sample autocorrelation is negative within this range of lags. Possible solutions to this issue are discussed later. There is also a strange dip in the blue curve present possibly as a result of error in numerical integration.

4.3 Sample Autocorrelation and Power Densities

The model autocorrelation needs to be fitted to the shape of the sample autocorrelation. *PermDataACF.m* is a script that obtains independent realisations of gaps along subsets from the data, then finds the average sample autocorrelation and its confidence interval to plot alongside the true model autocorrelation. The more realisations considered, the more reliable the average sample autocorrelation will be and the tighter the confidence intervals containing it, so this script takes 816 realisations. From [1], it is given that data at least 100mm apart can be assumed to be independent. Consequently, for the odd layers in the preform, the permeabilities of 17 gaps are obtained from subsets (rows) 1,3 and 5. For the even layers, permeabilities of 17 gaps are obtained from subsets (columns) 1,4,7 and 10.

The average sample autocorrelation obtained from this script is plotted in black below within a 95% confidence interval (red dash lines show the limits) alongside some sample autocorrelations of the individual realisations (thinner lines).

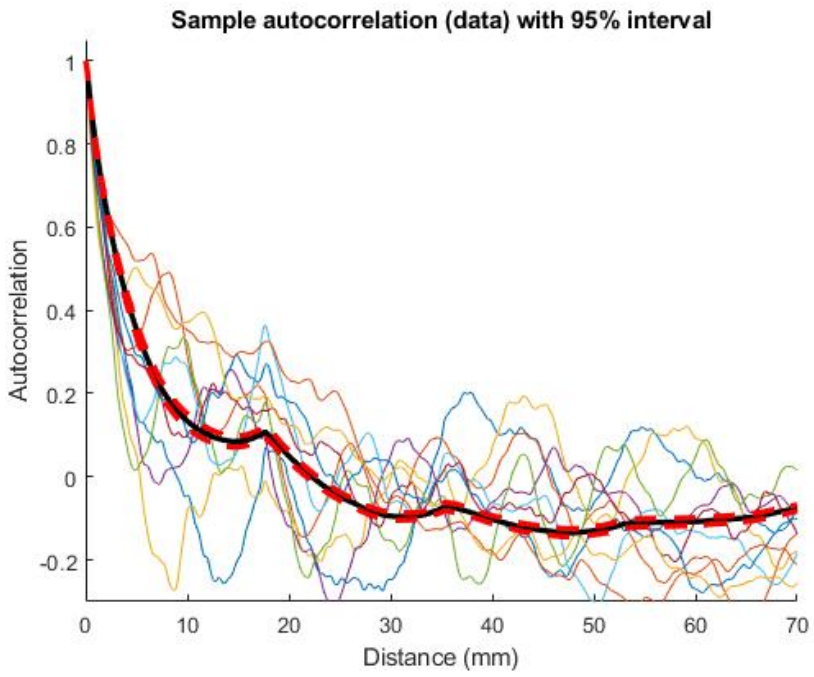


Figure 9: Average of sample autocorrelations with 95% confidence interval

The confidence interval is very tight due to the large number of realisations used for the average. The limits of the confidence interval were calculated by defining the autocorrelation at each fixed lag as a collection of 816 independent and identically distributed random variables and using the central limit theorem, giving:

$$P(t_{0.025,815} \leq \frac{S_n - n\mu}{\frac{s}{\sqrt{n}}} \leq t_{0.975,815}) = 0.95$$

Here, $t_{\alpha,\nu}$ is the critical value of the student's t-distribution with ν degrees of freedom, S_n is the sum of all the autocorrelations, n is the number of variables (816) and s is the sample variance. The inequality can be rearranged for the mean μ to obtain the limits of the confidence interval. Within this rearrangement there is a division by n, so it's clear that if n is large these limits enclose the true mean μ tightly. The width of the interval increases over the distance.

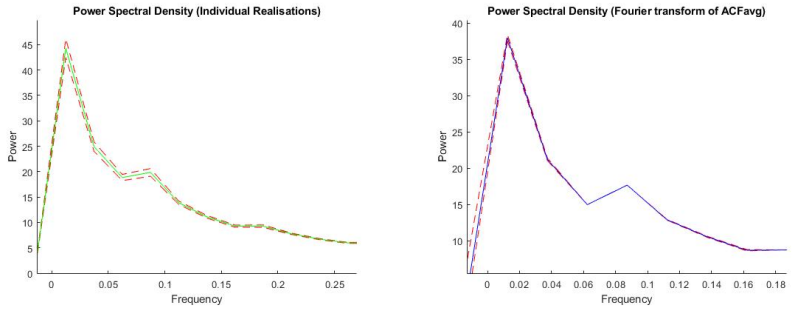
One method that could be used to fit a model to the data is to trial different functions for the covariance of the X_1 and X_2 processes and figure out which can give us the shape of the sample autocorrelation above. If a visually promising model is found, then simulations of the process can be ran and a test can be conducted (e.g. Mann-Whitney U-test) on the null hypothesis that the permeability data comes from the same distribution. The aim of this section is to find a visually promising model.

The sample autocorrelation is a strange object. It is below zero beyond a distance of 20mm and has 3 peaks of diminishing amplitude evenly spread across the interval at 17.65, 35.3 and 52.95mm. The first of these aspects cannot be achieved by using the Matérn function described in the last section as this function leads to an autocorrelation that is strictly positive and exponentially decaying towards zero. Therefore, other forms of the covariance function need to be considered.

One way that negative autocorrelation can be achieved is by constructing the covariance as a product of a decay term and an oscillatory term. The decay term is strictly positive and controls how quickly the autocorrelation decays towards zero, whilst the oscillation term provides the possibility of obtaining negative autocorrelation. The product is also multiplied by the variance of the process, which in our case is 1. Details and examples of functions that might be used for each of these terms can be found in the article by Hongyi Xu [4]. Initially, a function of the following form is considered for the covariances of the X_1 and X_2 processes:

$$E[X(0)X(x)] = \frac{1}{(1 + x/\theta)^k} * cos(x/l)$$

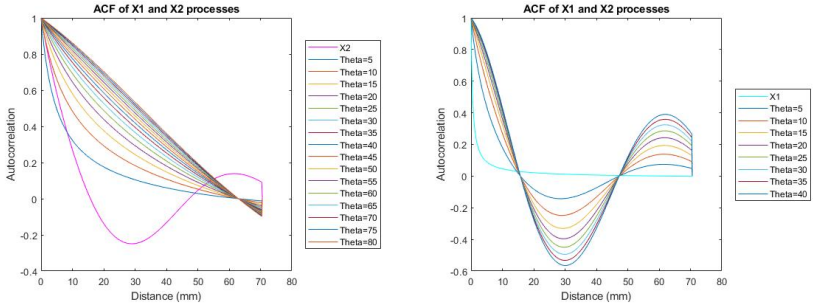
Here θ is a free parameter whilst x is distance and l is the correlation length. The correlation lengths come from the power spectral density, which is found (via the Wiener-Khinchin theorem) by taking the Fourier transform of the autocorrelation and centering it about zero. It wasn't clear how exactly this was done in [1], so I produced spectral densities via two methods. In the first, the individual realisations (such as gap 1, subset 1, layer 3) are transformed and then the average of these individual densities is taken. This has the advantage that a confidence interval can be constructed by the central limit theorem in the same way as for the average autocorrelation. The second method I used was directly applying the Fourier transform to the average autocorrelation and the upper/lower limits of it's confidence interval. This led to a shape more similar to the power spectral density that can be seen in figure 6. However, both of these methods produce peaks at the same wavelengths (40.099 and 10.025mm) so either might be considered.



(a) Individual Realisations (b) Average Autocorrelation

Figure 10: PSDs for VE length=5 pixels

To begin the investigation, k is initially set equal to 1. The effects of changing θ in each of the processes was tested by fixing the value in one of the processes whilst testing many values for the other. The two covariances for X_1 and X_2 are plotted individually, which will help later as the autocorrelation is simply a linear combination of the two curves. If the combined behaviour of these covariance functions resembles the data, the model should be close visually.



(a) Changing theta for X1 (b) Changing theta for X2

Figure 11: Covariance Functions of X1 and X2

For X_1, smaller values of theta create L-type shapes which decay to zero very quickly then move from tiny positive to tiny negative values. Increasing θ shifts the curve into a more cosine-like shape with slower initial decay. Meanwhile, for X_2, lower values of θ lead to faster initial decay at low lags whilst larger values have slower initial decay and deeper minima. In contrast, the data decays extremely quickly at lower lags then grows steadily past it's minimum. As a result, since any one covariance function of this form cannot obtain both of these characteristics (as they are cosines), X1 must be responsible for the initial rate of decay whilst X2 is responsible for the depth of the

theoretical acf. In general, committing to either one of these properties (fast initial decay/eventual slow growth) sacrifices the other property, which is the first major sign that this model for the covariance function isn't suitable or requires another term. With l=10.025 for X2, the curve goes back above zero far too quickly, so some parameters must be changed.

The first parameter to change was the correlation length for X2. In the paper [2], the second correlation length was estimated to be 16.3mm. Substituting this value gives theoretical acf's that stay below zero between about 27mm and 70mm, which is closer to the data. On top of this, the distance corresponding to the minimum is near to the global minimum of the data autocorrelation. It's clear that this change has created a more appropriate curve, but unfortunately the initial rate of decay is too slow and at larger lags the rise is too sharp.

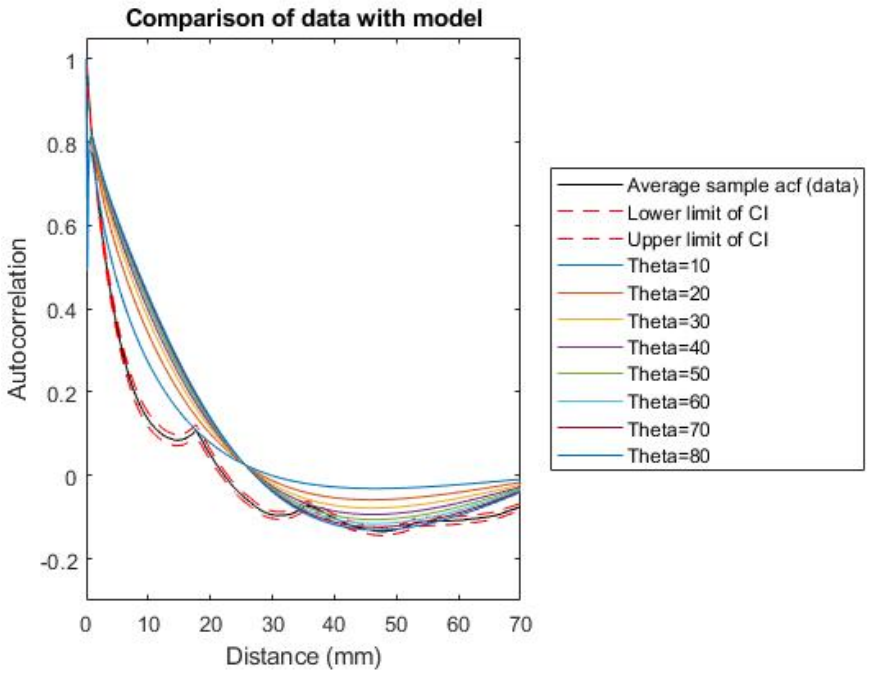


Figure 12: Example of comparison with X2 length 16.3

A number of efforts were made in order to correct this. First, theta was reduced in the covariance of X_1 to very small values (below 0.5), which gave an L-shape (X_1 covariance virtually zero for lags greater than zero). The behaviour of the theoretical acf for small lags (the dip thought to be due to numerical integration error) becomes more erroneous as a result, which is discussed in the next paragraph. Secondly, the order of the decay term (term

in brackets in the denominator) in the model was changed to -2. This had some benefit on the rate of decay of the theoretical acf and came closer to behaviour at larger lags. Trying order -4 also helped with the rate of decay but reduced the depth of the minimum. Finally, an extreme approach was taken with the order of the decay terms, making them much more negative in an attempt to obtain the desired rate. Doing this forced an increase in theta for both X1 and X2 by significant amounts, and again did not create a shape that fits.

Besides difficulties in obtaining the desired characteristics in the curve, there is a significant dip at the shortest two distances in the plot above. This might be due to error in numerical integration, giving a slightly misleading picture of what the model acf actually is, therefore the *TestIndicatorCov.m* function is used instead in what follows (here, the approximate indicator function has been replaced by an exact indicator).

As a last attempt at obtaining the desired rate of decay, an exponential of the form $exp(-r/\theta)$ was introduced replacing the inverse power in the decay term of each covariance function. Doing this benefits the rate of decay noticeably, for example choosing theta=10 for X_2 gives similar behaviour for distances between 0 and 10mm. However, this choice is detrimental at the larger lags, which is nowhere near the data.

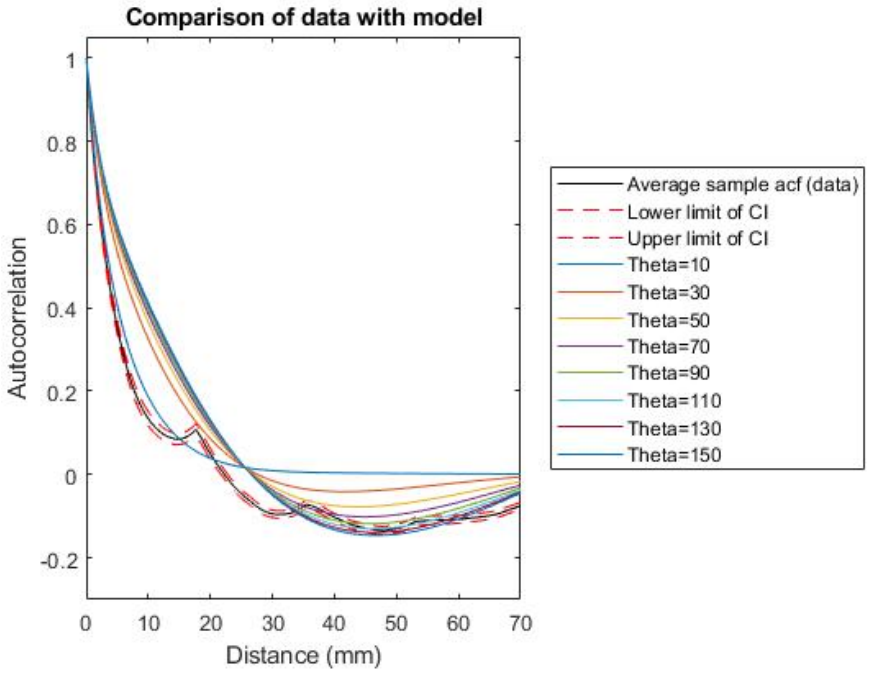


Figure 13: Example of comparison with exponential decay term

The current idea for covariance with a sole cosine as it's oscillatory term is insufficient both to create the desired rate of decay and the small peaks evenly spread throughout the data autocorrelation. Another term with more free parameters should be considered in addition to this cosine term. One candidate is a sine term which can be added to the covariance function in the following way:

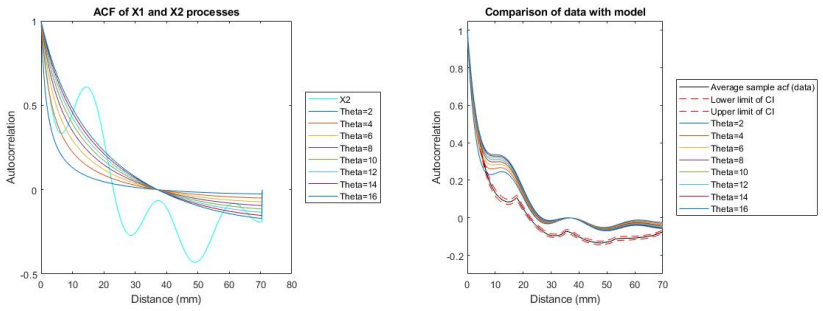
$$E[X(0)X(x)] = \frac{1}{(1+x/\theta)^k} * [cos(x/l) + a * sin(x/L)]$$

Here, a is another free parameter whilst L is a correlation length potentially not equal to that in the argument of the cosine term. k is assigned value one.

This time, instead of blindly attempting to fit the free parameters, the slope of the data near zero was calculated for comparison with the derivative of the covariance functions at zero. This was done in order to check that the rate of decay can definitely be obtained, otherwise an entirely different model should be considered. The slope of the data autocorrelation to it's first non-zero distance is -0.1549 which then decreased to -0.2639 between this and the next data point before beginning to gradually increase towards zero. The derivative of the covariance function defined above at distance zero is equal to $\frac{-1}{\theta} + \frac{a}{L}$.

It appears from this that the free parameters a and θ can be chosen to obtain the desired rate of decay here, as there is no restriction on a to be strictly positive (although in order for the covariance function to remain below 1 for all non-zero lags there is a restriction on the absolute value of a). Of course, it would be more helpful to take this analytical approach by differentiating the model autocorrelation altogether. However, there is a term arising from a bivariate normal CDF with a covariance matrix depending on the X_3 process which misbehaves at distance zero. The term is monotonically decreasing over the range of lags and so doesn't contribute to the creation or location of the small peaks and so it is sufficient to consider the covariance functions alone in order to form the small peaks present in the sample autocorrelation function.

Using positive values for a quickly proved difficult, as increasing it's value led to the covariance exceeding 1 without creating peaks of sufficient amplitude in the model autocorrelation. As a result negative values of a were tested which led to much better behaviour. Since sine starts from zero, using a negative value for a led to faster initial decay. A large enough absolute value permits small peaks to be formed along the distance of the model autocorrelation, as can be seen in the plot below where a=-0.5 for both processes, and L=3.8 for X_2 (with $L = l$ for X_1).



(a) Covariances of X1/X2

(b) Sine Model

Figure 14: Effect of including sine term

The model curve on the right image is still a long way off from the data curve below it. The peaks aren't as sharp and the oscillations dominate too much at larger lags. As a result, the sine term was squared in the hope of creating this sharpness near the peaks and reducing the effect of the oscillations at larger lags. This led to the plot below.

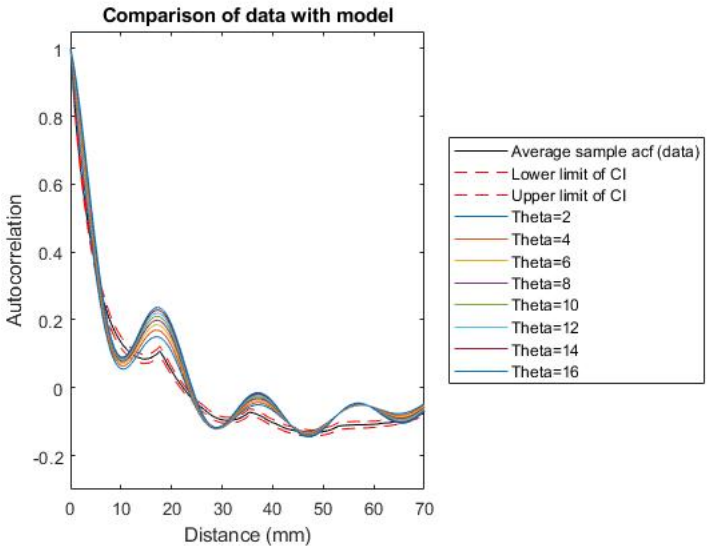


Figure 15: Effect of squaring sine term

Summarising this section, it's possible that covariance functions of this type could be appropriate for modelling the sample autocorrelation of the permeability data, but fitting purely visually may not be the best way to find a model that explains the data practically and with evidence from an engineering standpoint. With this in mind, more work needs to be done.

5 Conclusion

5.1 Results and Discussion

The purpose of this project was to account for the dependence between permeability data along a gap into a model. To meet this purpose, a model involving three Gaussian processes was introduced and analysed with emphasis on autocorrelation - an object which contains information on how the strength of correlation between two data points at a given distance (lag) changes across a gap. The idea was to create a model with an autocorrelation matching that of the data, then perform a simulation before testing alongside the sample data to see if there is any evidence that the distributions of the two samples match.

The main difficult choice in this analysis was how to choose the covariance functions of the two prominent Gaussian processes so that the autocorrelation generated can be negative and small sharp peaks can be formed across the range of distances. This eliminated Matérn processes, a popular choice

in literature, as they are strictly positive. Functions consisting of a product between a decay term (inverse power or exponential) and an oscillatory term (trigonometric) allowed these properties to be achieved, but introduced many more free parameters. A cosine term alone was insufficient to create the peaks, so a sine term was added alongside it with a scalar multiplier controlling its magnitude which finally gave a model autocorrelation visually resembling the sample autocorrelation.

The problem that remains from this analysis is how one might optimise the fit of the many free parameters. Besides this, the latent third process which underpins the model has properties that are mostly unknown. The fact that the marginal distribution is a mixture and the model is a linear combination creates complexity in terms of knowing which distribution is responsible for each data point, exacerbating both these issues.

5.2 Further work

To extend on the work discussed in this report, the autocorrelation of the model needs to be considered in conjunction with the power spectrum, as there is evidence to suggest the latter can be decomposed into an expansion of sine and cosine terms. On top of this, the third Gaussian process (X_3) needs to be investigated further - this process is key in order to understand the underlying mechanics of the model. This might involve adding trigonometric terms to the covariance function, which was considered for the other two Gaussian processes.

6 References

[1] Kong Jiawen. *Statistical analysis and stochastic modelling of permeability of composites* (MSc Dissertation 2017/18)
[2] Mikhail Y Matveev, Frank G Ball, I Arthur Jones, Andrew C Long, Peter J Schubel, and MV Tretyakov. *Uncertainty in geometry of fibre preforms manufactured with automated dry fibre placement and its effects on permeability.* (Journal of Composite Materials, 52(16):2255–2269, 2018)
[3] Laurence Shaw. *Automated Fibre Placement Project Report* (2019)
[4] Hongyi Xu. *Constructing Oscillating Function-Based Covariance Matrix to Allow Negative Correlations in Gaussian Random Field Models for Uncertainty Quantification* (2020)