NTNU – Trondheim
Norwegian University of
Science and Technology

Structural performance of parallel wires

Fredrik Øiahals Slungård
Trond Østen

Civil and Environmental Engineering (2 year)
Submission date: June 2015
Supervisor: Jochen Kohler, KT

Norwegian University of Science and Technology
Department of Structural Engineering

Department of Structural Engineering
Faculty of Engineering Science and Technology
NTNU – Norwegian University of Science and Technology

Master thesis 2015

Fredrik Øiahals Slungård and Trond Østen

Structural performance of parallel wire

1. Introduction

Steel cables are important components of major bridges. For the design of bridge cables the corresponding design value for the cable tension capacity is considered. The design value is evaluated by multiplying the characteristic strength by a partial safety factor. Hereby, the partial safety factor for the design resistance of bridge cables depends on four main aspect: i) the effect of the cable length; ii) the Daniels' effect; iii) the target reliability and iv) the damage allowance, i.e. how many wires failures are allowable.

To date, a framework that takes into account the above mention aspects for the calibration of the safety factor exists. However, the framework makes use of several simplifications. The simplifications relate in particular to the effect of deterioration on the cable capacity over time and to the considerations of system effects in the cable. This makes it by now impossible to consider the beneficial effects of deterioration protection and cable inspection and maintenance in the design process explicitly

2. Short description of the research project

The work to be performed in this master thesis consists of:

- Modelling probabilistically one (or more) aspects or phenomena influencing the cable capacity and reliability (e.g. initial strain due to installation, corrosion, fatigue etc.);
- Perform a specific literature review on these phenomena;
- Create a framework for updating the aforementioned models based on different information (e.g. initial wire tests, inspection during life etc.)
- Implement Matlab scripts (or other programming languages) for the models above.

The thesis must be written according to current requirements and submitted to the Department of Structural Engineering, NTNU, no later than June 10th, 2015.

NTNU, January 21th, 2015

Jochen Köhler

Preface

This master thesis concludes our two-year master's degree in civil engineering at the Norwegian University of Science and Technology (NTNU). It has been carried out during the spring semester of 2015, at the Department of Structural Engineering.

This thesis focus on a literature review on basic concepts of bridge cables, system effects that influence the cable capacity, the use of statistical methods for updating the parameters of the strength distribution and reliability. A framework containing all these effects made in MATLAB. Neither of the authors had much prior knowledge about steel cables, statistics or calculation of reliability. We therefore feel that we have challenged ourselves and learned a lot from this study.

We would like to thank our supervisor associated Professor Jochen Köhler and PhD candidate Michele Baravalle, for all the help and time they have used on us. They always directed us in the right direction and gave us guidance whenever we needed it.

This master thesis has been very educational and interesting and we are very happy that we chose to write a master thesis provided by the Department of Structural Engineering at NTNU.

Trondheim 10.06.2015

Trond Østen   Fredrik Slungård

Trond Østen   Fredrik Øiahals Slungård

Abstract

Many of the cable bridges in Norway, and Europe are in general nearing their estimated lifespan. Due to the sheer number of bridges, it is not possible to rebuild or redesign all of these. However, many are still in good condition, a scheme to verify their reliability are required. The cables make up the load bearing part in a cable bridge, failure of these cables leads to the failure of the entire bridge. One of the main concerns, especially regarding older bridges, is that the cables are subject to corrosion damage and the uncertainties related to the remaining reliability.

The main goal of this thesis is to create a framework in MATLAB that displays the system effects that influence the capacity of the cable and an updating scheme for the strength- and reliability distribution of the cable. The first part of the thesis focus on the properties of steel cables and system effects that influence the cable capacity. Daniels effect and length effect are commonly considered as system effects. Daniels effect take into account that the strength of a cable consisting of 150 wires and more tends towards a normal distributed of the cable strength. This leads to a capacity reduction between 6-8 %. Length effect takes into account that the number of links in a wire increase over time, which lead to a reduction factor around 8-10 % of the capacity. The effect of initial wire strain is shown to have an impact on the cable capacity and should therefore be taken as a reduction factor along the same lines as Daniels- and length effect.

The second part of the thesis focus on the updating of the strength and reliability distribution of the cable. An updated strength distribution of the cable is found with the use of Bayesian statistics. Baye’s method updates the strength distribution when additional information are implemented. This new information are gathered through use of proof or failure testing of the wires. The probability of failure are based on the updated strength distribution, which again are used to calculate the reliability of the cable.

Sammendrag

Mange av de norske og europeiske hengebroene ble bygget på slutten av 1960 tallet. Flere av disse nærmer seg nå sin estimerte livstiden uten at dette nødvendigvis har en direkte sammenheng med den eksisterende bærekapasiteten til broene. De sosioøkonomiske kostnadene gjør det umulig å bytte ut alle broer innenfor et kort tidsrom. En løsning er å oppdatere påliteligheten etter den faktiske styrken til broen i dag. Dette er midlertidig ikke noe som kan gjøres uten grundig analyser av tilstanden til både kablene og broen forøvrig. En av de største bekymringene, og usikkerhetene er faren for skader på kablene, siden disse utgjør bæresystemet for en hengebro. En slik usikkerhet er uakseptabel og bør ikke være et beslutningsgrunnlag, både på grunn av økonomiske og etiske hensyn.

Hovedmålet med denne oppgaven var å lage et rammeverk i MATLAB, som viser hvilke systemeffekter som påvirker kabelstyrken og en metode for å kalkulere påliteligheten til kabelen. Den første delen av oppgaven fokuserer på de grunnleggende egenskapene til en stålkabel og hvordan de forskjellige systemeffektene påvirker kapasiteten til kabelen. Hvis antall stålvaiere i kabelen overstiger 150-200 vaiere, blir styrken redusert med en faktor på 6-8 % som en følge av Daniels effekt. Lengde effekten tar hensyn til at en vaier blir inndelt i flere elementer, som følge av tidsavhengige effekter. Beregningene viser at styrken til en kabelen blir redusert med 8-10%. Underveis blir det vist at innledende tøyning i vaierne burde inkluderes i dimensjoneringen på lik linje med Daniels- og lengde effekt.

Del to av oppgaven forklarer hvordan fordelingen til styrken og påliteligheten av kabelen kan oppdateres. Oppdateringsprosessen bygger på bruk av Baye’s statistikk. Baye’s statistikk bygger på å kombinere tidligere informasjon, med ny informasjon for å finne den oppdaterte fordelingen. Den nye informasjonen blir implementert ved enten å være test- eller brudd-last av vaierne. Ved å implementere begge metodene i rammeverket, vil dette føre til en robust modell. Den oppdaterte styrke fordelingen brukes til å beregne sannsynligheten for svikt, som videre blir brukt i beregningene av påliteligheten. Ved å oppdatere påliteligheten til kabelen, kan broens estimerte livsløp forlenges, og store kostnader kan bli spart for samfunnet.

Table of content

Notations

P_f	Probability of failure
E	Modulus of elasticity
PDF	Probability density function
CDF	Cumulative density function
LHS	Latin Hypercube sampling
μ	Mean value
σ	Standard deviation
$E(X)$	Expected value
$Var(X)$	Variance
ε	lognormal variance
λ	Aging factor
L_ρ	Correlation length
b	Scale parameter (Weibull distribution)
c	Shape parameter (Weibull distribution)
$\Gamma(\cdot)$	Gamma distribution
E_n	Mean value
D_n	Variance
x_0	maximizing term
$L(\cdot)$	Likelihood function
$l(\cdot)$	log-likelihood function
β	Reliability

β_t	Target reliability
m_x	mean, destructive testing
s_x	standard deviation, destructive testing
θ	Parameters
$f'(\cdot)$	Prior PDF
$F'(\cdot)$	Prior CDF
$F''(\cdot)$	Posterior PDF
$F''(\cdot)$	Posterior CDF
q^*	Proof load
Log	Natural logarithm
‘	Prior information
‘’	Posterior information
ρ	Correlation

1 INTRODUCTION

Today people rarely view bridges as something spectacular, unless they break some kind of record or affect different landmarks in one way or the other. They simply see it as a tool made to simplify the travel from A to B. For engineers and designer the story is somewhat different. Questions concerning location, design, cost, safety etc. have to be considered, and answered satisfactory.

Bridges have always played an important part throughout history. Whether they were viewed as a way of making the transportation of goods easier and more effective or to be a display of power and technological superiority. The need to build bridges is nonetheless as vital today in ways of connecting remote islands and different parts of a country as it was a 100 years ago. The evolution of bridges reflects the type of technology at the given time. From the first heaps of stone across small streams and riverbeds in ancient times, to the arch bridges and aqueducts built by the romans, and the modern cable bridges built today.

1.1 GENERAL BACKGROUND

In countries like Norway, bridges play a vital role in securing the connection between remote islands and the mainland. While the focus on building new infrastructure played an important role during the late 1960s, many of the bridges built around this time have already reached or will soon reach their estimated lifetime. Replacing all the old bridges with new ones, will lead to immense economical cost. If a framework for assessing and updating the reliability of these bridges due to testing or inspection can give a more accurate value of the bridge remaining load bearing capacity and reliability, decisions can be taken based on more precise information.

1.2 OBJECTIVE OF STUDY

This study focus on the steel cables of cable bridges. The primary goal of this thesis is to create a framework for updating the capacity and the reliability of the steel cables. To reach the primary goals, some secondary goals were set:

- Present typical issues regarding steel cables (design layout, manufacturing process, protection methods etc.)

- Present and discuss several factors that can reduce the capacity of the cable (e.g. Daniels effect and initial strain.)
- Find an updating process for the strength distribution of the cable, based on different test methods

2 METHODLOGY

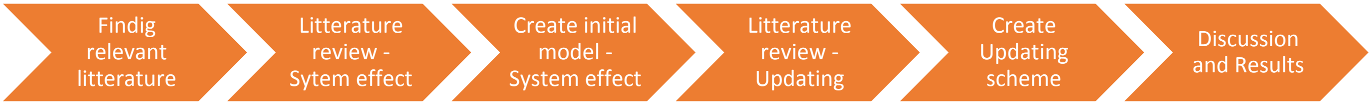
2.1 RESEARCH TYPE

The main purpose of this thesis is to create a working MATLAB script that updates the reliability of the wires/cables, based on new observation and information. The script will also include some system effects related to the capacity, such as Daniels effect and length effect. Fulfilling these purposes requires an evaluation of existing literature and extensive knowledge of the MATLAB program.

A quantitative research method is used in this report. Earlier publications concerning the effects mentioned above will be examined both quantitatively and deductively in order to establish the scope and reach of this thesis. By examining the literature deductively, it is possible to examine the conclusions of these earlier publications and verify if the calculations are correct. The verifications of these results will be done quantitatively by evaluating the original data from the publications, and create a verifiable statistical foundation that the updating scheme can be based upon.

2.2 RESEARCH METHOD

Initially, the work done by Faber et al. [1] will be examined to understand the basic concepts and calculations used in reliability assessment. By using BIBSYS scientific database (ORIA) and Google scholar searches, an extensive literature review will be conducted on the relevant topics. Finally, using MATLAB, an updating scheme based on the numerical observations found in Faber et al. [1] are created and discussed. The main focus of this paper is not the values themselves, but how the different parameters and system effects change the capacity and reliability of the cable.


Findig relevant litterature
Litterature review - Sytem effect
Create initial model - System effect
Litterature review - Updating
Create Updating scheme
Discussion and Results

2.3 REPORT STRUCTURE

The thesis consists of 11 chapters, a simple explanation of the content in each chapter is given below.

Chapter 1 contains a short introduction about the background and different goals of the thesis.

Chapter 2 explains the methods used to achieve the goals set in chapter 1.

Chapter 3 gives a short introduction to the basic terminology and history of cable bridges. It also explains how a cable is modelled.

Chapter 4 contains the methods and theory used in the estimations regarding the following system effects: Daniels effect and initial strain.

Chapter 5 is dedicated to the updating of the wires using both analytical and numerical solutions.

Chapter 6 calculates the changes in the partial safety factor and the reliability due to the updating scheme found in chapter 5.

Chapter 7 shows the results found using the framework created in MATLAB, from chapter 4-6

Chapter 8 discuss the results and assumptions made in the paper.

Chapter 9 concludes the work and recommends areas that should be further studied.

Chapter 10 includes the sources used in the paper.

Chapter 11 contains the list of appendixes.

3 A SHORT INTRODUCTION TO CABLES

This chapter is meant to familiarize the reader with the basic concepts and terminology regarding cable bridges. The following chapters explain the main difference between the main bridge types, different cable systems, basic protection schemes and how a cable is modelled using parallel and series systems. If the reader is already well versed in the basic concepts of bridge engineering, it is possible to skip chapters 3.1 – 3.3.

3.1 THE MAIN CABLE

The main cables make up the loadbearing part of the bridge, their integrity is therefore vital for both the safety and the load bearing capacity of the bridge. This chapter focuses on some of the main aspects on the modelling of the cables, different types of strands in the cables and ways of testing the strength of the cable.

Although modern cable bridges are a relative new construction form, they have existed for quite some time. Originally made up of vines, plank or even trees, they have transported people across rivers, streams and canyons for centuries. Despite several accounts of temporary cable bridges being built, the first permanent modern cable bridge was not built until 1823 in Geneva [2]. The use of cables instead of chains, gave rise to a new set of problems concerning the durability. At the time, there was no effective protection methods against corrosion of the thin wires in the cables. Many British engineers therefore favored the use of pin-connected iron chains instead of cables [2]. Despite these initial problems concerning corrosion, a number of bridges on the European mainland used cables instead of chains.

The first major bridge to make use of *air-spun* cables, i.e. “The air spinning method”, was the “Niagara Bridge”. Constructed between 1851 and 1855, using this new method the cables could be made *in situ* and spun between the anchoring points with the use of a spinning wheel [2]. In previous construction methods, the cables had to be uplifted from the ground, which was considered as a rather cumbersome and dangerous task. This new method allowed the spinning of several cables simultaneously and removed the need for transportation of large cables. Today, two in situ methods are mainly used: the *high*- and *low-tension method*. The high-tension method use counterweights to simulate the stress of a freely hanging wire when they are dragged between the anchoring points during the construction. One of the main advantages of this method is that each wire carries its own weight and the stresses in the wires are easily controlled during the erection process. *The low-tension* method on the other hand allows a lower stress in the wires, usually around 50 % of the self-weight [2]. Since the counterweight only carries about half of the self-weight, the remaining stresses have to be carried by the existing wires. This will in the beginning give rise to large strains in the existing wires, but as more wires are assembled these strains are reduced [2].

Although much more efficient than the original methods, there were still some drawbacks. The spinning process is both time-consuming and vulnerable to climate effects. The main concern using this method is the accumulation of water on the wires during the erection process. Naturally this is not favorable concerning corrosion later in the cables life. As a response to this problem, the pre-fabricated parallel wire strand method (PPWS) was developed. The wires are prefabricated in large bundles consisting of a 100 or more wires. The main advantage of this method lies in the use of a controlled environment that both significantly reduces the chance of water being accumulated inside the cable and the construction time [2].

For cable bridges, the load bearing capacity lies in its cables. Opposed to conventional bridges using pillars, the cables carry the load solemnly as a tension force transferred to the anchors and the pylons [2]. Cable bridges are categorized into three main categories as seen in Figure 1.

a) Suspension system
b) Fan system
c) Harp system

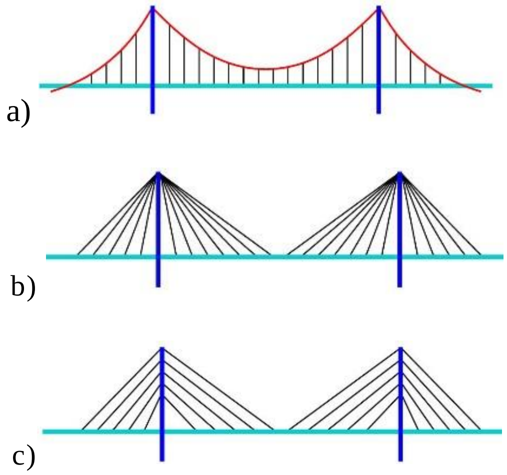


Figure 1 Cable bridges [3]

For suspension system, the load is transferred from the deck to the cables by vertical hangers connecting the cable to the deck. Suspension bridges usually consist of three spans, two side spans and one main span. The main cable is supported at four points; at the two anchoring blocks and at the two pylons. Well known suspension bridges using this system include "Hardangerbrua" and "George Washington bridge".

The fan system can be applied either as a *semi fan* system with several different anchoring points spread over a certain height of the pylon, or as a *pure fan* system with all cables at a single anchoring point. Normally the anchoring points of a semi fan system are spread over a relative small height of the pylon, this result in similar behavior of both semi- and a pure fan system. This again simplifies the preliminary analysis. The anchoring cable connecting the pylon top to the end supports are vital for the stability of the system, for the efficiency. It is therefore imperative that this cable is in tension for *all* load combinations [2].

The *harp system* will be an unstable system. Much like the *fan system*, the flexural stiffness of the deck and pylons must therefore be taken into account to achieve equilibrium. Unlike the fan system, the upper cable is stabilized by its connection to the end support. Here, the lower cable must be inspected to secure stability in the system [2].

3.2 STRANDS

As mentioned earlier the wires are bundled together forming individual *strands.* Depending on the type of strands, the number of wires vary from only a handful to several hundred. The wires usually have a diameter ranging from 5-7 mm made of high strength steel.

Helical strands consist of several layers of wire, where each successive layer is spun in the opposite direction of the helix. Due to the twisting effect, the modulus of elasticity decreases with a factor of 15 – 25 % and the design stress is chosen as $0.9f_{cbd}$, where f_{cbd} denotes the design stress of a straight wire [2]. When the twisting occurs it gives a contribution to elongation. Luckily this only happens when the strands are loaded for the first time, making the contribution to elongation irreversible. To make sure that there is no contribution to elongation at a later stage the cables are pre-tensioned with an overloading of 10 – 20 % [2]. When the cables twist, they also self-compact, which omits the need for bands or wraps to hold the strands together.

Locked-coil strands are made up of wires arranged in a normal helical strand where the outer layers are made of special wires with a Z-shape. Much like the helical strands, the locked-coil is self-compacting and more compact than any other types of strand. The strands are always prefabricated with both full length and cross section and delivered on sockets. The modulus of elasticity for locked-coil are slightly higher than the one found in helical strands, and results in a reduction of axial stiffness. Although the cross section has to be increased about 10% in order to compensate for the lower breaking strength, this only slightly increases the elongation compared to the parallel wire strands [2].

Parallel-wire strands do not suffer a reduction in breaking strength or stiffness due to twisting, however, the void ratio for pure parallel-wire strands are the highest of all the strand types [2]. Due to problems related to unacceptable stresses in the reeling process, other strands types were favored until these problems were solved in the 1960s. There are several types of sub-categories for parallel-wire strand, they are all similar in that wires are spun in a hexagonal pattern and bound together by bands or wraps. *New PWS cable* let the wire bundle twist itself in order to ease the unreeling; this also makes the cable self-compacting, without any reduction to the axial stiffness or tensile strength [2].

3.3 PROTECTION

The load capacity of a cable bridge relies solemnly on the integrity of the wires. Preventing deterioration of the wires has therefore always been a source of concern for engineers. The thin wires are extremely vulnerable to surface corrosion, and many of the wires will be inaccessible in later inspections. The standard procedure today is coating the wires with several layers of zinc and different products of polyester, glass fiber and acrylic resin. An outer layer consisting of plastic or steel then protects the load bearing cable [2].

A newer method introduces dehumidification of the cable by injecting dry air through the cable. In these cases the protectoral zinc layer can be omitted to secure air flow through the voids between the wires. Since changing the cables is costly, significantly amounts of research is based upon the effort of providing good enough protection against corrosion, fatigue etc. [2].

3.4 SYSTEM MODEL

The cables in modern bridges are usually modelled using two basic concepts; the parallel system and the series system. Cables consist of several hundred wires, and as long as the integrity of these wires are intact, the cable will not fail. Failure of the system occurs when *all* the wires break, which follows the simple reasoning: as long as one wire is intact, the cable still carries some load. It is important to note that failure of the wires can happen within a very small timespan, but they will break individually and not at the exact same time. This way of considering a system is called a *parallel system (*see Figure 2*)*.

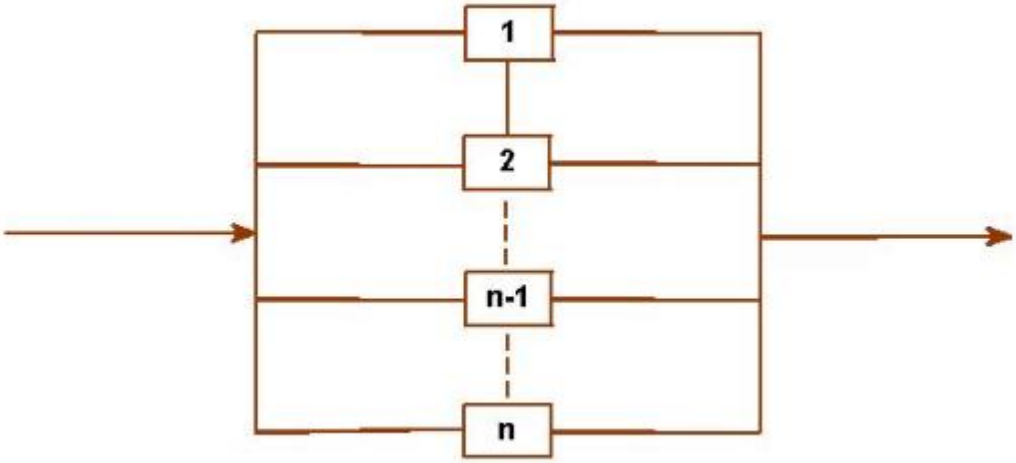


Figure 2 Parallel system

Since the probability of failure for a parallel system requires failure in all its members, the failure of the system is given as the intersection of the failure events of all wires. Being the intersection dependent on the correlation, first order bounds can be found considering full and no correlation [4].

$$\prod_{i=1}^{n} p_{fi} \leq P_f \leq \min\left[p_{fi} \right] \qquad (3.1)$$

Where p_{fi} is the probability of failure of the i-th wire. The left part assumes no correlation and the right part assumes full correlation, i.e. a high capacity in one wire, corresponds to a high capacity in the rest. It is worth noting that assuming full correlation is regarded as a conservative estimate, and for negative correlations the right part of equation (3.1) becomes zero [4].

The system above can be seen as an idealization of the cable. If all the wires were perfect with no defects of any kind, the system in Figure 2 would be a true representation of reality with each wire representing link 1, 2…, n. Defects are created both during the production and accumulated during the lifespan of the cable [4]. These defects are included by modelling each wire as a *series system*. The easiest way to explain such a system is considering a chain. If one of the links in a chain breaks, the entire chain breaks, it is the same way with the wires. If the wire breaks at one point, it will no longer retain its load bearing capacity. Although not entirely true with the modern cable types, where a broken wire regain some of its load bearing capacity, the main concept still holds.

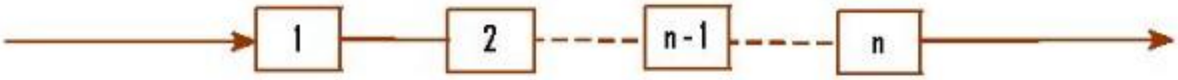


Figure 3 Series system

The number of elements in such a system depends on the strength correlation length; a shorter correlation length yields a higher number of elements in the series. For such a system, the probability of failure is equal to the union of the failure events of all elements [4]. The capacity of the system is equal to its weakest element, this is also known as the "weakest link effect" [4].

$$\max(P_{fi}) \leq P_f \leq 1 - \prod_{i=1}^{n}(1 - p_{fi}) \approx \sum_{i=1}^{n} p_{fi} \qquad (3.2)$$

4 CABLE CAPACITY

One of the main aspects of this thesis is the investigation of different system effects on a cable, the effects investigated in this chapter are: Daniels and length. The chapter consist of three main parts: The initial capacity are estimated using a Weibull distribution, the capacity reduction due to Daniels effect are investigated and finally, the reduction in capacity when wires are subjected to initial strain. All numerical values for prior and additional information are taken from Appendix A - Numerical input values.

4.1 INITIAL INVESTIGATION OF THE CABLE CAPACITY

The initial strength of a wire is based on prior information given by test data provided by the manufacturer or performed in-situ. Based on this information, the strength distribution of the wires are found by fitting the data to a Weibull distribution. The parameters of the distribution are estimated using the method of Maximum likelihood. The theory presented in chapter 4.1 is implemented directly in the script given in Appendix D –MATLAB script.

4.1.1 WEIBULL DISTRIBUTION

The Weibull distribution is named after the Swedish engineer Waloddi Weibull who studied the strength of materials and their distributions. The general Weibull distribution is a 3-parameter distribution, depending on the three parameters *a, b* and *c* [5]. In this case it is possible to omit the location parameter, *a*, which reduce the distribution to a 2-parameter Weibull, from now on simply called the Weibull distribution. The probability density function and cumulative distribution function are given in equation (4.1) and (4.2), from now on denoted PDF and CDF [5].

$$f_{X|\mathrm{B,C}} = \frac{c}{b}\left(\frac{x}{b}\right)^{c-1} exp\left[-\left(\frac{x}{b}\right)^{c}\right] \qquad (4.1)$$

$$F_{X|\mathrm{B,C}} = 1 - exp\left[-\left(\frac{x}{b}\right)^{c}\right] \qquad (4.2)$$

Where *x* is the load level on the wire, *b* is the scale parameter and *c* is the shape parameter. The scale parameter, *b*, governs the height of the PDF. Reducing the shape parameter, *c*, compresses the distribution, i.e. reducing the width of the PDF as seen in Figure 5. Different books use different notations for the scale and shape parameters. Faber et al. [1] denotes these parameters as *u* and *k*. In this paper the notations *b* and *c* are used, since this is the notation used in most mathematical textbooks. The CDF and PDF of the Weibull distribution are plotted in Figure 4 and Figure 5. The effects of changes in the scale and shape parameters are seen here.

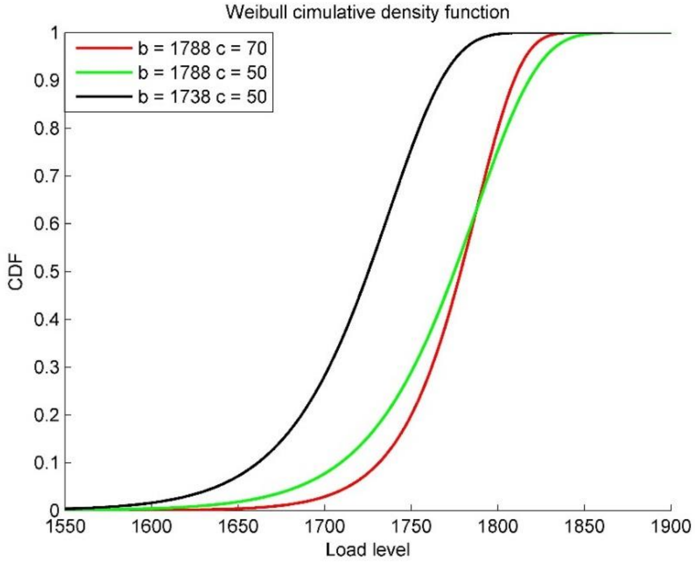


Figure 4 CDF of the Weibull distribution

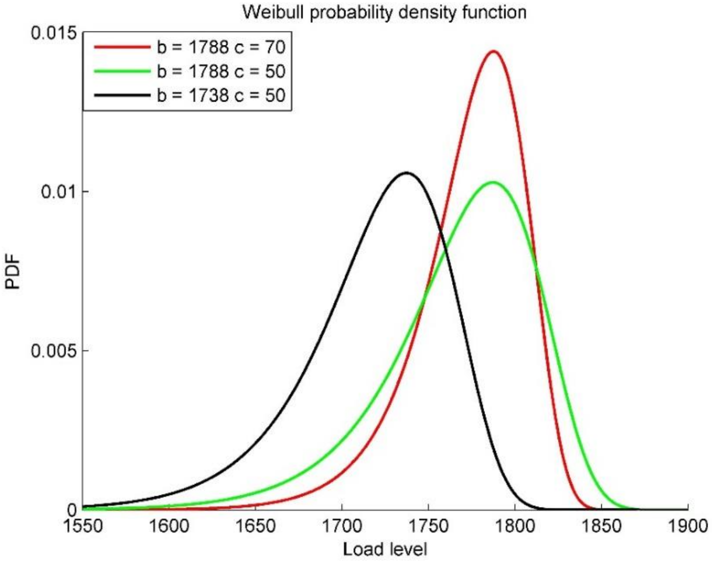


Figure 5 PDF of the Weibull distribution

Aging factor

λ is in this case seen as an aging factor of the wire, taking into account different defects due to initial cracking, corrosion fatigue etc.

$$\lambda = \frac{L}{L_\rho} = \frac{L}{lL_0} \quad (4.3)$$

L_p is the correlation length of the wire, L the length of wire, L_0 is the length of test specimen, and l represents the aging correlation factor. For new wires, the *correlation length* is equal to or larger than the actual length of the wire [1]. During the lifetime this length decreases as the wire accumulates more defects due to corrosion, fatigue cracks etc. As the wire reaches its expected lifespan, the correlation length could be reduced to a value around the diameter of the wire, e.g. 5-7 mm [1].

More specific, when the wire is new, λ has the value one to three, meaning that the wire is divided into one to three links. During its lifetime, the wire accumulates defects, which increase the value of λ over time. This results in an increasing number of links in the series system (see Figure 3). The series system will have uncorrelated and identically Weibull distributed links. The capacity of each link is represented by the Weibull CDF given in equation (4.2). The capacity of the system can now be derived by inserting the probability of failure for one link into equation (3.2). The CDF of the system is derived in equation (4.4).

$$F_X\left(x\right) = 1 - \prod_{i}^{\lambda}(1 - p_{fi}) = 1 - \left(1 - \left(1 - exp\left[-\lambda\left(\frac{x}{b}\right)^c\right]\right)\right)^\lambda = 1 - exp\left[-\lambda\left(\frac{x}{b}\right)^c\right] \qquad (4.4)$$

The PDF dependent on λ is found by differentiating equation (4.4) [1].

$$f_X\left(x\right) = \frac{\partial F(\mathrm{x})}{\partial x} = \frac{\lambda c}{b}\left(\frac{x_i}{b}\right)^{c-1} exp\left[\lambda\left(\frac{x_i}{b}\right)^c\right] \quad (4.5)$$

The mean value and the variance dependent on λ are given in equation (4.6) and (4.7) [1].

$$E\left(Z\right) = b\lambda^{-1/c}\Gamma\left(1 + \frac{1}{c}\right) \qquad (4.6)$$

$$V\left(Z\right) = b^2\lambda^{-2/k}\left[\Gamma\left(1 + \frac{2}{c}\right) - \Gamma^2\left(1 + \frac{1}{c}\right)\right] \qquad (4.7)$$

Where equation (4.4) and (4.5) represent the CDF and PDF respectively. x_i is the capacities for the different test specimen [5]. Γ is the gamma distribution given by the “Euler integral of the second kind” in equation (4.8) [5].

$$\Gamma(k) = \int_0^\infty e^{-v} v^{k-1} dv \qquad (4.8)$$

For more detailed information about the Gamma distribution, the authors refer to section 2.9 in Weibull: A handbook [5].

4.1.2 MAXIMUM LIKELIHOOD

The *maximum likelihood method (ML)* has proven to be both versatile and robust in the calculation of the *maximum likelihood estimations (MLE)*. Dating back as far as the 1700s it is now a widespread tool due to its ability to cope with all types of samples, independent of the samples distribution type. The key feature of MLE is that it estimates the uncertainties of the estimated parameters [5].

The concept of MLE is to find the vector $\widehat{\boldsymbol{\theta}}$ which maximimes the likelihood function $L(\theta \mid x_i)$, i.g. the highest chance of having realized the data (observations), where $\widehat{\boldsymbol{\theta}}$ are the parameters *b* and *c*. For a Weibull distribution, the vector $\widehat{\boldsymbol{\theta}}$ denotes the estimation of the shape and scale parameters of the distribution. In cases where the MLE cannot be given in a closed form the vector is the solution of a system of non-linear equations. When this is the case iterative methods have to be used [5].

The likelihood functions

The likelihood estimates is a function of the distributional parameters given in the vector $\boldsymbol{\theta}$ based on the sampled data. E.g. if the sample data consists of *N* independent elements, the likelihood is simply the product of *N* factors, termed *likelihood elements (L_i)* [5]. The independent likelihood function given in equation (4.9).

$$L(\theta \mid x_i) = \prod_{i=1}^{n} L_i(\theta) \quad (4.9)$$

In cases where the gradient method is used (partial derivatives), the *log-likelihood function* makes the process of performing these derivations easier [5]. As seen in equation (4.10) the log-likelihood function is now taken as the sum of the terms, not the product. This minimizes the error in the function; the error is now contained to one subsection of the sum, instead of being multiplied as it is in the likelihood function.

$$l(\theta) = \ln\left[L_i(\theta)\right] = \sum_{i=1}^{n} l_i(\theta) \qquad (4.10)$$

Maximizing the likelihood function

One of the methods used to find the set of values in the vector $\boldsymbol{\theta}$ that maximizes our likelihood function is the “Nelder-Mead simplex method”. The method utilizes a direct search method instead of a set of derivatives. “A *simplex* is the most elementary figure that can be formed in dimension N with N + 1 sides” [6]. A triangle in the 2D-plane and a pyramid with a triangular base in the 3D-space are both examples of simplexes.

The method operates by moving the simplex around the function until it surrounds the minimum before contracting the simplex until it reaches a predetermined acceptable error [6]. Modifying the method in such a way so that the function is maximized instead of minimized, is quite easily done by setting a negative sign in front of the log likelihood function. This is shown in equation (4.11).

$$\min_{\theta}\left[-l(\theta \mid x_i)\right] \; (4.11)$$

Uncertainties of Maximum likelihood estimates

Since the Maximum likelihood estimations themselves are an estimation, some uncertainties related to their true value exist. By invoking the the central limit theorem, i.e. when $n \to \infty$ the maximum likelihood estimations will be normally distributed with a mean that converge towards the parameters true value.

The covariance matrix yields the variance and the covariance of the parameters θ_i, as showed in equation (4.13). The square root of the diagonal elements denotes the standard deviation of the parameters. By inverting the Fisher information matrix, the covariance matrix is found. As seen in equation (4.12) the Fisher matrix is the double derivative of the log likelihood function, more easily explained, the Fisher matrix is derived using the following steps: If we derivate the *Jacobian* matrix, a well-known matrix in Finite Element Method analysis, we get the Hessian matrix. The information matrix is simply the negative of the Hessian matrix [7]. We can therefore conclude that the Fisher matrix is simply the negative derivation of the Jacobian.

The Weibull distribution yields a 2-dimentional multivariate normal, when $n \to \infty$. This means that the covariance matrix will be a 2x2 matrix.

$$\text{F} = \text{Fisher matrix} = \begin{pmatrix} -\dfrac{\partial^2 l(\theta \mid x)}{\partial \theta_i \partial \theta_i} & -\dfrac{\partial^2 l(\theta \mid x)}{\partial \theta_i \partial \theta_j} \\ -\dfrac{\partial^2 l(\theta \mid x)}{\partial \theta_j \partial \theta i} & -\dfrac{\partial^2 l(\theta \mid x)}{\partial \theta_j \partial \theta_j} \end{pmatrix} = \begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix} \qquad (4.12)$$

$$\text{C} = \text{Covariance matrix} = inv(F) = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \qquad (4.13)$$

4.1.3 SUMMARY

Now that both the Weibull distribution and the method of maximum likelihood are described, it is possible to calculate the strength of a wire. The mean strength of a wire is given by equation (4.6) which take into account the length effect. The values for *b* and *c* are found by using the method of maximum likelihood on the given tests data. The capacity of the entire cable can then be modeled as the sum of all the wires, where the strength is distributed using Weibull, as shown in equation (4.14).

$$\textit{Cable}\ \text{capacity} = \left(\sum_{i=1}^{n} b \lambda^{-1/\mathrm{c}} \Gamma\left(1 + \frac{1}{c}\right) \right) \frac{d^2 \pi}{4} \qquad (4.14)$$

The resulting cable capacity is shown as a function of λ in Figure 6.

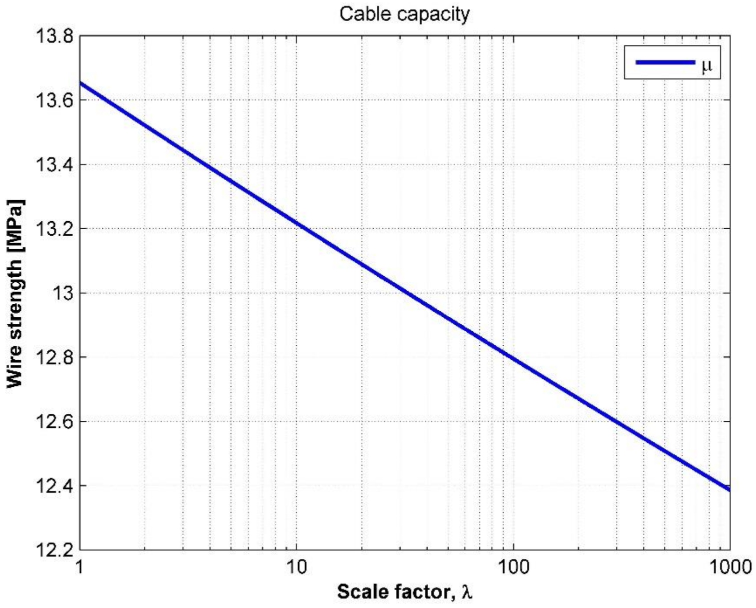


Figure 6 Cable capacity as the sum of wires

From Figure 6 it can be seen that the length effect will reduce the capacity with approximate 9.8% during the lifetime of the wire. In general, the reduction due to length effect is set to 10% [1].

4.2 DANIELS SIZE EFFECT

The method for calculating the capacity outlined in chapter 4.1 overestimates the cable capacity when the number of wires exceeds 150 [1]. By including Daniels size effect, the overestimation is accounted for. The main assumption in Daniels theory is that the strength of the wires are normally distributed [8], a short introduction to the normal distribution and its properties is therefore given below. This is implemented in a MATLAB script in Appendix D –MATLAB script.

4.2.1 NORMAL DISTRIBUTION

One of the most known and used distribution is the Normal, or Gaussian distribution. One of the main reasons for this is the so called *Central limit theorem* which states: If a set of random samples, n, is selected from a population the samples will be normally distributed as n increases without bounds [9]. Simply explained; if we have a finite number of sampling points with a given distribution, e.g. ten strength tests fitted to a Weibull distribution. As we add more tests, the shape will gradually change and resemble that of a normal distribution.

The normal PDF and CDF are given by the following equations:

$$PDF = f_x(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] (4.15)$$

$$CDF = F_X\left(x\right) = \int_{-\infty}^{x} f_X\left(x\right)dx \, (4.16)$$

In most practical cases the mean and variance has to be calculated using estimates based on a finite number of observation. The most used way of doing this is the so-called “method of moments” where μ is equal to the sample mean, and σ^2 is the sample variance [9].

$$E(X) = \mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (4.17)$$

$$Var\left(X\right) = \ \sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_i)^2 \quad (4.18)$$

4.2.2 DANIELS THEORY

Named after the mathematician who discovered it, its original purpose was to estimate the strength of wool fibers in the textile industry. Engineers have later adapted the theory to calculate the strength in cables with large bundles of wires [8]. A cable composed of a large number of wires is modelled as a parallel system with elements having identically distributed capacities. The capacity of a system tends to be normally distributed, when the number of wires are large. The main assumption in Daniels theory is that the probability that a wire breaks under a given load, *z*, is such that $1 - F_Z(z)$ zeros faster than $\frac{1}{z}$ [8]. More practically, Daniels effect will be valid if the number of wires are smaller than 150 [1]. In bridge cables the number of wires are relatively high. By invoking the central limit theorem it can be shown that the strength of the wires are normally distributed if we make the assumption that the number of wires, n → ∞ [10]. The mean and standard deviation of the normal distribution are given by Faber et al. [1].

$$E_n = [nx_0\left[1 - F_z\left(x_0\right)\right] + c_n] \qquad (4.19)$$

$$D_n = x_0\left[nF_z\left(x_0\right)\left(1 - F_z\left(x_0\right)\right)\right]^{1/2} \qquad (4.20)$$

$$c_n = 0.966n^{1/3}a \qquad (4.21)$$

$$a^3 = \frac{f_Z^2(x_0)x_0^4}{2\,f_Z(x_0) + x_0\,f_Z^{'}(x_0)} \qquad (4.22)$$

$$x_0 = max\left\{x\left(1 - F_Z\left(x\right)\right)\right\} \qquad (4.23)$$

Where F_Z is the Weibull CDF, c_n a correction term and x_0 a term that maximize the Weibull distribution, as seen in Figure 7.

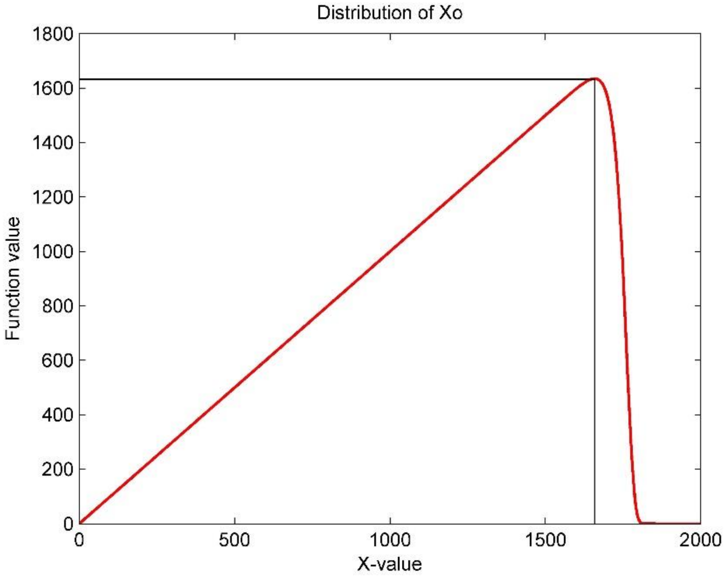


Figure 7 Graph showing how x₀ is found

Figure 7 shows how the value of *x0* found. The red curve is the function $x\left(1 - F_Z(z)\right)$, seen in equation (4.23). The value of *x* that maximize equation (4.23), is described by the black line. The value of the found *x0*, is then used to maximize the mean and standard deviation in equation (4.19) and (4.20) [10].

4.3 CAPACITY WITH INITIAL STRAIN

In this paper, the origin of initial wire strains are assumed to originate from the manufacturing process or during the erection phase. The initial strains are modelled as random numbers and the impact of the wire strains on the cable-capacity are investigated. The yield and failure strength of the wires, including Daniels, are assumed log-normally distributed and simulated using random numbers with a given mean, standard deviation and correlation. Due to the use of random numbers in the simulations for the initial wire strain and wire capacity, Monte Carlo simulations have to be performed to get an accurate picture of the actual cable capacity. The calculations of the cable capacity due to initial strains are performed in a MATLAB script presented in Appendix D –MATLAB script.

Strain

The yield strain is calculated using Hooke's law, which denotes the relationship between stress-strain using the modulus of elasticity. The general law is only valid when the material behaves elastically, i.e. the material goes back to its original form after deformation, and the relationship between stress-strain is linear [11]. The general formula of Hooke's law is:

$$\sigma = \mathrm{E}\varepsilon \qquad (4.24)$$

Where E is the modulus of elasticity, σ is the stress and ε is the strain [11].

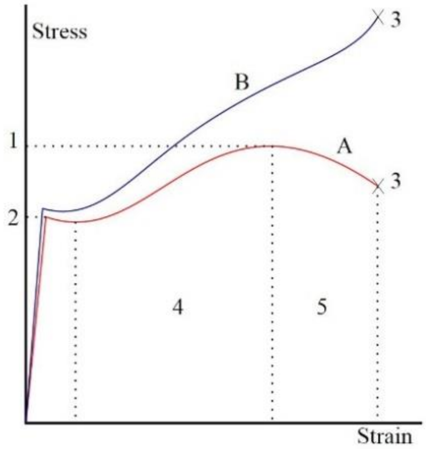


Figure 8 Stress-strain curve [12]

As mentioned above, the general Hooke's law is only valid for the elastic area of the stress-strain curve, up to *point 2* in Figure 8. Point two is the *yield strength* of the material. After this, the relationship between stress-strain is no longer linear. The behavior of a material is highly dependent on whether it is categorized as a brittle or ductile material. A brittle material (e.g. carbon) will have a small plastic elongation zone before breaking, simply put; failure occurs suddenly, with very small deformations beforehand [11]. A ductile material, like steel, will have large deformations before failing and displays a long plastic elongation zone [11].

The general Hooke's law can be modified, so that the stress can be calculated over the entire stress-strain curve. A modified stress strain relationship is shown in equation (4.25) [1].

$$\sigma_i(\varepsilon_i) = \begin{cases} \varepsilon_i E & \varepsilon_i \le \varepsilon_{y,i} \\ \varepsilon_{y,i} E & \varepsilon_{y,i} < \varepsilon_i < \varepsilon_{u,i} \\ 0 & \varepsilon_i \ge \varepsilon_{u,i} \end{cases} \qquad (4.25)$$

Where ε_i, ε_y and ε_u is acting strain, yield strain limit and failure strain limit. The total stress in the cable can then be found using equation (4.26) [1].

$$\sigma = \frac{1}{N_G} \sum_{i=1}^{n} \sigma_i(\varepsilon_i) \qquad (4.26)$$

N_G takes into account that the wires can come from different batches. For small cables, it can be assumed that all the wires are from the same batch.

Initial wire strain

Initial wire strain is implemented in the calculations by assuming that the initial wire strain can be represented as uniformly distributed random numbers over a certain domain. The strain used to calculate the stress in each wire in equation (4.25) can then be found by using equation (4.27), which takes into account the initial strain (ε_0) in the wires.

$$\varepsilon = \varepsilon_0 + \varepsilon_i \qquad (4.27)$$

The stress in the cable can then be found by using equation (4.26).

4.3.1 *RANDOM NUMBERS*

The initial strains are modelled using random numbers, these random number generators are divided into two main categories.

True random number generators (TRNG) are generated by observing natural phenomena, such as atmospheric noise, occurrence of lightning etc. However, due to the nature of these measurements, gathering enough information (numbers) to use in the simulations can take several months or even years [13]. Modern computer programs therefore use so-called *pseudo random numbers generators (PRNG)*. These are not random in the sense that they can be reproduced knowing the *seed* of the sequence. Although this may initially seem like a bad thing, the return period of good PRNGs is so long that for all practical purposes they may be considered random[14]. Unlike TRNGs, pseudo random numbers are deterministic, and is therefore well suited for simulations where the results needs to be reproduced later [14].

As seen in chapter 3.4, the wires are modelled as a parallel system, making the assumption that the stresses are equally distributed between the wires. The initial strains are therefore modelled using a uniformly random numbers, as seen in Figure 9. In a uniform distribution the numbers have equal density, meaning that there is no single value that tends to cluster, and the numbers are equally distributed over the given domain as seen in Figure 9.

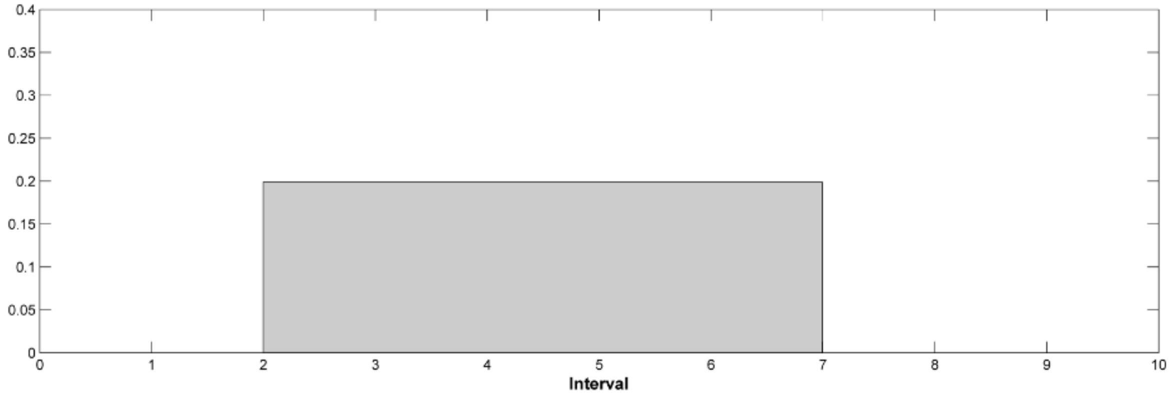


Figure 9 Uniformly distributed random numbers

4.3.2 CORRELATION BETWEEN YIELD AND FAILURE STRAIN

The correlation between two variables, e.g. between yield and failure strains shows the *tendency* two variables have of varying together. A positive correlation indicates that a higher value than the mean of *X* relates to a higher value of *Y*, and vice versa.

Correlation (4.28) indicates how two variables are connected. The value varies in [-1 1], where $\rho = 1$, denotes full correlation, e.g. if the variable $X = 1$, then the variable $Y = 1$. For a *linear* correlation, a positive covariance indicates that the maxima of two variables is in the same interval, while a negative covariance indicates that the maxima of one variable corresponds with the minima of the other [15].

$$\rho = \frac{E[(X - E[X])(Y - E[Y])]}{\sigma_X \sigma_Y} \qquad (4.28)$$

A linear correlation implies that changing one variable affects the other. However, this is not always the case; *Spurious correlation* can display a strong correlation, e.g. ρ = 1, despite having no real connection between the variables. Even though changing one, will have a mathematical effect on the other, they are not connected in any real sense [1, 16].

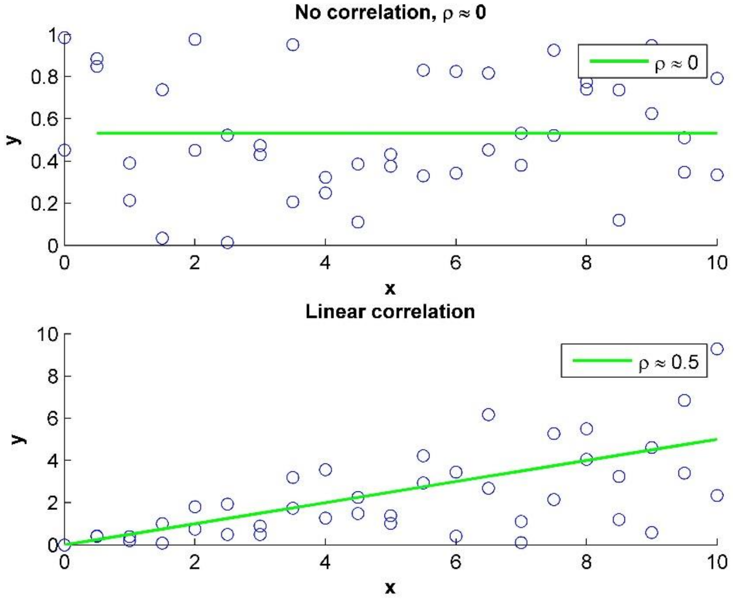


Figure 10 Correlation

In many cases, we wish to simulate a system using random numbers, e.g. initial strain in wires. In such a case, the correlation is usually already given and the random variables are initially uncorrelated. Correlated values may be found by rotating the coordinate system using the following equations [17].

$$x' = x \cdot \cos(\alpha) + y \cdot \sin(\alpha) \qquad (4.29)$$

$$y' = -x \cdot \sin(\alpha) + y \cdot \cos(\alpha) \qquad (4.30)$$

$$\rho = \cos(\alpha) \qquad (4.31)$$

$$x' = \mathrm{x} \cdot \rho + y \cdot \sqrt{1 - \rho^2} \qquad (4.32)$$

$$y' = -x \cdot \sqrt{1-\rho^2} + y \cdot \rho \qquad (4.33)$$

Where [*x y*] are the initially uncorrelated random numbers and equation (4.32) and (4.33) yields the correlated pair of random numbers [x x'] and [*y y*'].

Correlation between yield strain and failure strain

The correlation factor between yield- and failure strain is not given directly in model codes. The correlation factor can be found by looking at the correlation between yield stress and failure strain, because the correlation between the modulus of elasticity and both yield stress and failure stress are zero [15]. This makes it possible to use the correlation between yield stress and failure stress directly by assuming that the plastic elongation zone is flat. The correlation between failure strain and yield strain can then be taken directly from the JCSS model code as the correlation between yield stress and failure strain [15]. As seen from Table 1, the correlation factor is -0.45.

	f_y	f_u	E	ν	ε_u
f_y	1.0	0.75	0	0	–0.45
f_u	0.75	1.0	0	0	–0.60
E	0	0	1.0	0	0
ν	0	0	0	1.0	0
ε_u	–0.45	–0.60	0	0	1.0

Table 1 Correlation Matrix [15]

Now that the correlation between yield- and failure strain is known, two correlated values are found using equation (4.32). The correlated values will now have a large scatter, and since the inverse lognormal distribution is used to calculate the values for yield and failure they have to be in the domain zero to one. This is done by taking the normal CDF with the found correlated values, mean equal to zero and the standard deviation equal to one as arguments.

4.3.3 LOGNORMAL DISTRIBUTION

To find the stress in equation (4.25) the values for yield- and failure strain are needed. When the assumption that Daniel's effect is valid, the distribution for yield strain and failure strain can be modeled as a lognormal distribution. This is due to that the lognormal distribution is defined over the positive real space [1]. The distribution is therefore immensely useful when describing natural variables such as the strength of wires. By definition, *x* is log normally distributed if $y = \ln(x)$ is normally distributed.

The result are similar CDF and PDF for the normal and lognormal distributions [18].

$$f_X(x) = \frac{1}{x\varepsilon\sqrt{2\pi}} exp\left[-\frac{(\ln(x)-\lambda)^2}{2\varepsilon^2}\right] \qquad (4.34)$$

$$F_X(x) = \int_{-\infty}^{x} f_X(x)\,dx \qquad (4.35)$$

Where ε and λ are the mean and variance of $y = \ln(x)$ [18].

$$\lambda = E\left[\ln(x)\right] = \mu_{\ln(x)} \qquad (4.36)$$

$$\varepsilon^2 = \sigma^2_{\ln(X)} \qquad (4.37)$$

Hence, the moments can be found [18]

$$E(X) = \mu_X = \exp\left(\lambda + 0.5\varepsilon^2\right) \qquad (4.38)$$

$$std(X) = \sigma_X = \sqrt{\mu_X^2\left(\exp\left(\varepsilon^2\right)-1\right)} \qquad (4.39)$$

The correlated values for the yield and failure strain is then found using the inverted lognormal distribution, with the mean and standard deviation found in equation (4.38) and (4.39).

4.3.4 MONTE CARLO SIMULATIONS

Since only a certain number of random numbers are generated, the values for yield- and failure strain are only approximated values. To be certain that the found values are correct, this process should be simulated until the value converge. This can be done using Monte Carlo simulations. Monte Carlo is one of the basic and most used methods for calculating probability of failures and probability density functions [14]. Much because the method in itself is very simple and easy to understand, it can handle all kind of distributions, limit state functions and is correct at the limit. The method had little use in the past, because the simulations process is very demanding for the computer, but with today’s modern computer’s it has become more used. Monte Carlo simulation works by producing a large amount of uniformly distributed pseudo-random numbers between the values zero and one.

For each simulation, each basic variable is sampled and the limit state function evaluated. Sampling is performed by evaluating the inverse normal CDF, with the random generated number as an argument. The probability of failure is then given in equation (4.40).

$$p_f = \frac{N_0}{N} \qquad (4.40)$$

Where N_0 is number of failures and N is the total number of simulations [4].

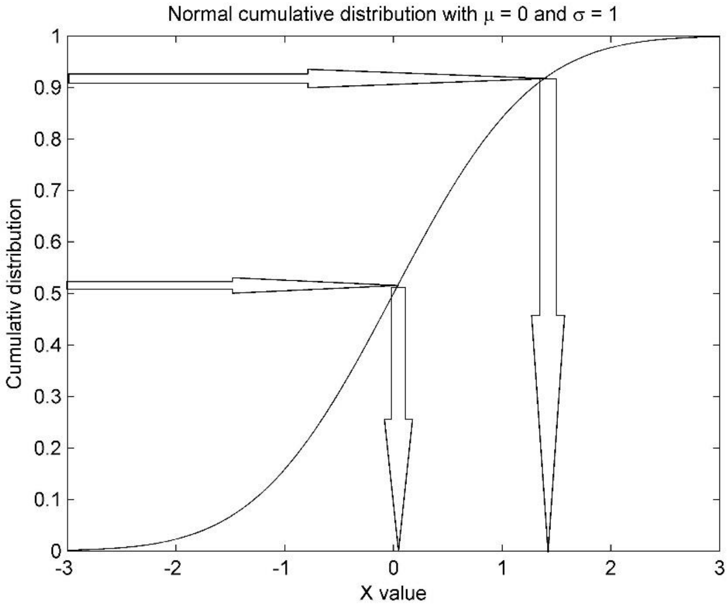


Figure 11 Normal cumulative distribution for Monte Carlo

One of the main drawbacks with crude Monte Carlo sampling is that it often displays a slow convergence rate, especially for rare events such as the failure of a wire. For complex simulations, this is a time consuming operation. By introducing different *variance reduction techniques* the number of simulations needed are significantly reduced, by making the operation more efficient [19]. The technique used in this paper is called *Latin hypercube sampling (LHS)* and is a subgroup of the *Importance sampling method*.

Latin hypercube sampling

Whereas many of the existing sampling methods are deterministic, one of the advantages of LHS is that the process is random [3]. The method samples the points from the MC simulation in an *n*-

dimensional hypercube so that “only the marginal distribution is stratified” [3]. Perhaps a bit technical, the process is explained by Huynh et al.[19] with the following three steps.

1) Divide each dimension of our space in N sections.
2) Generate a discrete random variable uniformly distributed in the section.
3) Repeat the previous steps for all the sections.

These steps are explained in the following example: The *stratified* sample vectors [Θ_1 Θ_2] is found by choosing a random number in each interval, u_i and dividing the Y-axis into N sections Figure 12 has five such intervals, where e.g. interval 1 has a range of 4 – 5. The randomly selected value u_1 is now in the interval between 4 and 5, u_2 between 3 – 4 etc.

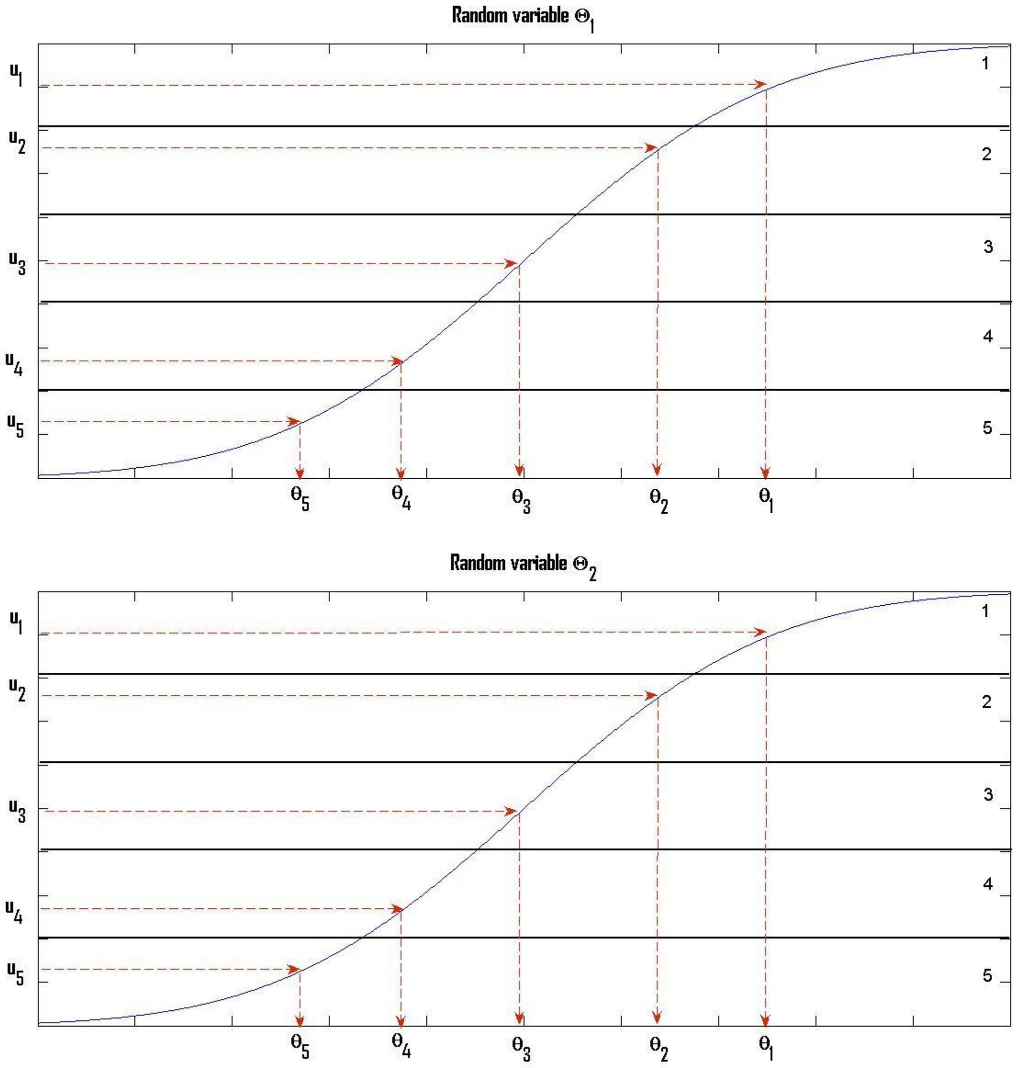


Figure 12 Stratification of two random variables

The stratified samples are now presented in column 4 in Table 2. The values in the table does not represent any real life data, but is just a simple example intended to show the main aspects of the Latin hypercube. The following assumptions is made in Figure 12:

1) Y – axis range, [0 1] divided into 5 equal intervals
2) X – axis range, [a b] = [0 10]
3) Create the random number $u_i = U(\min, \max)$
4) $\theta_i = (U(0,1)\cdot(b-a))+a$
5) Randomize the order that the values of *θi* are inserted into the Latin Hypercube

By creating a new set of random numbers in columns 6 (Shuffle), without considering the different stratum intervals. These random numbers decide the *rank* the different numbers in column 5 will have, the highest shuffle number have rank 1, and the lowest rank 5. By doing so, the order in which the numbers are inserted into the Latin Hypercube are randomized in the sense that the lowest value of *Θ1* does not correspond to the lowest value of *Θ2*.

	Strat	**Min (value)**	**Max (value)**	**θ_i**	**Shuffle**	**rank**
Θ_1	**1**	**0.40**	**0.50**	**0.45**	**0.98**	**1**
	2	**0.30**	**0.40**	**0.38**	**0.65**	**3**
	3	**0.20**	**0.30**	**0.25**	**0.12**	**5**
	4	**0.10**	**0.20**	**0.14**	**0.87**	**2**
	5	**0.00**	**0.10**	**0.06**	**0.55**	**4**
Θ_2	**1**	**0.40**	**0.50**	**0.42**	**0.36**	**4**
	2	**0.30**	**0.40**	**0.33**	**0.86**	**2**
	3	**0.20**	**0.30**	**0.21**	**0.23**	**5**
	4	**0.10**	**0.20**	**0.17**	**0.99**	**1**
	5	**0.00**	**0.10**	**0.05**	**0.47**	**3**

Table 2 Calculating the Rank used in the Latin Hypercube matrix

Inserting the values of the two variables by their respective rank yields the Latin Hypercube matrix in Table 3.

Variables **Rank**	Θ_1	Θ_2
1	**0.45**	**0.17**
2	**0.14**	**0.33**
3	**0.38**	**0.05**
4	**0.06**	**0.42**
5	**0.25**	**0.21**

Table 3 Latin Hypercube matrix

As seen in Table 3, there is no immediate correlation between the values, meaning that a higher value of Θ_1 does not correspond with a higher value of Θ_2. To get correlated values of the Latin hypercube matrix, the method shown in chapter 4.3.2 must be implemented. It is important to note again that these values does not represent any real values or simulation, such as the ones shown in Figure 13 and Figure 14 below.

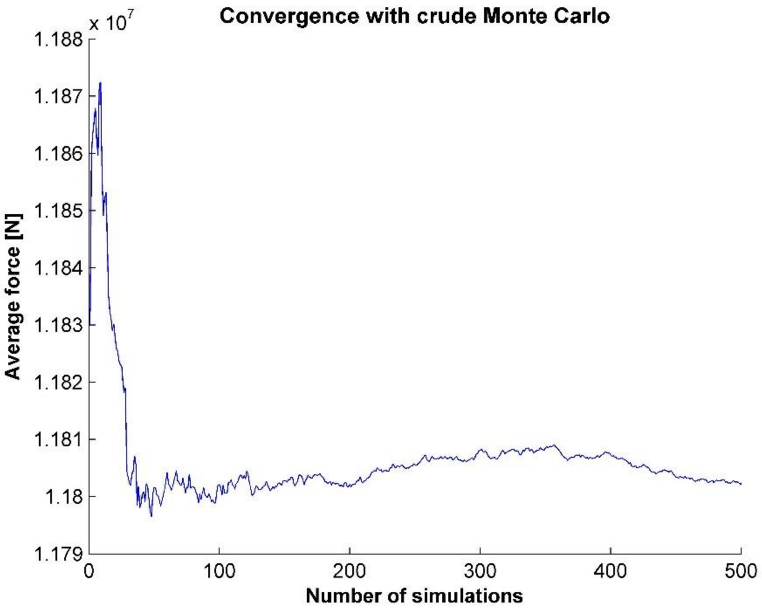


Figure 13 Convergence of Crude Monte Carlo

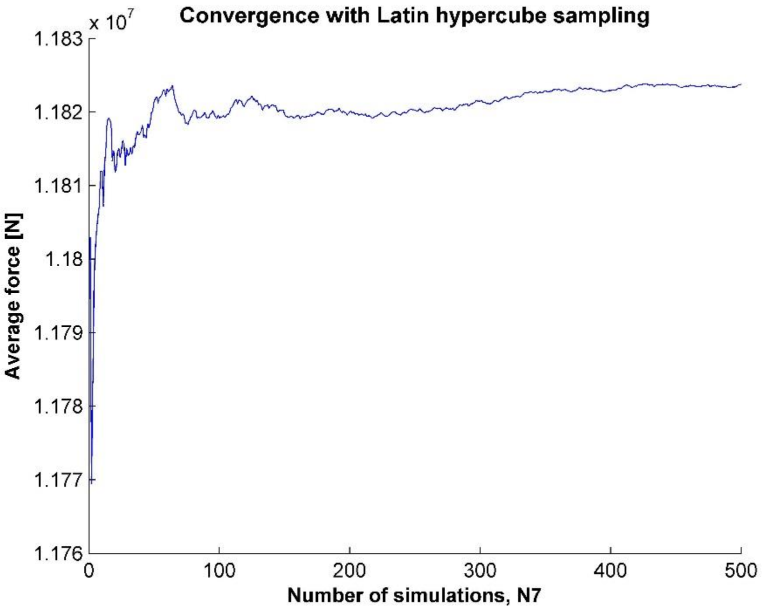


Figure 14 Convergence of Monte Carlo with Latin hypercube sampling

As seen in Figure 13 and Figure 14, implementing the LHS greatly reduces the number of simulations required. As an effect, the standard deviation will only be approximately 1/10 compared to the one found using crude Monte Carlo simulation [20].

5 UPDATING

The main scope of this thesis is the updating of wire strength based on additional information. This chapter contains the different aspects in this process. The first two chapters contain a short introduction regarding the two main aspects of statistical thinking and the different test methods used to gather new information. Chapter 5.3 and 5.4 is a more hands-on approach to the actual updating, and compares and explains the difference between choosing an analytical and a numerical approach in the updating scheme. The last chapters contain the different integration methods used in MATLAB and the updated fitted distribution.

5.1 CLASSICAL OR BAYESIAN STATISTICS

In the world of statistics, there are three ways of thinking. Two of the cases are considered classical, and the third is the Bayesian approach. Classical statistic use the principle of indifference, this is more simply put that all possible events is assigned the same probability. Commonly used examples is a coin toss or a dice toss where all outcomes has the same chance of occurring, and the chances of observing the next outcome is not based on the previous. I.e. the parameters are given a fixed value, in classical statistics the prior information is therefore considered as an unknown constant [21].

Bayesian statistic will summarize all the known information of the unknown parameter(s). To find the best estimations of the parameter(s), data is used from prior information and additional tests combined to find an updated estimation. This estimation will take into account the uncertainties of the given data [22]. The possibility of combining earlier information with new ones offer a great advantage compared to the classical approach.

5.2 TEST METHODS

Updating of the reliability for existing structures are usually done by performing different tests of vital load bearing components. The two test methods that are considered are *destructive testing* and *non-destructive testing*, also denoted *proof load testing*.

5.2.1 DESTRUCTIVE TESTING

Performing a destructive test, e.g. loading a wire until failure, yields high-quality information in the sense that we can deduct the failure load of the wire with relatively small uncertainties.

While destructive testing is the only way to determine the exact failure load of an element, it has its drawbacks. The element is destroyed in the process and cannot be reused. Performing several tests requires several elements, which in many cases is considered a rather expensive way of gathering information. Some practical issues also arise; Destructive testing is impractical for estimating the capacity for existing cables, because only a few wires can be removed from the cable before the load bearing capacity is reduced. This greatly limits the use of destructive testing on existing cables. The obtained parameters of the result (i.e. failure capacities of all the wires) can be estimated with sample mean and standard deviation found using equation (5.1) and (5.2).

$$m_X = \frac{1}{n}\sum_{i=1}^{n} X_i \quad (5.1) \qquad s_X = \sqrt{\frac{1}{n-1}\cdot\sum_{i=1}^{n}(X_i - m_x)^2} \quad (5.2)$$

5.2.2 NONDESTRUCTIVE TESTING

With nondestructive testing, the elements are tested for a specific load level. If the wire survives, the strength is equal or greater than the specified load. This kind of testing yield inequality type of information [23]. It should be pointed out that the proof load level has to be set reasonably high. If not, the test will not yield much information regarding the strength of the wire [18]. Prior data from failure test or material data should therefore be considered beforehand.

In this paper, two methods for updating the capacity due to proof loading are considered. A series of proof loading tests or a single proof load test on an existing cable. The updated PDF and CDF with a single proof load test for an existing component are given in equation (5.3) and equation (5.4) [24].

$$f_R^{"}(\mathrm{x}) = \frac{f_R(x)}{1 - F_R(q^*)} \quad (5.3)$$

$$F_R^{"}(x) = P_r\left(R \le x \,|\, R \ge q^*\right) = \frac{\Pr(\mathrm{R} \le \mathrm{x} \cap \mathrm{R} > q^*)}{\Pr(\mathrm{R} > q^*)} = \frac{F_R(\mathrm{x}) - \mathrm{F}_R(q^*)}{1 - F_R(q^*)} \quad (5.4)$$

Where q^* is the proof load, f_R the PDF and F_R the CDF of the material resistance.

Equation (5.3) and (5.4) is only valid for $x > q^*$, or else the value will be zero. The effect of this is called a truncated model, and is seen in Figure 15 and Figure 16.

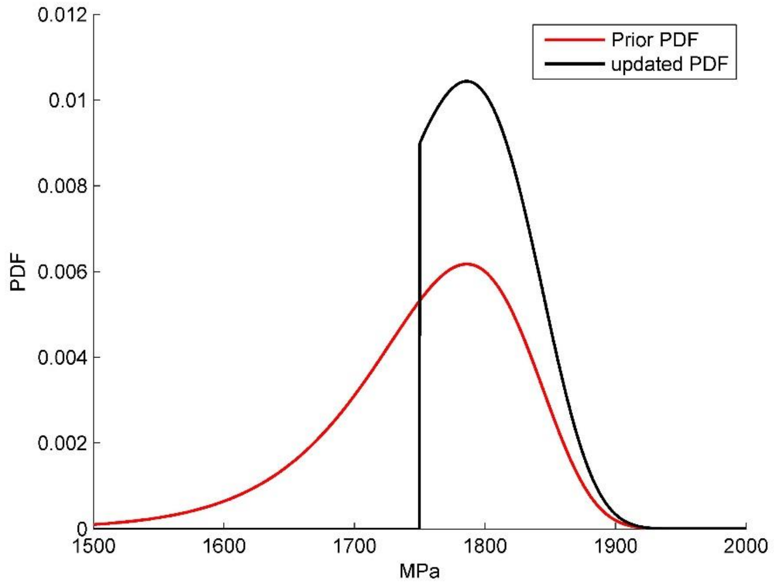


Figure 15 Updated PDF due to proof load

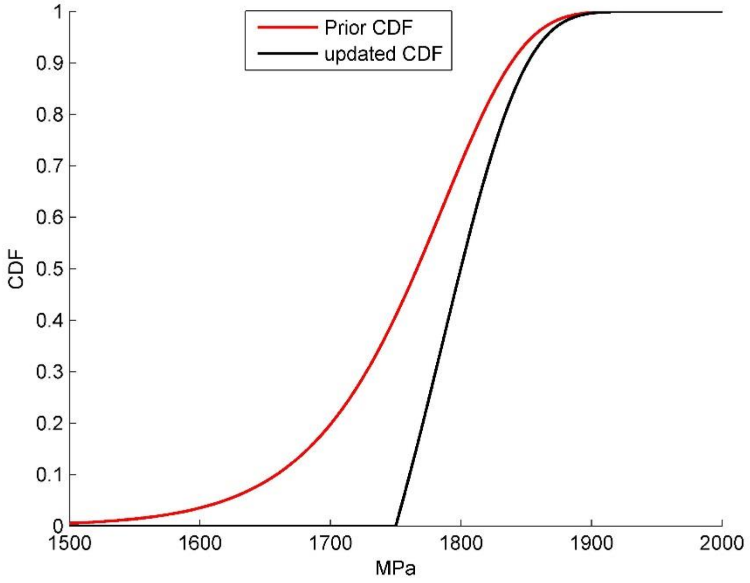


Figure 16 Updated CDF due to proof load

If the proof load testing of several elements are to have a meaning, the correlation between the tested elements must be reasonably high. If the correlations are low, the uncertainties dominates the results, gaining little information. In the paper *Probabilistic models for proof load testing* [25] it is shown that the required number of proof tests are low, because the shape and values of the functions remain similar. This is because the information that is gained from proof load testing has a limited content [25].

5.3 UPDATING WITH ANALYTICAL SOLUTION

Analytical solutions are based on the conjugate prior function [26], which gives a closed form solution of the distribution. Although the method is simpler, the results are not as accurate as in with the numerical solution. If the exact solution is needed, then a numerical solution must be used. An analytical solution lets the user see the effect between prior information and new information much clearer than in the numerical solution. The updating process is described in detail below.

The analytical solution for a normal distribution is presented below. When the shape and scale parameters of the Weibull distribution is unknown, the prior natural conjugate does not exist [22]. An approximate solution of the analytical solution for a Weibull can be found in *"A compendium of conjugate priors"* [27], but a solution for the analytical Weibull distribution will not be presented here. Two analytical normal solutions presented. The two cases are:

1. unknown mean and known standard deviation
2. unknown mean and standard deviation

The calculations of case 1 and 2 are presented in Appendix E – MATLAB script for analytical updating.

5.3.1 *UNKNOWN MEAN AND KNOWN STANDARD DEVIATION*

The normal distributed variable $X \sim N\,(M_x, \sigma_x)$ is considered where the standard deviation σ_x has a known value and the mean is considered as a random variable described as a normal distribution. $M_x \sim N\,(\mu', \sigma')$. The normal conjugate prior is used to find the analytical solution.

Equation (5.5) describe the posterior distribution of M_x, with updated parameters μ'' and σ'' in equation (5.6).

$$f''(\mu_x) = \frac{1}{\sqrt{2\pi}\sigma''}\exp\left(-0.5\left(\frac{\mu_x - \mu''}{\sigma''}\right)^2\right) \qquad (5.5)`$$

$$\mu'' = \frac{\frac{\mu'}{n} + \frac{\bar{x}}{n'}}{\frac{1}{n} + \frac{1}{n'}} \qquad (5.6) \qquad\qquad \sigma'' = \sqrt{\frac{\frac{\sigma_X \sigma'}{n'n}}{\frac{\sigma_x^2}{n} + \frac{\sigma_x^2}{n'}}} \qquad (5.7)$$

$$n' = \frac{\sigma_X^2}{\sigma'^2} \qquad (5.8) \qquad\qquad \sigma' = \sqrt{\frac{\sigma_x^2}{n'}} \qquad (5.9)$$

Where

- *σx, μ', σ'* and *n'* are respectively the known standard deviation, the prior mean, prior standard deviation of *Mx* and number of tests.
- $\bar{x}$ and *n* are respectively the mean value of the additional tests and number of additional tests.

The predicted analytical solution found in equation (5.10) then gives the updated strength distribution, due to the load level *x*. Equation (5.11) is the predictive standard deviation.

$$f(x\,|\,\hat{x}) = \frac{1}{\sqrt{2\pi}\sigma'''}\exp\left(-0.5\left(\frac{x-\mu''}{\sigma'''}\right)^2\right) \qquad (5.10)$$

$$\sigma''' = \sqrt{\sigma''^2 + \sigma_X^2} \qquad (5.11)$$

In contrast to the numerical solution, it is easier to weigh the prior information with the additional information from new tests in the analytical solution. This weighting factor is shown in equation (5.9), where the uncertainty of the prior mean is calculated based on the known standard deviation of the prior distribution and number of prior tests. The predictive posterior distribution with varying number of tests are given in Figure 17 [28].

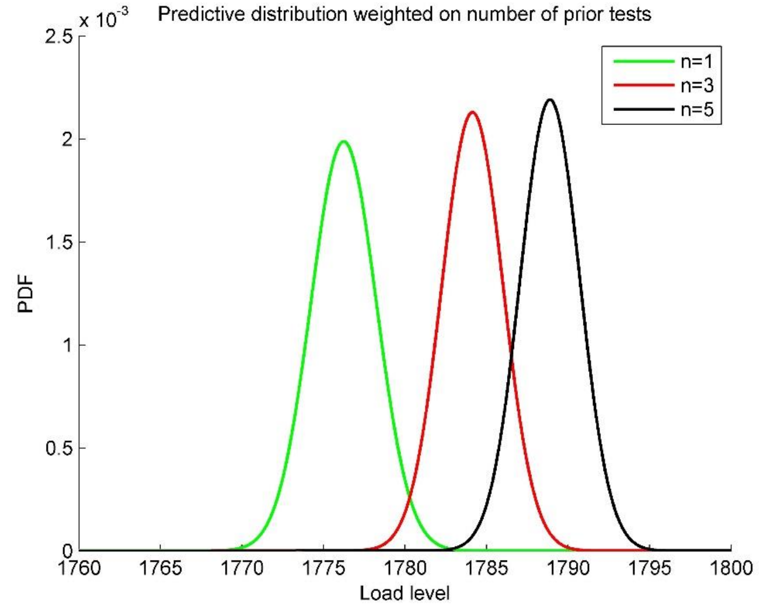


Figure 17 Weighted predictive distribution due to number of prior tests

5.3.2 UNKNOWN MEAN AND STANDARD DEVIATION

A normal distribution of the variable *X* has the parameters *(M_X, $\sum_X$)*, where both the mean and the standard variation is considered as random variables. The analytical solution is found by using the natural conjugate prior for the distribution of the parameters. In this case the conjugate prior is the Normal-Inverse-Gamma-2 distribution, see Raiffa and Schlaifer [29] and Rackwitz [30].

$$f_{M,\Sigma}(\mu,h\,|\,m',s',n',v')=\frac{\sqrt{hn'}}{\sqrt{2\pi}}\exp\left(-0.5\left(\frac{\mu-m'}{\frac{1}{\sqrt{hn'}}}\right)^{2}\right)\frac{1}{2\Gamma\left(\frac{v'}{2}\right)}\left(0.5v's'^{2}h\right)^{0.5v'-1}\exp(-0.5v's'^{2}h)v's'^{2}$$

(5.12)

$$m=\frac{1}{n}\sum_{i=1}^{n}\hat{x}_i \quad (5.13) \qquad s^2=\frac{1}{n-1}\sum_{i=1}^{n}(m-\hat{x}_i) \quad (5.14) \qquad v=n-1 \quad (5.15)$$

Where *m*, s^2, *n* and *v* is based on additional tests of $\mathbf{x_i}$.

$$m''=\frac{n'm'+nm}{n'+n} \qquad (5.16)$$

$$n'' = n + n' \quad (5.17)$$

$$s''^2 = \frac{(v's'^2 + n'm'^2) + (vs^2 + nm^2) - n''m''^2}{(v' + \delta(n')) + (v + \delta(n)) - \delta(n'')} \quad (5.18)$$

$$v'' = (v' + \delta(n')) + (v + \delta(n)) - \delta(n'') \quad (5.19)$$

$$\delta(x) \equiv \begin{cases} 0 & for \;\; x \le 0 \\ 1 & for \;\; x > 0 \end{cases} \quad (5.20)$$

By changing out *m', s', n'* and *v'* in equation (5.12) with *m'', s'', n''* and *v''*, it gives the predictive posterior distribution. The influence of different numbers of prior tests on the predictive posterior distribution are shown in Figure 18.

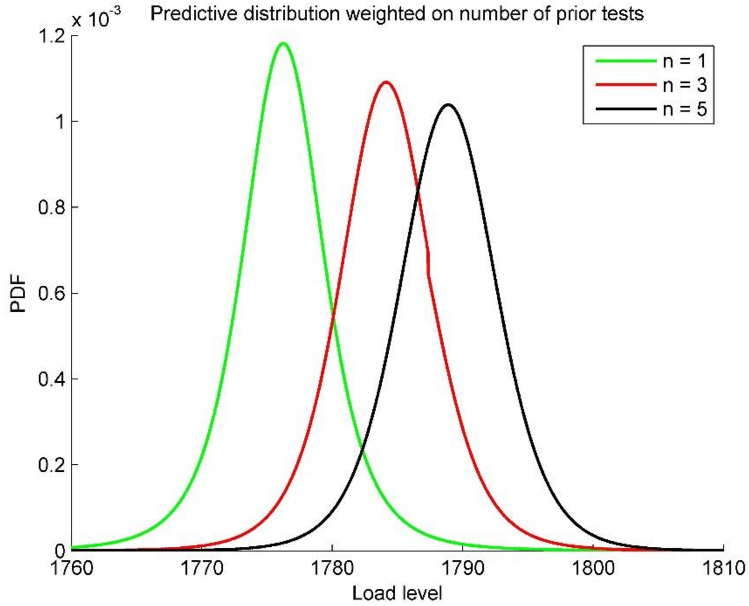


Figure 18 Weighted predictive distribution based on number of prior tests

5.4 UPDATING WITH NUMERICAL SOLUTION

The numerical solution is based on finding a numerical answer, for the problem at hand. By finding the solution of the integration, the exact answer is found. Different integration methods for solving the integrals for the numerical solution will later be presented in chapter 5.5.

To find an answer numerically is both time consuming and often difficult, so an approximate solution is found by using the analytical solution presented in chapter 5.3. The steps for a numerical updating scheme is presented below. The theory below is implemented in a MATLAB script shown in Appendix D –MATLAB script.

5.4.1 BAYE'S RULE, BAYE'S THEOREM AND THE LAW OF TOTAL PROBABILITY

Baye's rule is valid in cases where current or future probabilities relates to the probabilities found in earlier observations. When considering the behavior of a structure this will often be a valid point. The densities and strengths found in previous or similar cases will be correlated in one way or another. The mathematical formula describing "Baye`s rule" finds the probability of event A occurring, given that event B has already happened or is true.

$$P(A|B) = \frac{P(\mathrm{B}|\mathrm{A}) \cdot P(A)}{P(B)} \quad (5.21)$$

$$P(\mathrm{A}) = \int_{\psi} P(\mathrm{A}|\mathrm{B}) P(\mathrm{B}) \quad (5.22)$$

Where

- *P(A)* is the probability of event *A* occurring
- *P(B)* is the probability of event *B* occurring
- *P(B|A)* is the probability of event *B* occurring, when event *A* has occurred
- *P(A|B)* is the probability of event *A* occurring, when event *B* has occurred

Equation (5.21) is Baye's rule or Baye's theorem [5].

In its original form, the theorem states that *P(A)* and *P(B)* are independent of each other, i.e. uncorrelated. Naturally, this will not hold true for many practical cases. For instance: the failure probability of a structure depends on many variables, including fatigue, corrosion, deterioration etc. These variables will usually correlate in some way; we therefore need to introduce a way to express this correlation when Baye's theorem is valid [24]. The law of total probability is described in equation (5.22), finds the unconditional probability estimate.

5.4.2 LIKELIHOOD FUNCTION

The likelihood function and the log likelihood function for destructive testing are given below [7].

$$L(\theta \mid \mathrm{x}_i) = \prod_{i=1}^{n} f(x_i \mid \theta) \qquad (5.23)$$

$$l(\theta \mid x) = \sum_{i=1}^{n} \log(f(x_i \mid \theta)) \qquad (5.24)$$

Equation (5.23) and (5.24) does not consider the proof load, and requires modification. This is done by adding a term that describe probability of failure for the elements that fail the proof loading test. This term is then raised to the power of the number of tests that pass the proof load test. The modified likelihood- and log likelihood function for proof load testing are given in equation (5.25) and (5.26) [31].

$$L(\theta \mid \mathrm{x}_i) = \left[\prod_{i=1}^{k} f(x_i \mid \theta)\right]\left[(1 - F(\mathrm{q}^* \mid \theta))^{n-k}\right] \qquad (5.25)$$

$$l(\theta \mid x) = \left(\sum_{i=1}^{n} \log f(x_i \mid \theta)\right) + (\mathrm{n} - \mathrm{k})\log(1 - \mathrm{F}(\mathrm{q}^* \mid \theta)) \qquad (5.26)$$

5.4.3 BIVARIATE NORMAL DISTRIBUTION

The covariance matrix include the parameters uncertainties and correlation. As seen in equation (5.27) the elements in the matrix are expressed by the variance on the diagonal, and the standard deviation multiplied with the correlation factor ρ [32].

By including the correlation between the parameters in the ***C*** matrix and the mean values of the parameters in the vector ***μ*,** it can be shown that the general PDF for a multivariate normal distribution is given by equation (5.28).

$$\mathbf{C} = \begin{pmatrix} \sigma_{\theta_1}^2 & \rho\sigma_{\theta_1}\sigma_{\theta_2} \\ \rho\sigma_{\theta_2}\sigma_{\theta_1} & \sigma_{\theta_2}^2 \end{pmatrix} \qquad (5.27)$$

$$f(\theta) = \frac{1}{\sqrt{(2\pi)^n \cdot \det(\mathbf{C})}} \exp\left(-\frac{(\boldsymbol{\theta} - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu})}{2} \right) \qquad (5.28)$$

Where n is the rank of the covariance matrix, for a 2x2 matrix the rank n = two [32]. For the bivariate 2-dimentional case, the multivariate jointly normal distribution can be rewritten as:

$$f'(\theta_1, \theta_2) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)} \left[\frac{(\theta_1 - \mu_1)^2}{\sigma_1^2} + \frac{(\theta_2 - \mu_2)^2}{\sigma_2^2} - \frac{2\rho(\theta_1 - \mu_1)(\theta_2 - \mu_2)}{\sigma_1\sigma_2} \right] \right)$$

(5.29)

The multivariate jointly normal distribution will now take the uncertainties of the estimated parameters into account. A 3D plot of the bivariate jointly normal distribution can be seen in Figure 19.

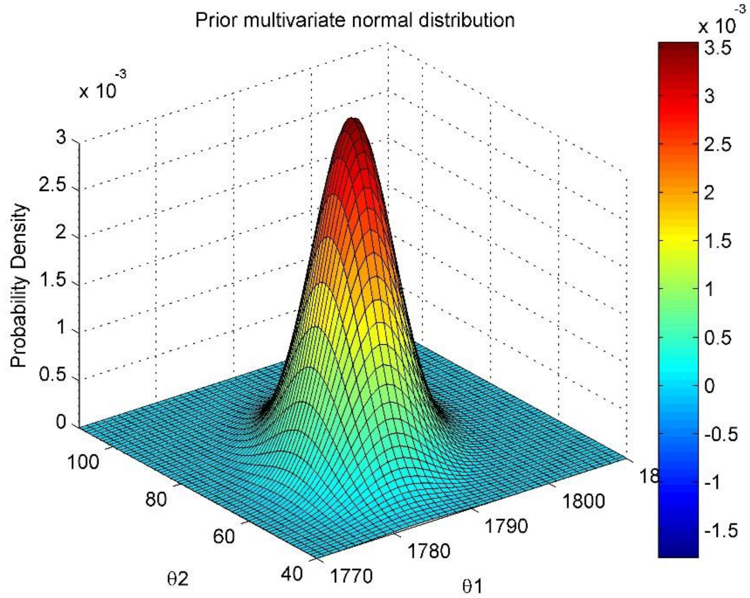


Figure 19 Prior multivariate normal distribution of θ_1 and θ_2

5.4.4 *POSTERIOR DISTRIBUTION*

The posterior distribution of the parameters is found by combining the equation for Baye's rule (5.21), the multivariate jointly normal distribution (5.29) and the likelihood function (5.25). The posterior distribution is written as [18].

$$f^{''}(\theta \,|\, \hat{x}) = \frac{f^{'}(\theta) \cdot L(\theta \,|\, \hat{x})}{C} \qquad (5.30)$$

$$C = \int_{\theta} f^{'}(\theta) \cdot L(\theta \,|\, \hat{x}) d\theta \qquad (5.31)$$

The updated information regarding the values of the parameter(s) θ, is now included in the posterior distribution (5.30). To make sure that the area under the graph is one, a normalizing constant C is introduced in equation (5.30). The normalizing constant (equation (5.31)), is in general found by integrating between the limits ±∞. An approximation of the integration limits is often used to save computing time.

A general procedure for finding the integration limits can be found by plotting the density function of the normalizing factor dependent on the parameters θ_1 and θ_2 (see Figure 20). The integration limits is deducted by implementing a stopping criterion. This stopping criterion is given by the following equation:

$$\begin{aligned} & f(\theta_{1M}, \theta_{2M}) - f(\theta_{1i}, \theta_{2i}) \approx f(\theta_{1M}, \theta_{2M}) \\ & f(\theta_{1M}, \theta_{2M}) = \text{ Maxima} \\ & f(\theta_{1i}, \theta_{2i}) = \text{ Arbitrary point} \end{aligned} \qquad (5.32)$$

I.e. when the difference between the maxima (red area) and an arbitrary point is equal to the maxima. Outside the black square showed in Figure 20, the values of θ_1 and θ_2 are below the bounds of interest, meaning values of order 10^{-22}. Simplified, the integration limits are found by invoking 10^{-4}-$10^{-22} \approx 10^{-4}$. By implementing this, the integration limits are shown as the black square in Figure 20.

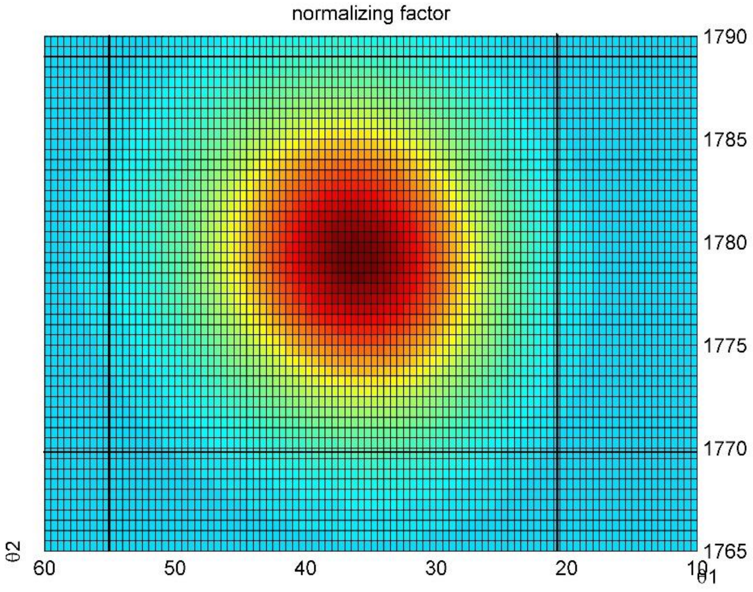


Figure 20 Normalizing factor dependent on θ₁ and θ₂

5.4.5 PREDICTIVE DISTRIBUTION

The dependent predictive distribution is now defined as the product between a given PDF for a certain distribution and the posterior distribution found in equation (5.30).

$$f'''(\mathrm{X}\,|\,\theta) = f(X\,|\,\theta)\,f''(\theta) \qquad (5.33)$$

A predictive distribution that is dependent on the parameter θ is not interesting from an engineer's point of view. The unconditional predictive distribution is found by utilizing the law of total probability given in equation (5.22).

$$f_X(X) = \int_{-\infty}^{\infty} f(X\,|\,\theta)\,f''(\theta)\,d\theta \qquad (5.34)$$

The predictive probability density function will now be unconditional with respect to the parameter θ, and is now a function with respect to the variable *x*. From an engineer's point of view, this part is interesting due to that is gives the updated distribution as a function of the load [18].

5.5 INTEGRATION METHOD

In cases where the exact solution to an integral is unknown, or uncertain, adaptive methods are used. Computer programs utilize numerical integration in order to estimate the integrals. Although modern computer programs yields a high degree of accuracy, some error will always occur. The origin of this error lies in the way the programs evaluates the integral by invoking different quadrature rules. Simply explained, these quadrature rules divide the domain into different parts and approximates the area in each part. Two different methods of numerical integration are used in the paper, where both are based on the Newton-Cotes method.

Newton-Cotes method

Newton-Cotes method evaluates an integral by assuming that the values of the function is known at equally spaced points [30]. As seen in Figure 21, if we increase the number of integration points between the limits, the numerical solution converge toward the exact solution.

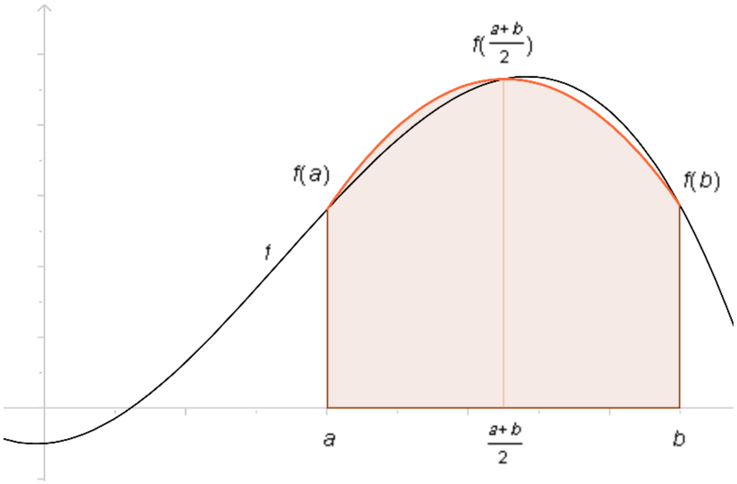


Figure 21 Newton-Cotes formula for n = 2 [30]

There are two ways to express the Newton-Cotes formula, the closed and open type. While the closed type uses *all* points in the function, the open type does not use the values at the endpoints of the function. Equation (5.35) denotes the open form of the integral, where w_i is a weighting factor.

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n-1} w_i f(x_i) \quad (5.35)$$

5.5.1 RECTANGULAR INTEGRATION

One of the basic open Newton-Cotes formula is the so-called rectangle method, or midpoint rule. As the name might suggest the integration is simplified by dividing the function into different rectangular sections, as seen in Figure 22.

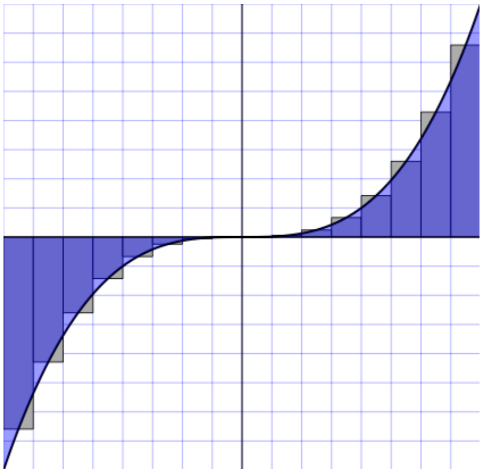


Figure 22 Midpoint approximation [33]

Using simple logical reasoning, the numerical integration can now be simplified as the sum given in equation (5.36). Where *h* denotes the width of the rectangles, and *n* the number of rectangles.

$$\int_a^b f(\mathrm{x})\,\mathrm{dx} \approx h\sum_{n=0}^{n-1} f(x_n) \qquad h = \frac{b-a}{n} \tag{5.36}$$

One of the main advantages using the rectangular method is that it is relatively easy to calculate the area of the rectangular sections, making the method an efficient one. As the number of rectangles increases, the numerical estimation goes towards the exact solution (n→∞).

5.5.2 TRAPEZOIDAL INTEGRATION

As seen in Figure 23, trapezoidal integration estimates the integral by approximating the function as a linearization between the endpoints *a* and *b*. The source of error is greatly reduced, since the linearization is a better fit than the rectangular method.

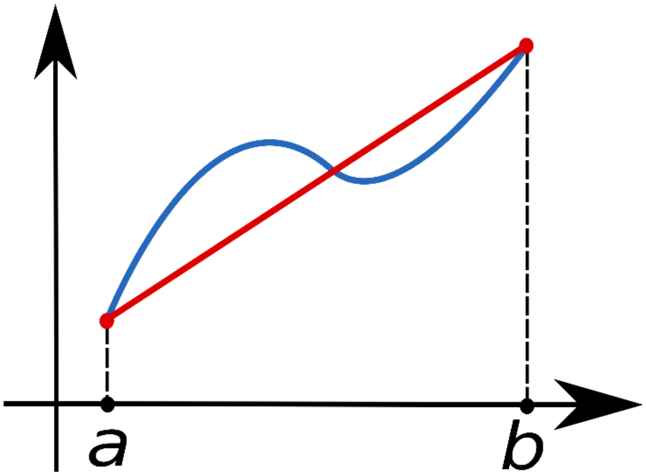


Figure 23 Linear approximation [34]

The definite integral is now approximated using equation (5.37).

$$\int_a^b f(x)dx \approx (\mathrm{b}-a)\left[\frac{f(\mathrm{a})+f(\mathrm{b})}{2}\right] \qquad (5.37)$$

Compared to the rectangular method, a trapezoidal estimate both converges faster and reduces the source of error greatly. Figure 24 compares the estimated linearization (red line) to that of the function (blue line). The trapezoidal rule is exact for all functions with polynomial degree of one or less [34].

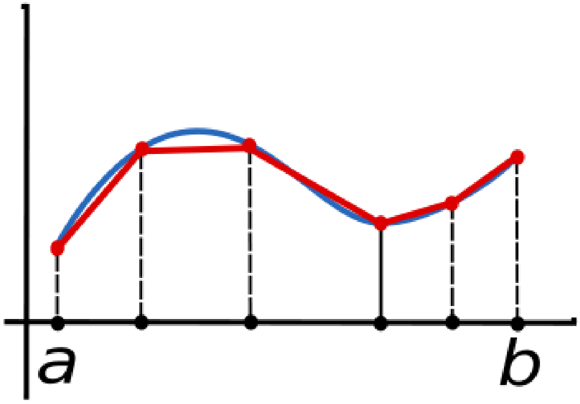


Figure 24 Trapezoidal integration [34]

The source of error in the numerical integration is given as the difference between the exact solution and the numerical integration.

5.6 ESTIMATED PARAMETERS OF THE PREDICTIVE DISTRIBUTION

The predictive graph is of no use as it is now, because the parameters of the distribution are unknown. To make it possible to compare the updated posterior distribution with the prior distribution, a distribution must be fitted to the predictive posterior distribution. Method of least square is one way of doing this.

5.6.1 METHOD OF LEAST SQUARE

The method of least square is used to find the value of parameters that minimize the sum of a series of differences are shown in equation (5.38) [35].

$$\min \sum_{i=1}^{n} (t_i - a_i(\mathrm{p}_j))^2 \quad (5.38)$$

Where

- t_i are known values
- $a_i(\mathrm{p}_j)$ is a function with values for parameters p_j

Least square method is used in both linear and non-linear equations. When the equation is non-linear, Gauss-Newton method can be employed. This method is based on the same method as the maximum likelihood. By finding the Hessian matrix, Fisher matrix or the gradients of the sum, an estimation of the parameters can be deduced with an optimization process [36], [35]. The method of least squares is often used to find an approximated distribution for a given data set.

5.6.2 LOWER TAIL LIMIT

An approximated distribution for the entire predictive graph is not of interest, because the characteristic resistance of a material is described by the 5% fractile [37]. Therefore, only a certain range of the predictive distribution should have a close as possible fit. This range can be set at a certain level, either by calculating, or by simply setting the value as a constant. An approximated fit of the lower tail up to a chosen value of the predictive distribution. Here the value is set to 1550 MPa, and shown in Figure 25.

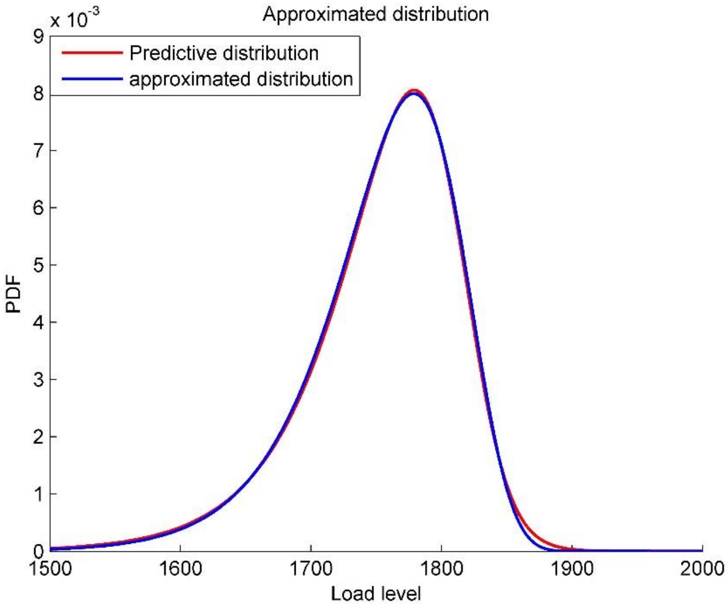


Figure 25 Approximated Weibull distribution fitted for lower tail

Some considerations should be taken when performing approximations using least square method. The function will only find the first minima, so if the function contains more than one minimum point, it can lead to errors. Some caution is therefore recommended.

6 RELIABILITY AND PROBABILITY OF FAILURE

The methods in which the probability of failure and the reliability of the system are updated are given in this chapter. By utilizing the updated strength distribution, the reliability of the system is updated accordingly. Some of the main concepts regarding target reliability is given in chapter 6.1, where the first part also explains how it is possible to estimate the reliability assessment over time. How the probability of failure of the wires are calculated, is explained in chapter 6.2.

6.1 RELIABILITY

The inhabitant or users of a bridge are always concerned about the safety of the bridge. If a bridge feels unsafe, no one will use it. In engineering terms, the word “Safety” is a rather inadequate term, since it is a highly subjective word, where the exact meaning differs from one person to another [4]. Instead, the concept of reliability is introduced; this defines the probability of failure for a component and yields a quantifiable answer of how safe the component is [4]. This must not be confused with objectivity. Probability of failure can rather be seen as a result where we assume a certain *degree of confidence* based upon the amount of information available. A high amount of information corresponds with a high degree of confidence [4]. Reliability is evaluated based on three criterion:

- Cost effectiveness of the component
- Danger to environment
- Safety of people

For structures the reliability is determined by the *design equations* which is governed by the relationship between the resistance (load capacity) and the load, as given in (6.1) [4].

$$\begin{gathered}\{\text{Resistance}\} > \{Load\} \\ R > S\end{gathered} \qquad (6.1)$$

The probability of failure is given as the probability that the total load exceeds the resistance, the reliability, P_S can then be expressed as [4]:

$$P_f = \Pr(R \le S); \mathrm{P}_f \in [0,1] \qquad (6.2)$$

$$P_s = 1 - P_f \qquad (6.3)$$

It is important to note that the following yields the probability that the object will serve its intended purpose for a specific period only when the *defined* conditions are in play. Understanding limitations of these conditions is vital when interpreting the results of equation (6.3) [4]. This is perhaps more simply explained if a cable with a designated lifetime, e.g. 100 years is considered. If the effects of corrosion is omit, or neglected due to new production methods, construction methods etc. the estimated probability of failure is only valid if there is no corrosion.

The *target reliability index* can be seen as a 'safety margin' where we introduce a desired safety index (β_t), usually ranging from 3 – 5 [37]. To calculate the reliability of a structure the basic design equation is often used.

$$M = R - S \qquad (6.4)$$

Equation (6.4) yields the following mean and variance [4]:

$$E\left[M\right] = \mu_M = E\left[R\right] - E\left[S\right] \quad (6.5)$$

$$Var\left[M\right] = \sigma_M^2 = Var\left[R\right] + Var\left[S\right] \qquad (6.6)$$

If R and S are normal distributed, and using only the mean and standard deviation of the variables β is given as [4]:

$$\beta = \frac{\mu_M}{\sigma_M} \qquad (6.7)$$

Failure occurs when $M \leq 0$, i.e. when $R \leq S$. For normal distributed variables, the failure probability is then given as [4]:

$$\Pr(F) = F_M(0) = \Phi\left(\frac{0 - \mu_M}{\sigma_M}\right) = \Phi\left(-\beta\right) \qquad (6.8)$$

By rewriting equation (6.8), the reliability can be described by the probability of failure.

$$\beta = -\Phi^{-1}(P_f) \qquad (6.9)$$

Values for the target reliability index are given in codes e.g. Eurocode [37] or JCSS [38], for different levels of consequence of failure and cost of reducing the probability of failure. The index consider the effect of failure and the cost of reducing the chance of such a failure. 'Effect of failure' include not only economic loss, but also social, loss of human life and environmental consequences.

Time dependent reliability

The reliability of the cable will change its lifetime. For predicting the reliability over time, two different ways are considered.

The reliability can be considered as time invariant. Simply put; the probability of failure over a given period is independent from the probability of failure of the previous period. For elements that is not subjected to wear or degradation, this method is used. The time invariant reliability distribution is expressed in equation (6.10) [18].

$$\text{Time dependent reliability} = -\text{norminv}\left(1-\left(1-P_f\right)^n\right) \qquad (6.10)$$

Where the probability of failure is constant and *n* is the number of years. For a series system, considering the reliability as time invariant will be a conservative assumption [18]. The distribution of time dependent reliability over a hundred years, when considering a yearly probability of failure equal to $1{\cdot}10^{-7}$ is seen in Figure 26.

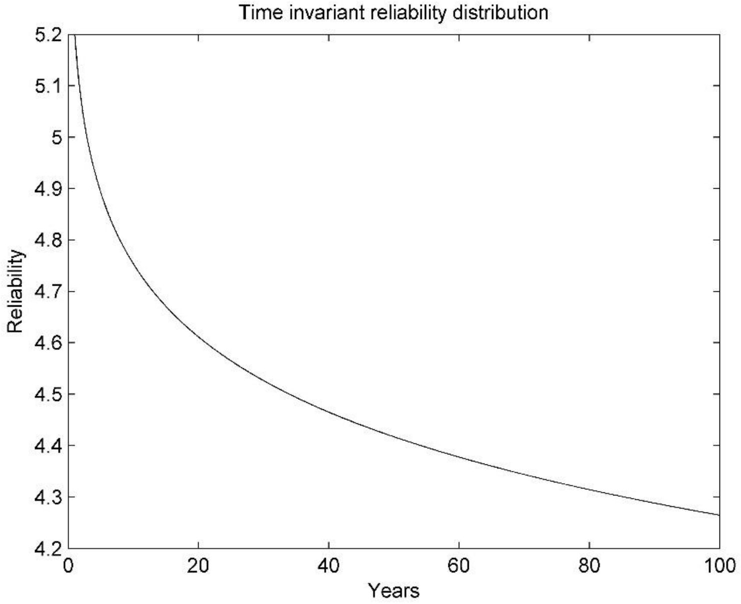


Figure 26 Time invariant reliability distribution

The reliability can also be considered as time variant. This means that the probability of failure for a given period is dependent on the probability of failure of the previous periods. The length of these periods can be, e.g. days, weeks, months, years etc. A year is often used for this period [18]. An example for time variant method is how the probability of failure for a cable increases over time, due to corrosion and wear. The expression for the time variant reliability is given in equation (6.11) [18].

$$\text{Time variant reliability} = -\text{norminv}\left(1-\prod_{i=1}^{n} 1 - P_{f\,n}\right) \qquad (6.11)$$

6.2 PROBABILITY OF FAILURE

The probability of failure given in equation (6.8) is only valid when both the load and the resistance are described by a normal distribution. This holds true for some problems, but not generally. Instead the probability of failure can be found more generally by using e.g. First Order Reliability Method, Monte Carlo simulations, integral etc. How the probability of failure is derived using Monte Carlo and integration is presented next. The presented methods are based on that the parameters of the load distribution are unknown, but with an assumed value for the coefficient of variation. The parameters of the resistance are presumed known.

6.2.1 *PROBABILITY OF FAILURE USING MONTE CARLO SIMULATIONS*

Monte Carlo can calculate the probability of failure by running simulations of the basic design equation (6.4), and counting how many of the simulations that get a negative value. The probability of failure for a Monte Carlo simulation is defined in equation (4.40). How a Monte Carlo simulation works was described in detail in chapter 4.3.4, and will therefore not be presented here. Monte Carlo simulations will only give the approximated answer, but provided that enough simulations are performed it will converge towards the exact answer. The exact answer is found using the integral presented below.

6.2.2 PROBABILITY OF FAILURE USING INTEGRAL

The probability of failure can also be calculated directly using equation (6.12).

$$P_f = \iint_{R\leq S} f_r(r)\, f_s(s)\, ds dr = \int_0^{\infty}\left[\int_0^{s} f_r(\mathrm{r}) dr\right] f_s(s) ds = \int_0^{\infty}\left(1-F_s(r)\right) f_r(r) dr \qquad (6.12)$$

Both Monte Carlo and the integral require a load distribution. The parameters of the load distribution are found by combining two equations, where the first equation is found by rewriting equation (6.4) so that the resistance is equals the load. This maximizes the probability of failure and the rewritten design equation is shown in equation (6.13).

$$\frac{r_k}{\gamma_m} = s_k \gamma_s \qquad (6.13)$$

Where r_k and s_k are described as the inverse cumulative distribution for the resistance and the load respectively. γ_m is the partial safety factor for the resistance and γ_s is the partial safety factor for the load. The second equation is found by combining the equation for mean and standard deviation of the load distribution, with the given coefficient of variation. An often-used distribution for load is the Gumbel distribution.

6.2.3 GUMBEL DISTRIBUTION

The Gumbel distribution is used to model e.g. the maximum load on a wire. It is also applied in modeling loads on structures. The PDF and CDF are given in equation (6.14) and (6.15) [39].

$$f(x) = \frac{1}{\beta}\exp\left(-\left(\frac{x-\mu}{\beta}\right) + \exp\left(-\left(\frac{x-\mu}{\beta}\right)\right)\right) \qquad (6.14)$$

$$\mathrm{F(x)} = \exp\left(-\exp\left(-\left(\frac{x-\mu}{\beta}\right)\right)\right) \qquad (6.15)$$

Where μ is the location factor and β is the scale parameter. The mean and standard deviation of the Gumbel distribution are given as:

$$E\left[X\right] = \mu + \beta\gamma \qquad (6.16)$$

$$\text{Standard deviation} = \frac{\pi}{\sqrt{6}}\beta \qquad (6.17)$$

Where γ is the Euler's constant equal to 0.577215 [40].

6.2.4 PARTIAL SAFETY FACTOR

The partial safety factor for bridge cables depend on four main factors [1].

1. Target reliability
2. Damage allowance
3. Daniel's effect
4. Length effect

The partial safety factor calculated due to the four main factors above is presented in equation (6.18), which is adopted from Faber et al.[1].

$$\gamma_x = \frac{x_k}{\xi_L \xi_D \mu_x (1-\alpha_x \beta_t V_x)(1-\delta_{all})} \qquad (6.18)$$

Where:

- ξ_L take into account the reduction due to length effect (weakest link effect of a series system)
- ξ_D take into account the reduction due to Daniels effect
- δ_{all} take into account broken wires in a cable
- x_k is the characteristic value
- $\mu_X, \alpha_X, \beta_X, V_X$ is respectively the mean, the FORM sensitivity factor, target reliability and coefficient of variation.

The partial safety factor for a single wire is calculated by omitting the factors that consider the system effects. The reduction factors for Daniels effect and length effect are set to one, while the damage accumulation (number of broken wires) is set to zero. The partial safety factor for a single wire is found in equation (6.19)

$$\gamma_x = \frac{x_k}{\mu_x (1-\alpha_x \beta_t V_x)} \qquad (6.19)$$

6.2.5 LOAD PARAMETER

All the parameters that's needed for finding the load distribution are now known. The parameters of the Gumbel load distribution are found in equation (6.20) and (6.21).

$$\beta = \frac{F_R^{-1}(\mathrm{x}_{\text{resistance}})}{\gamma_m \gamma_s \left(\dfrac{\pi}{\mathrm{cov}_s \sqrt{6}} - \gamma - \ln(-\ln(\mathrm{x}_{\text{load}})) \right)} \qquad (6.20)$$

$$\mu = \frac{F_R^{-1}(\mathrm{x}_{\text{resistance}})}{\gamma_m \gamma_s} + \beta \ln\left(-\ln\left(x_{load}\right)\right) \quad (6.21)$$

Where

- $x_{\text{resistance}}$ and x_{load} are the lower and upper percent fractile. For resistance the 5% fractile is used, and the 95% - fractile is used for the load [37].

7 RESULTS

The results are mainly presented as graphs taken from MATLAB followed by a short summary explaining each graph or table. The results are presented in the same order as the theory chapters, meaning that Daniels effect are presented first, initial strain second etc. The meaning of these results and their validity is discussed in chapter 8.

7.1 DANIELS EFFECT

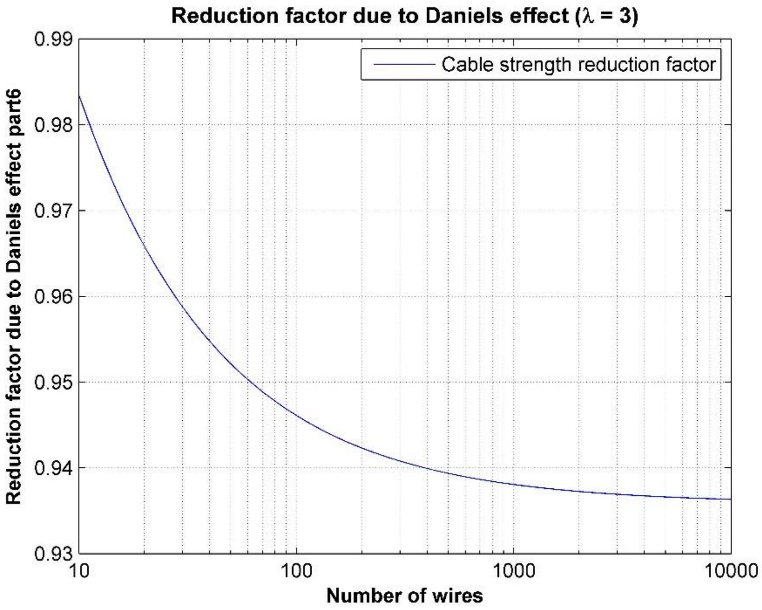


Figure 27 Reduction factor due to Daniels effect

As seen in Figure 27 above, the largest capacity reduction occurs when the number of wires range from 100 – 1000. If the number of wires are greater than 1000, the reduction factor is (more or less) constant.

7.2 INITIAL STRAIN IN CABLE

Two cases with different initial strain for brittle and ductile wire behavior are presented. Values for the mean and standard deviation for the yield- and plastic elongation zone are taken from Faber et al. [1]. Figure 28 shows the capacity without initial strain, while Figure 29 shows the capacity if the initial strain of the wires lies between 0.0 – 1.0%.

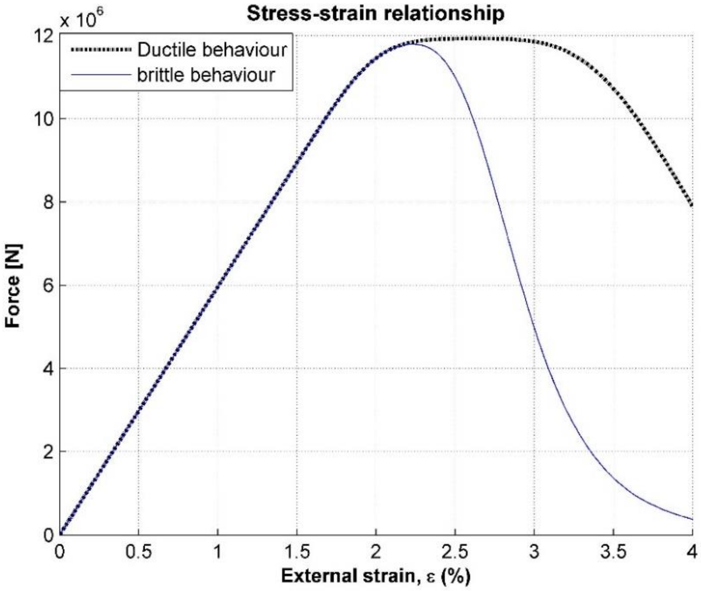


Figure 28 Average cable capacity, no initial strain

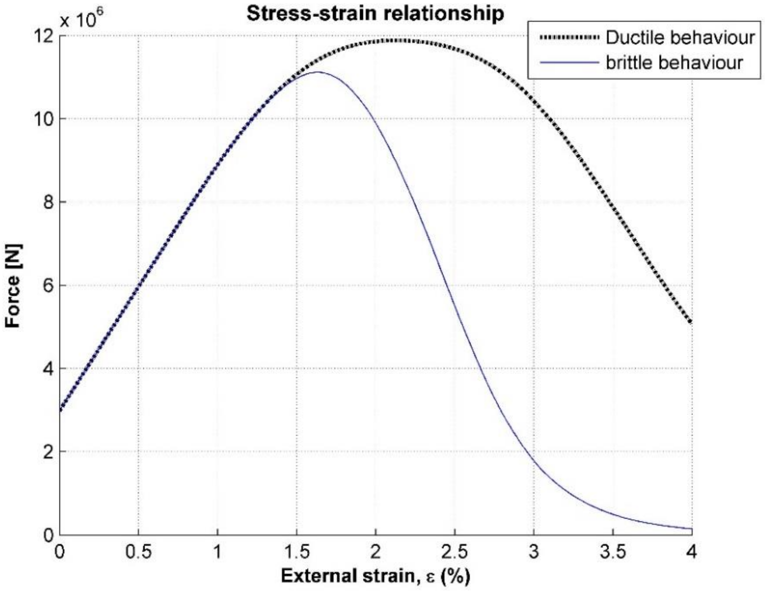


Figure 29 Average cable capacity, with initial wire strain between 0-1%

The capacities for a cable, with a given initial strain are shown in Table 4 below.

Initial strain [%]	**Max capacity [MN]**
0.0 – 0.0	11.8229
0.0 – 1.0	11.1887
1.0 – 2.0	11.1886

Table 4 Maximum cable capacity due to different initial strain

These results show a reduction around 0.6 MN, in both cases where the initial strains are relatively large 0-1% and 1-2%.

7.3 INTEGRATION

Three different integration limits for the normalizing factor in equation (5.31) are presented in Table 5 using the following integration limits: $\mu\pm4\sigma$, $\mu\pm8\sigma$ and the stopping criterion given in chapter 5.4.4.

Proof load (q*) and number of Failed/Passed tests		**Adaptive quadrature integration**			**Trapezoidal integration** **With calculation step 0.5**		
q*(MPa)	Failed/ Passed	Stopping criterion	$\mu\pm4\sigma$	$\mu\pm8\sigma$	Stopping criterion	$\mu\pm4\sigma$	$\mu\pm8\sigma$
1700	8/0	0.9982	1.0078	1.0015	1.0000	1.0000	1.0000
1680	7/1	1.0002	1.0068	1.0091	1.0000	0.9999	1.0000
1660	6/2	0.9982	1.0158	1.1753	1.0000	0.9995	0.9999
1640	2/6	0.9976	0.8659	0.5958	1.0000	0.9999	1.0000
1620	0/8	1.0000	NAN	NAN	1.0000	NAN	NAN

Table 5 Area under the normalized distribution

Table 5 above, shows the area of the normalized predictive distribution given by the methods explained in chapter 5.4.4 and the integration methods in chapter 0, shortly summarized, the area of the predictive function should be 1. As seen above, the adaptive quadrature integration will not yield a satisfactory accuracy. The trapezoidal integration on the other hand, offers a stable normalized area independent of the proof load and whether the wires pass or fail the proof load.

7.4 PREDICTIVE DISTRIBUTION

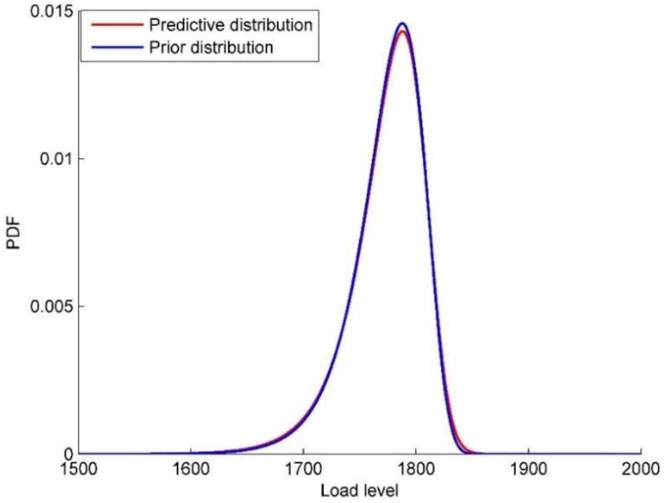


Figure 30 Updated strength of wire with proof level at 1620 MPa

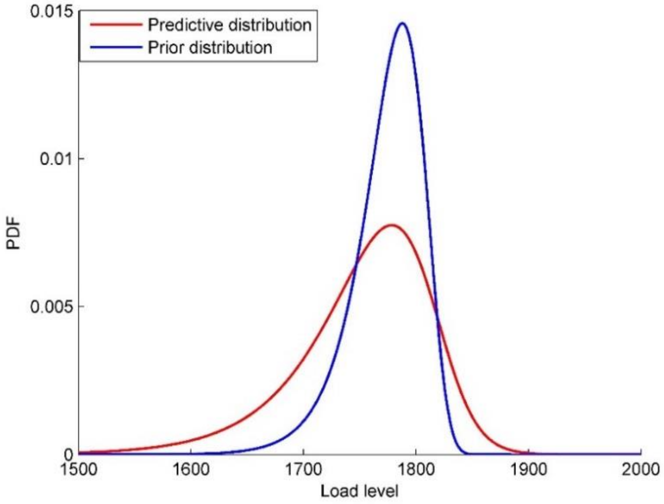


Figure 31 Updated strength of wire performing failure test

As seen in Figure 30, a proof load around 90% of the failure load yields little information, under the assumption that the wires are new. The effects on the predictive distribution if a wire fails is larger, as seen in Figure 31. As it can be seen from Figure 30 the updated probability now take into account the uncertainties of the parameters deduced from the maximum likelihood estimation of both the prior and the additional tests. The effect of this can be seen in that the predictive distribution curve is wider than the prior distribution. The reason is that the uncertainties of the parameters are now included in the distribution.

7.5 UPDATED CABLE CAPACITY

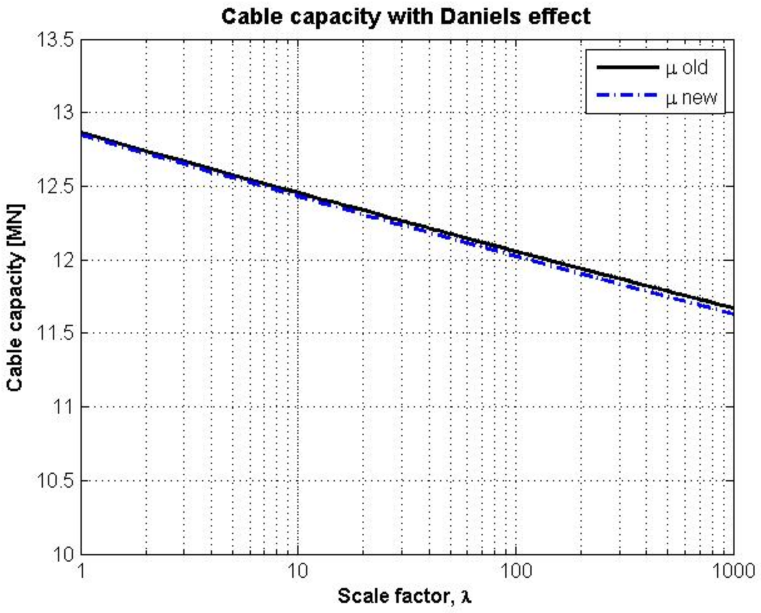


Figure 32 Cable capacity when all test pass the proof load test

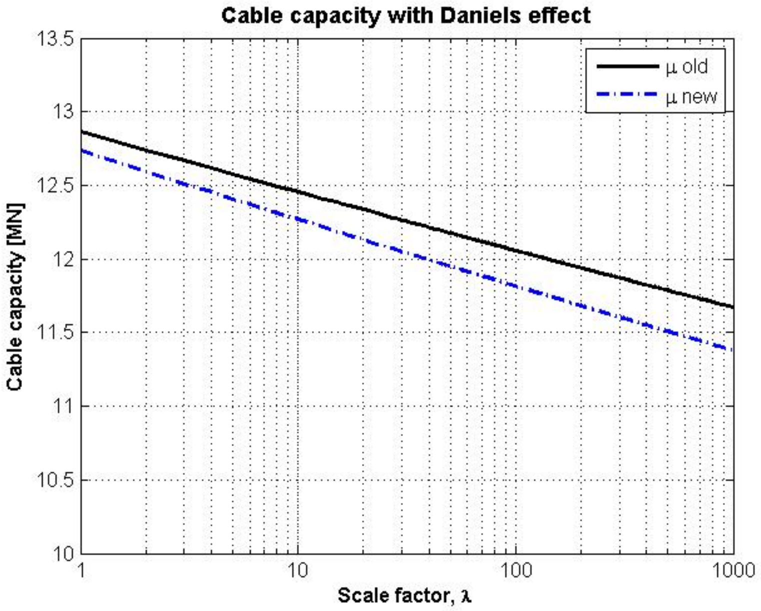


Figure 33 Cable capacity when one test fails the proof load test

The effects of the information gathered from proof load test and failure tests shown in chapter 5.2 becomes apparent in Figure 32 and Figure 33. Here the difference between the amounts of information gathered when none wire fails the proof load test, compared to when one test fail.

7.6 RELIABILITY

Target reliability	Prior safety factor	Posterior safety factor Proof load 1620 MPa	Posterior safety factor Proof load 1660 MPa
3	1,0212	1,0221	1,0412
4	1,0405	1,0423	1,0802
5	1,0606	1,0633	1,1223

Table 6 Partial safety factor for a single wire

As seen in Table 6 above, increasing the failure load from 1620 MPa to 1660 also increase the safety factor of the system. While a proof load of 1620 MPa have small impact on the safety factor, about 0.25% if β_t = 5, increasing the proof load to 1660 MPa result in an increase of 5.5% if β_t = 5.

			Proof load level 1620 MPa	
Target reliability	Prior probability of failure	Prior reliability	Posterior probability of failure	Posterior reliability
3	$1{,}0076 \cdot 10^{-16}$	8,2212	$4{,}0625 \cdot 10^{-16}$	8,0523
4	$8{,}4745 \cdot 10^{-17}$	8,2419	$3{,}3893 \cdot 10^{-16}$	8,0744
5	$7{,}0766 \cdot 10^{-17}$	8,2634	$2{,}8057 \cdot 10^{-16}$	8,0975

Table 7 Probability of failure and reliability of a wire, using proof load level, q* = 1620 MPa

The effects of updating the posterior distribution with a proof load q* is shown in Table 7. Given the wire samples shown in Appendix B - Partial safety factor, probability of failure and reliability , the effects of performing a proof test on the posterior changes the reliability of about 2% compared to the prior reliability. The validity of the presented values are discussed in chapter 8.5.

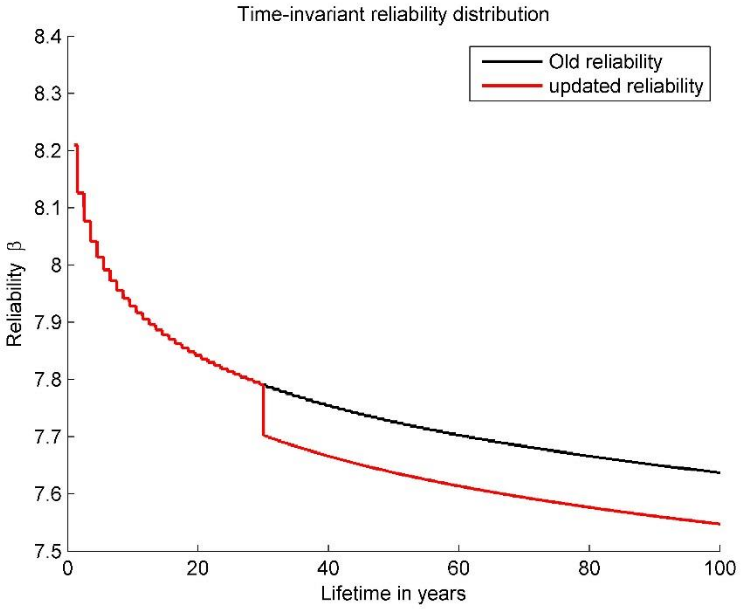


Figure 34 Time invariant reliability update, with values from row 3 in Table 6

Using the time invariant reliability updating method in equation (6.10) the time invariant reliability of a wire is shown in Figure 34. It is here seen that the updated reliability of the wire is smaller than the original reliability. The validity of this is discussed in chapter 8.5.

8 DISCUSSION

This chapter discusses the results and their validity. Issues that arose during the calculations or during the theoretical focus, is also mentioned. The chapter is built up like the rest of the thesis, where the first part focuses on the initial goals concerning Daniels effect and the effects of initial strain.

8.1 DANIELS EFFECT

One of the main issues during the investigation of Daniels effect was trying to understand the underlying construct of the theory. The theory, at least the one we were able to find, seemed to separate themselves into two groups. The first mainly consist of the paper Daniels wrote himself, which focuses on the mathematical aspect and not so much the practical. Although the basic concepts are clear, most papers and books simply state the formulas used in the calculation, with reference to the original paper. This a field the authors should have given more time and devotion. Perhaps some deeper knowledge of the construct and mechanics of the theory could have been gained by studying the original paper further. The main reason was, however, that there were not enough time. The results, at least as far as the authors can discern, seems to be correct. The graph resembles the one given in Faber et al. [1], the difference between the two are (most likely) due to differences in the software, different methods used in the calculations etc.

8.2 INITIAL STRAIN IN CABLE

Figure 28 and Figure 29 shows the average distribution for ductile and brittle behavior of steel cables. Both distributions is presented to give a picture of the difference in strength between the ductile and brittle case. In reality, it is the same if the wires have brittle or ductile behavior, since the cable's behavior in general always displays brittle characteristic.

Table 4 shows the maximum average strength of steel cables. Here it's seen that the maximum capacity will vary with approximately 0.6 mega Newton from the case with no initial wire strain to the case with 0-1% initial wire strain. There are several reasons for why the steel cables should be modeled with different initial wire strain. The assumption that there exist no initial strain in the cable is very unlikely, because there will always be factors such as climate, manufacturing error, assembly error etc. that affect the wires.

The distribution of the initial wire strain found using a random number generator over a certain domain. The random number generator in MATLAB is as previously stated based on a uniform distribution. When enough random numbers has been generated this will mean that every value in the domain will have equally chance to occur, as seen in Figure 9. In reality, the initial wire strain will most likely not be a uniform distribution, but instead be more like a normal distribution. This can come from that the wires in a cable mostly come from the same batch, so the manufacturing error will therefore be relatively the same throughout the entire cable. Climate and assembly error will therefore be the factors that contributes to differences of strain in the wires. In most cases, these errors will led to small changes in the initial wire strain. Since there is no data for a good fit of the mean and standard deviation of a normal distribution, the uniformly distribution is used instead. The uniform distribution make it possible to show an approximation of the strength, when the initial wire strain is in a certain domain. It should immediately be pointed out that this assumption can lead to errors when the average cable capacity is found.

8.3 INTEGRATION

In reliability updating the probabilities used in the distribution functions are low compared to other engineering fields, often operating with values lower than 10^{-16}. This requires a high degree of accuracy from the software, and approximation errors in MATLAB have a great effect on the final results. As seen by the scales in Figure 35, small inaccuracies in the integration limits will have a great impact on the accuracy of the integration.

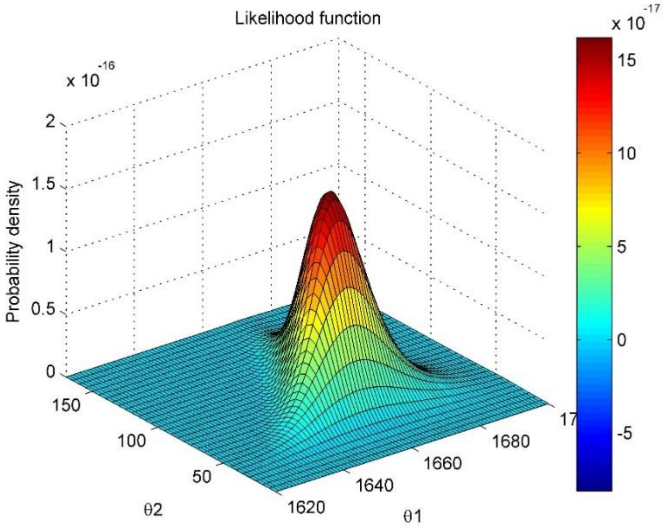


Figure 35 Likelihood distribution of θ_1 and θ_2, performing failure test

Initial estimations proved that the default accuracy in MATLAB were far too inaccurate for the demands in this report. This problem were overcome by manually modifying the number of digits MATLAB used in the estimations. However, some errors may still occur due to inaccuracies in the integration boundaries for the normalizing factor used in the predictive distribution. As far as the authors have discerned, invoking trapezoidal integration with optimized boundaries yields an accurate estimate of the normalizing factor, in the sense that the area of the predictive distribution equals one, cf. Table 5. If the integration limits are set as $\mu \pm 4\sigma$ and $\mu \pm 8\sigma$ the solution becomes unstable when only a few or none of the wires fails during the proof load. This is connected to the uncertainties of the parameters when all wires pass the proof test, the standard deviation of $\begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}$ are now $\begin{bmatrix}0.2027 & 2.8959\end{bmatrix} \cdot 10^{14}$. In this case, an integration limit of $\mu \pm 8\sigma$ is far too inaccurate.

Integration methods

Two different integration methods is used in the integration of the normalizing factor and the predictive distribution, these are:

- Trapezoidal integration
- Global adaptive quadrature rule

Appendix C – Integration accuracy show the result of how the area under the predictive graph vary for these two integration methods over a certain domain. Generally, the results using a trapezoidal integration is more consistent and correct than the ones found using the adaptive method. Since both θ_1 and θ_2 are squared in the bivariate prior distribution (equation(5.29)), the trapezoidal method will in general not give the exact answer for all functions. By decreasing the time step in the integration from 1.0 to 0.5 the numerical solution with a trapezoidal integration is exact.

8.4 UPDATED STRENGTH

A question regarding the effect of proof load testing on the strength distribution, was raised in chapter 5.2. Figure 32 and Figure 33 shows the updated strength of a cable, as a function of λ. In Figure 32 all test pass the proof load test, while in Figure 33 the updated strength of the cable, when one test fail the proof load test shown. It’s here seen that the change in strength now is much larger than in Figure 32. Therefore, its arguable that removing a few wires from the existing cable and perform failure testing on these tests will give a better estimate of the strength rather than performing proof loading.

From Figure 32 and Figure 33 it's seen that the updated strength will have a steeper curve than the prior strength. This difference will increase when the proof load level is set higher. The reason for this can be seen in Figure 30 and Figure 31. Since the predictive distribution takes into account the uncertainties of the estimated parameters, its distribution will be lower and wider than the posterior. This makes the scale factor *c*, of the Weibull distribution smaller. In equation (4.6) it's seen that the capacity of the cable uses $\lambda^{-1/c}$ as part of the equation. This will lead to an increasing error of the mean value as the value of λ increase.

8.5 RELIABILITY

In Figure 34 the reliability was shown using time invariant reliability method. It was here seen that the reliability of the wire was very large. This comes from that the calculated probability of failure was very small, as seen in Table 7. An attempt of recalculating the probability of failure was attempted using Monte Carlo simulations with Latin hypercube sampling. We were immediately not able to calculate the probability of failure with the computers at hand, because the probability is so small. This means that a very large number of simulations are needed for the probability of failure to converge, which created a memory problem on our computers. Therefore, the calculated probability of failure is found using the integral method, even if the answer seems incorrect.

It is also seen that the values for the updated reliability of the wire are less than the original estimate. In general, the updated reliability should be higher than the prior reliability at the inspection time, or else the entire procedure wasted with respect to prolonging the lifetime of the cable. The authors can think of two reasons why the reliability is lower after the updating.

1. The time invariant reliability method is used for calculating the reliability distribution. This approximation is very ruff, since the probability of failure will increase over time, instead of being a constant value as it was assumed in Figure 34. A better way of calculating the reliability is to use the time variant reliability method. This method will take into account the periodically changes in the probability of failure, but due to time issues, the time invariant method was used instead.
2. An error occurred during the calculations of the predictive posterior distribution and/or in the calculation of the posterior distribution. As seen in Table 7 the difference between the prior and posterior probability of failure is too large. This may lead to that the updated reliability always will have smaller values than the original. One remedy to this can be to calculate the probability of failure with Monte Carlo, First Order Reliability Method, etc.

9 CONCLUSION

One of the main goals of this thesis has been the creation of a working MATLAB script that asses the reliability of a bridge cable. The group was also tasked with implementing and examining both the effects of initial strain in the wires, reduction due to length effect and the reduction in strength due to Daniels effects. The predictive strength distribution were found simulating both failure load and proof load.

Daniels effect

The reduction factor due to Daniels effect are shown in Figure 27, chapter 7.1. Although the figure shows a reduction in capacity for the first 150 wires, Daniels theory is not valid until the number of wires are higher than 150. This is not an immediate problem, since the number of wires in a cable are higher than this.

Number of Wires	100 – 1000	1000 – 10 000
Capacity reduction (%)	0.946 – 0.939	0.939 – 0.937

Table 8 Daniels reduction factor

As seen in Table 8 Daniels reduction factor above, a reduction of about 6% is expected for a cable consisting of 1000 wires or less. This is also the area where the number of wires have the greatest impact on the reduction factor. If the number of wires exceed 1000, the reduction factor is relatively constant, the difference between the reduction when the cable has 1000 wires compared to 10 000 are only 0.2%. Usually, a conservative estimate around 8 % is used in the design process, as seen from Table 8, this is a conservative estimate. However, this is under the assumption that the wires are all new, and the number of system defects are low, i.e. the value of λ is low. Therefore, if the number of wires are between 100 – 1000, the effect of Daniels should be investigated further.

Initial strain

Investigating the effects of the initial stress showed that a reduction of about 0.63 MN occurs when the initial strain is in the range of 0 – 1% (see Table 4). This corresponds to a reduction of 5.36% compared to the original capacity. Since this has a large impact on the load bearing capacity of the cable, initial strain should therefore be accounted for in the design process, especially at critical design points where the concentration of stresses are high.

Reliability updating

The reliability of the system was calculated using two different proof load levels. A load level of 1620 MPa corresponds to 90% of estimated prior mean breaking load and 1660 MPa correspond to 92.8% of mean breaking load. Although the load level is higher than the one used in practical cases, the main aspects of the updating process still holds. Even when the proof load is greater than 90% of the breaking load, failing a test still gives more information with respect to the capacity of the wires, as seen in Figure 32 and Figure 33. The results of the probability of failures are rather low compared to what we expected to find, with the results that the target reliability is high and the exact results are inconclusive using the wire test data given in this paper.

9.1 FURTHER WORK

A functioning script has been made, several simplifications and assumptions were made. Although it is possible to deduct how accurate these assumptions are using simple reasoning, the effects should be investigated further. The group wish to propose the following areas for further study.

Daniels effect

Although the calculation of Daniels effect is based on the previous work in Faber et al. [1], the group was unable to gain a deeper understanding of the underlying construct of the theory. The exact effect should be investigated more thoroughly, with the goal of gaining extensive understanding of the theory and its background. Within this work, the exact long term effects of λ and its variables should be investigated further, e.g. being able to omit or reduce corrosion damage due to newer protection methods will have a great impact on λ and Daniels reduction factor.

Initial strains

In this paper the initial strains were modelled using a uniform distribution, this may not represent the real distribution and should be investigated further. The exact values and their origin is also of interest, since these are not investigated here. Especially the size and origin of these strains in older bridges and construction methods requires further investigation, e.g. through studying older Codes and papers.

Reliability updating

The reliability of the cable was calculated by finding the probability of failure, by integration of the load and resistance distributions. It is also possible to calculate the probability of failure using First Order Reliability Method (FORM), Second Order Reliability Method (SORM), Monte Carlo, etc. Implementing one of these methods in the script may give a better estimate of the probability of failure. Another aspect is that the wires are subjected to a relatively high proof load, usually around 70 – 90% of the estimated breaking strength. For older wires with several defects, this could give rise to some irreversible strains that affect the capacity.

10 SOURCES

1. Faber, M.H., S. Engelund, and R. Rackwitz, *Aspects of parallel wire cable reliability.* Structural Safety, 2003. **25**(2): p. 201-225.
2. Gimsing, N.J. and C.T. Georgakis, *Cable Supported Bridges: Concept and Design*. Third ed., Chichester, UK: Chichester, UK: John Wiley & Sons, Ltd.
3. Kroese, D.P., T. Taimre, and Z.I. Botev, *Variance Reduction*. 2011, Hoboken, NJ, USA: Hoboken, NJ, USA: John Wiley & Sons, Inc. 347-380.
4. Schneider, J., B. International Association for, and E. Structural, *Introduction to safety and reliability of structures*. 2006, International Association for Bridge and Structural Engineering: Zürich.
5. Rinne, H., *The Weibull distribution : a handbook*. 2008, Chapman & Hall/CRC: Boca Raton, Fla.
6. Haupt, R.L. and S.E. Haupt, *Introduction to Optimization*. 2004, Hoboken, NJ, USA: Hoboken, NJ, USA: John Wiley & Sons, Inc. 1-25.
7. Faber, M.H., *Risk and safety in engineering.* Lecture notes ETH Zurich, Zurich, Switzerland, 2009.
8. Daniels, H., *The statistical theory of the strength of bundles of threads. I.* Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 1945. **183**(995): p. 405-435.
9. Panik, M.J., *The Normal Distribution*. 2012, Hoboken, NJ, USA: Hoboken, NJ, USA: John Wiley & Sons, Inc. 101-121.
10. Madsen, H.O., S. Krenk, and N.C. Lind, *Methods of structural safety*. 2006, Mineola, N.Y: Dover Publications.
11. Irgens, F., *Fasthetslære*. 7. utg. ed. 2006, Trondheim: Tapir akademisk forl.
12. Wikipedia. *Stress-Strain curve*. 02.06.2015; Available from: http://en.wikipedia.org/wiki/Stress%E2%80%93strain_curve.
13. Haahr, D.M. *Introduction to Randomness and Random Numbers*. Available from: https://www.random.org/randomness/.
14. James, P., *Monte Carlo*. 2005, Chichester, UK: Chichester, UK: John Wiley & Sons, Ltd. 125-141.
15. JCSS, *The Jcss probabilistic Model Code. part lll resistance models - Structural steel*. 2001.
16. Lee Abbott, M. and J. McKinney, *Correlation*. 2013, Hoboken, NJ, USA: Hoboken, NJ, USA: John Wiley & Sons, Inc. 127-145.
17. Radzevich, S.P., *B: Elements of Coordinate System Transformations*. 2013, Oxford, UK: Oxford, UK: John Wiley & Sons Ltd. 211-224.
18. Melchers, R.E., *Structural reliability : analysis and prediction*. 2nd ed. ed. 1999, Chichester: Wiley.
19. Huynh, H.T., V.S. Lai, and I. Soumaré, *Stochastic Simulation and Applications in Finance with MATLAB ® Programs*. 2012.
20. Cheng, J. and M.J. Druzdzel. *Latin hypercube sampling in Bayesian networks*. in *FLAIRS Conference*. 2000.
21. *Classical Probability*. 2008, Princeton; Oxford: Princeton; Oxford: Princeton University Press. 75.
22. Hamada, M.S., et al., *Bayesian Reliability*. 2008, Springer Science+Business Media, LLC: New York, NY.
23. Faber, M.H., D.V. Val, and M.G. Stewart, *Proof load testing for bridge assessment and upgrading.* Engineering Structures, 2000. **22**(12): p. 1677-1689.
24. Wheatland, M.S., *Bayesian Data Analysis*. 2010: World Scientific Publishing Co. Pte. Ltd. 121-142.
25. Grigoriu, M. and W.B. Hall, *PROBABILISTIC MODELS FOR PROOF LOAD TESTING.* Journal of Structural Engineering, 1984. **110**(2): p. 260-274.
26. Diaconis, P. and D. Ylvisaker, *Conjugate priors for exponential families.* The Annals of statistics, 1979. **7**(2): p. 269-281.

27. Fink, D., *A compendium of conjugate priors.* See http://www. people. cornell. edu/pages/df36/CONJINTRnew% 20TEX. pdf, 1997: p. 46.
28. Ditlevsen, O. and H.O. Madsen, *Structural reliability methods*. Bærende konstruktioners sikkerhed. 1996, Chichester: Wiley.
29. Raiffa, H. and R. Schlaifer, *Applied statistical decision theory*. Studies in managerial economics. 1961, Boston: Division of Research, Graduate School of Business Administration, Harvard University.
30. Rackwitz, R., *Predictive distribution of strength under control.* Matériaux et Construction, 1983. **16**(4): p. 259-267.
31. Ang, A.H.S., W.H. Tang, and A.H.S. Ang, *Probability concepts in engineering : emphasis on applications in civil & environmental engineering*. 2nd ed. ed. 2007, New York: Wiley.
32. Fujikoshi, Y., V.V. Ulyanov, and R. Shimizu, *Multivariate statistics high-dimensional and large-sample approximations*. Wiley series in probability and statistics. 2010, Hoboken, N.J.: Hoboken, N.J: Wiley.
33. Wikipedia. *Rectangle method*. 26.04.2015; Available from: http://en.wikipedia.org/wiki/Rectangle_method.
34. Wikipedia. *Trapezoidal rule*. 27.03.2015; Available from: http://en.wikipedia.org/wiki/Trapezoidal_rule.
35. Björck, Å., I. Society for, and M. Applied, *Numerical methods for least squares problems*. 1996, Society for Industrial and Applied Mathematics SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104: Philadelphia, Pa.
36. Aitkin, M. and I. Aitkin, *A hybrid EM/Gauss-Newton algorithm for maximum likelihood in mixture distributions.* Statistics and Computing, 1996. **6**(2): p. 127-130.
37. Standard, N., *Eurokode 0: grunnlag for prosjektering av konstruksjoner*. 2008, Lysaker: Standard Norge. 80.
38. JCSS, *The Jcss probabilistic Model Code. Part I - Basis of design*. 2001.
39. Forbes, C., et al., *Extreme Value (Gumbel) Distribution*. 2010, Hoboken, NJ, USA: Hoboken, NJ, USA: John Wiley & Sons, Inc. 98-101.
40. Haugan, J., *Formler og tabeller*. 1992, Bekkestua: NKI-forl.

11 APPENDIX

APPENDIX A - NUMERICAL INPUT VALUES

Prior Information from Faber et al. [1].

Test	Strength [kN] (MPa)	Test	Strength [kN] (MPa)	Test	Strength [kN] (MPa)	Test	Strength [kN] (MPa)
1	69.0 (1793)	9	70.1 (1821)	17	68.7 (1785)	25	67.2 (1746)
2	67.3 (1749)	10	68.7 (1785)	18	68.9 (1790)	26	68.5 (1780)
3	67.5 (1754)	11	67.0 (1741)	19	69.8 (1814)	27	67.2 (1746)
4	68.5 (1780)	12	69.7 (1811)	20	66.9 (1738)	28	68.6 (1783)
5	67.2 (1746)	13	70.3 (1827)	21	68.1 (1770)	29	69.7 (1811)
6	68.7 (1785)	14	68.7 (1785)	22	66.9 (1738)	30	68.2 (1772)
7	68.0 (1767)	15	69.0 (1793)	23	66.5 (1728)		
8	69.0	16	67.5	24	68.0		

	(1793)		(1754)		(1767)		

Additional information

Tests	1	2	3	4	5	6	7	8
Strength [kN] (MPa)	65.0 (1690)	63.5 (1650)	62.3 (1620)	63.1 (1640)	63.8 (1657)	63.0 (1638)	64.3 (1672)	63.3 (1646)

APPENDIX B - PARTIAL SAFETY FACTOR, PROBABILITY OF FAILURE AND RELIABILITY RESULTS

Partial safety factor for a cable

Target reliability	Prior safety factor	Posterior safety factor Proof load 1620 MPa	Posterior safety factor Proof load 1660 MPa
3	1,7618	1,7634	1,7965
4	1,7952	1,7983	1,8637
5	1,8299	1,8346	1,9363

Probability of failure and reliability for proof load 1620 MPa in wire

Target reliability	Prior probability of failure proof load level 1620 MPa	Prior reliability proof load level 1620 MPa	Posterior probability of failure proof load level 1620 MPa	Posterior reliability proof load level 1620 MPa
3	1,0076*10^-16	8,2212	4,0625*10^-16	8,0523
4	8,4745*10^-17	8,2419	3,3893*10^-16	8,0744
5	7,0766*10^-17	8,2634	2,8057*10^-16	8,0975

Probability of failure and reliability for proof load 1620 MPa in cable

Target reliability	**Prior probability of failure proof load level 1620 MPa**	**Prior reliability proof load level 1620 MPa**	**Posterior probability of failure proof load level 1620 MPa**	**Posterior reliability proof load level 1620 MPa**
3	1,2305*10^-19	8,9905	4,8712*10^-19	8,838
4	9,0869*10^-20	9,0868	3,5474*10^-19	8,8734
5	6,6321*10^-20	9,0582	2,5504*10^-19	8,9100

Probability of failure and reliability for proof load 1660 MPa in wire

Target reliability	**Prior probability of failure proof load level 1660 MPa**	**Prior reliability proof load level 1660 MPa**	**Posterior probability of failure proof load level 1660 MPa**	**Posterior reliability proof load level 1660 MPa**
3	1,0076*10^-16	8,2212	3,7847*10^-5	3,9576
4	8,4745*10^-17	8,2419	2,0946*10^-5	4,0968
5	7,0766*10^-17	8,2634	1,108*10^-5	4,2419

Probability of failure and reliability for proof load 1660 MPA in cable

Target reliability	**Prior probability of failure proof load level 1660 MPa**	**Prior reliability proof load level 1660 MPa**	**Posterior probability of failure proof load level 1660 MPa**	**Posterior reliability proof load level 1660 MPa**
3	1,2305*10^-19	8,9905	9,3431*10^-13	7,0450
4	9,0867*10^-20	9,0237	4,9461*10^-13	7,1320
5	6,632*10^-20	9,0582	2,4919*10^-13	7,2257

APPENDIX C – INTEGRATION ACCURACY

Different integration methods				
Proof load	Adaptive quadrature integration Absolute error: 1e-12 Relative error: 1e-12	Adaptive quadrature integration Absolute error: 1e-8 Relative error: 1e-13	Trapezoidal integration calculation step: 1	Trapezoidal integration calculation step: 1/2
1610	1.0000	1.0000	1.0000	1.0000
1611	1.0000	1.0000	1.0000	1.0000
1612	1.0000	1.0000	1.0000	1.0000
1613	1.0000	1.0000	1.0000	1.0000
1614	1.0000	1.0000	1.0000	1.0000
1615	1.0000	1.0000	1.0000	1.0000
1616	1.0000	1.0000	1.0000	1.0000
1617	1.0000	1.0000	1.0000	1.0000
1618	1.0000	1.0000	1.0000	1.0000
1619	1.0000	1.0000	1.0000	1.0000
1620	1.0000	1.0000	1.0000	1.0000
1621	1.0000	1.0000	0.8300	1.0000
1622	1.0000	1.0000	0.8845	1.0000
1623	1.0000	1.0000	0.9331	1.0000
1624	1.0000	1.0000	0.9687	1.0000
1625	1.0000	1.0000	0.9886	1.0000
1626	1.0000	1.0000	0.9967	1.0000
1627	1.0000	1.0000	0.9993	1.0000
1628	1.0000	1.0000	0.9999	1.0000
1629	1.0000	1.0000	1.0000	1.0000
1630	1.0000	1.0000	1.0000	1.0000
1631	1.0000	1.0000	1.0000	1.0000
1632	1.0000	1.0000	1.0000	1.0000
1633	1.0000	1.0000	1.0000	1.0000
1634	1.0000	1.0000	1.0000	1.0000
1635	1.0000	1.0000	1.0000	1.0000
1636	1.0000	1.0000	1.0000	1.0000
1637	1.0000	1.0000	1.0000	1.0000
1638	1.0000	1.0000	1.0000	1.0000
1639	1.0000	1.0008	0.9991	1.0000
1640	1.0000	1.0008	0.9996	1.0000
1641	0.9999	0.9988	0.9947	1.0000
1642	0.9999	0.9988	0.9963	1.0000

1643	1.0008	0.9995	0.9974	1.0000
1644	1.0008	0.9995	0.9980	1.0000
1645	1.0009	0.9996	0.9987	1.0000
1646	1.0009	0.9996	0.9991	1.0000
1647	1.0004	1.0004	0.9976	1.0000
1648	1.0004	1.0004	0.9980	1.0000
1649	1.0004	1.0004	0.9984	1.0000
1650	1.0004	1.0004	0.9988	1.0000
1651	1.0014	1.0014	0.9985	1.0000
1652	1.0014	1.0014	0.9987	1.0000
1653	1.0014	1.0014	0.9988	1.0000
1654	1.0014	1.0014	0.9990	1.0000
1655	1.0014	1.0014	0.9991	1.0000
1656	1.0014	1.0014	0.9993	1.0000
1657	1.0014	1.0014	0.9994	1.0000
1658	0.9984	0.9984	0.9994	1.0000
1659	0.9982	0.9982	0.9994	1.0000
1660	0.9982	0.9982	0.9995	1.0000
1661	0.9982	0.9982	0.9995	1.0000
1662	0.9982	0.9982	0.9996	1.0000
1663	0.9982	0.9982	0.9996	1.0000
1664	0.9982	0.9982	0.9997	1.0000
1665	0.9982	0.9982	0.9997	1.0000
1666	0.9982	0.9982	0.9997	1.0000
1667	1.0021	1.0021	0.9997	1.0000
1668	1.0021	1.0021	0.9998	1.0000
1669	1.0021	1.0021	0.9998	1.0000
1670	1.0021	1.0021	0.9998	1.0000
1671	1.0022	1.0022	0.9998	1.0000
1672	1.0022	1.0022	0.9999	1.0000
1673	1.0002	1.0002	0.9999	1.0000
1674	1.0002	1.0002	0.9999	1.0000
1675	1.0002	1.0002	0.9999	1.0000
1676	1.0002	1.0002	0.9999	1.0000
1677	1.0002	1.0002	0.9999	1.0000
1678	1.0002	1.0002	0.9999	1.0000
1679	1.0002	1.0002	0.9999	1.0000
1680	1.0002	1.0002	0.9999	1.0000
1681	1.0002	1.0002	0.9999	1.0000
1682	1.0002	1.0002	0.9999	1.0000
1683	1.0002	1.0002	0.9999	1.0000
1684	1.0002	1.0002	0.9999	1.0000
1685	1.0002	1.0002	0.9999	1.0000

1686	1.0003	1.0003	0.9999	1.0000
1687	1.0004	1.0004	0.9999	1.0000
1688	1.0004	1.0004	0.9999	1.0000
1689	1.0002	1.0002	0.9999	1.0000
1690	1.0002	1.0002	0.9999	1.0000
1691	0.9982	0.9982	1.0000	1.0000
1692	0.9982	0.9982	1.0000	1.0000
1693	0.9982	0.9982	1.0000	1.0000
1694	0.9982	0.9982	1.0000	1.0000
1695	0.9982	0.9982	1.0000	1.0000
1696	0.9982	0.9982	1.0000	1.0000
1697	0.9982	0.9982	1.0000	1.0000
1698	0.9982	0.9982	1.0000	1.0000
1699	0.9982	0.9982	1.0000	1.0000
1700	0.9982	0.9982	1.0000	1.0000

APPENDIX D –MATLAB SCRIPT

Geometrical input, sample data and load limits

This chapter contains geometrical data, such as wirelengths, Area, E-moduls etc. Information concerning the old and new sample data are also shown here and finally information regarding the proof load limits.

```
clc; clear all;close all;
tic;
set(0,'DefaultFigureVisible','off')
format long g
```

```
n = Total number of wires in the cable
E = Modulus of elasticity
L0 = Length of test speciment
L = Total length of wires
d = Wire diameter
Area = Wire cross sectional area
l = Aging correlation factor
Lp = Correlation length
T = Testdata
```

```
n = 200;
E = 7.75e4;
L0 = 500;
L = 100e3;
d = 7;
Area=pi*d^2/4;
l = L/L0;
Lp = l*L0;

T_old=sort([69 67.3 67.5 68.5 67.2 68.7 68 69 70.1 68.7 67 ...
   69.7 70.3 68.7 69 67.5 68.7 68.9 69.8 66.9 68.1 ...
   66.9 66.5 68 67.2 68.5 67.2 68.6 69.7 ...
   68.2])*1e3/Area;
T_new = [1690 1650 1620 1640 1657 1638 1672 1646];
```

The proof load and load limits are shown below, these limits decide the stress range used in the figures. They're mainly used to simplify the calcluations in the sense that the wirestrengths are distributed around 1600 - 1800 MPa (by default).

```
proof_load = 1620;
step2 = 0.1;
load_limit_bot = 1500;
load_limit_top = 2000;
```

```
load_limit= load_limit_bot:step2:load_limit_top;
load_limit_bot2 = 1500;
load_limit_top2 = 1550;
load_limit2= load_limit_bot2:step2:load_limit_top2;
```

Prior distribution - Parameters

The following chapter contains information regarding the prior distribution based on new test data given in the vector T_new.

parmhat estimates the shape and scale parameter in the prior Weibull distribution. Phat collects these parametres in a vector.

```
parmhat=wblfit(T_old);
u_old=parmhat(1,1);
k_old=parmhat(1,2);
phat=[u_old k_old];
```

Calculates the covariance matrix using the script 'wblpdf', where the diagonal are the uncertainties (standard deviation) of the parameters.

```
covmatrix_prior=mlecov(phat,T_old,'pdf',@wblpdf);
uncertanties_prior = sqrt(diag(covmatrix_prior));
sigma_Uprior = uncertanties_prior(1,1);
sigma_Kprior = uncertanties_prior(2,1);
```

MEAN and VARIANCE of a Weibull distribution, A_old and B_old respectively. And the correlation between the shape and scale parameter, later used in the multivariate normal distribution.
Finally the prior weibullPDF are calculated with a range equal to 'load_limit' and shape and scale parameter 'u_old' and 'k_old'.

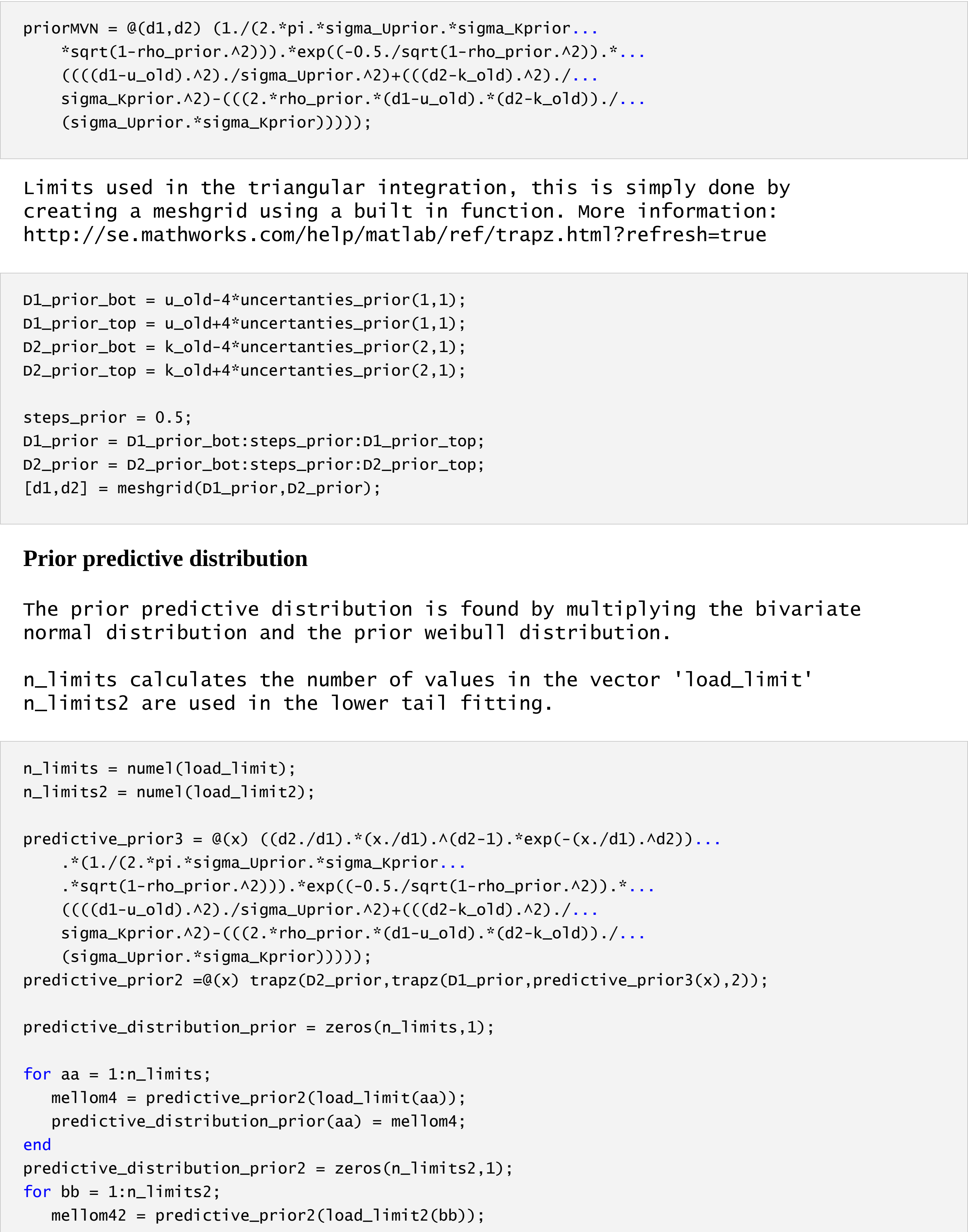
```
mu1 = [u_old k_old];
[A_old,B_old]=wblstat(u_old,k_old);
sigma3=sqrt(B_old);
rho_prior = covmatrix_prior(1,2)/(sigma_Uprior*sigma_Kprior);

prior_distribution = wblpdf(load_limit,u_old,k_old);
```

Multivariate normal density function and integration limits

In this case, it is a Bivariate density function given as:

$$\text{priorMVN} = f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} exp\left(- \frac{1}{2(1-\rho^2)^2}\left[\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}\right]\right)$$



```
priorMVN = @(d1,d2) (1./(2.*pi.*sigma_Uprior.*sigma_Kprior...
    *sqrt(1-rho_prior.^2))).*exp((-0.5./sqrt(1-rho_prior.^2)).*...
    ((((d1-u_old).^2)./sigma_Uprior.^2)+(((d2-k_old).^2)./...
    sigma_Kprior.^2)-(((2.*rho_prior.*(d1-u_old).*(d2-k_old))./...
    (sigma_Uprior.*sigma_Kprior)))));
```

Limits used in the triangular integration, this is simply done by creating a meshgrid using a built in function. More information: http://se.mathworks.com/help/matlab/ref/trapz.html?refresh=true

```
D1_prior_bot = u_old-4*uncertanties_prior(1,1);
D1_prior_top = u_old+4*uncertanties_prior(1,1);
D2_prior_bot = k_old-4*uncertanties_prior(2,1);
D2_prior_top = k_old+4*uncertanties_prior(2,1);

steps_prior = 0.5;
D1_prior = D1_prior_bot:steps_prior:D1_prior_top;
D2_prior = D2_prior_bot:steps_prior:D2_prior_top;
[d1,d2] = meshgrid(D1_prior,D2_prior);
```

Prior predictive distribution

The prior predictive distribution is found by multiplying the bivariate normal distribution and the prior weibull distribution.

n_limits calculates the number of values in the vector 'load_limit' n_limits2 are used in the lower tail fitting.

```
n_limits = numel(load_limit);
n_limits2 = numel(load_limit2);

predictive_prior3 = @(x) ((d2./d1).*(x./d1).^(d2-1).*exp(-(x./d1).^d2))...
    .*(1./(2.*pi.*sigma_Uprior.*sigma_Kprior...
    .*sqrt(1-rho_prior.^2))).*exp((-0.5./sqrt(1-rho_prior.^2)).*...
    ((((d1-u_old).^2)./sigma_Uprior.^2)+(((d2-k_old).^2)./...
    sigma_Kprior.^2)-(((2.*rho_prior.*(d1-u_old).*(d2-k_old))./...
    (sigma_Uprior.*sigma_Kprior)))));
predictive_prior2 =@(x) trapz(D2_prior,trapz(D1_prior,predictive_prior3(x),2));

predictive_distribution_prior = zeros(n_limits,1);

for aa = 1:n_limits;
   mellom4 = predictive_prior2(load_limit(aa));
   predictive_distribution_prior(aa) = mellom4;
end
predictive_distribution_prior2 = zeros(n_limits2,1);
for bb = 1:n_limits2;
   mellom42 = predictive_prior2(load_limit2(bb));
   predictive_distribution_prior2(bb) = mellom42;
end
```

maxi3 estimates the shape and scale parameter of the predictive weibull distribution using the combination of the two tha minimize the predictive distribution, i.e. the combination that gives the lowest strength.

u and k denotes the scale and shape parameters respectively, while A and B denotes the mean and variance.

```
[maxi3] = fminsearch(@(x) sum((predictive_distribution_prior'-...
    ((x(2)./x(1)).*((load_limit./x(1)).^(x(2)-1))....
    .*exp(-((load_limit./x(1)).^x(2))))).^2),[1780,70]);

u_old_pred = maxi3(1,1);
k_old_pred = maxi3(1,2);
[A_old_pred,B_old_pred] = wblstat(u_old_pred,k_old_pred);
```

Figure 1: Prior predictive distribution

Figure 1 compares the predictive prior with the approximated prior distribution, where the load limit are set as x-axis.

```
figure
hold on;
plot(load_limit,predictive_distribution_prior,'r','Linewidth',1.5);
plot(load_limit,wblpdf(load_limit,u_old_pred,k_old_pred),'k','Linewidth',1.5);
legend('Predictive prior','approximated prior');
xlabel('load level');
ylabel('PDF')
hold off;
```

Proof loading

This chapter cheks wheter or not a wire pass the proof load, if a wire fails it is considered broken and collected in the vector 'failed_tests' where the failure strengths are sampled.

tom = Samples the VALUE for the broken (failed) wires, if a wire pass the proof load, the strength of the wire are set = proof load

tom2 = Sampling vector for the NUMBER of passed/failed tests, passed tests are given a value = 1 and failed a value = 0.

n_new = Number of new test samples

```
n_new = numel(T_new);
tom = zeros(n_new,1);
tom2 = zeros(n_new,1);
```

The loop carries out the steps explained above.

```
for ii =1:n_new;
   if T_new(ii)>= proof_load;
       har = proof_load;
   elseif T_new(ii)<proof_load;
       har = T_new(ii);
   end
   tom(ii) = har;

   if T_new(ii)>= proof_load;
       har2 = 1;
   elseif T_new(ii)<proof_load;
       har2 = 0;
   end
   tom2(ii) = har2;

end
```

h = Collects the NUMBER of tests that have passed the proof load
failed_tests = Collects the VALUE of the failed wires
m = NUMBER of failed wires

```
h = sum(tom2);
failed_tests = tom(tom<proof_load);
m = numel(failed_tests);
```

Fisher information matrix

This chapter calculates the Fisher matrix for the given Weibull distribution. The first part estimates the Fisher matrix with respect to the PDF, and the last part with respect to the CDF.
The Fisher matrix consists of the double derivatives on the diagonal, and the partial derivatives of u and k.
In this case Fisher is a 2x2 matrix.

Fisher with respect to log-Weibull CDF:

```
syms u k p w
f1 = w.*log(1-(1-exp(-(p./u).^k)));
```

Elements in Fisher matrix:
b12 = Fisher(1,1)
c12 = Fisher(2,2)
e12 = Fisher(1,2)
h12 = Fisher(2,1)

```
b12(u,k,p,w) =  diff(f1,u,2);
c12(u,k,p,w) =  diff(f1,k,2);
d12(u,k,p,w) =  diff(f1,u);
e12(u,k,p,w) =  diff(d12,k);
g12(u,k,p,w) =  diff(f1,k);
h12(u,k,p,w) =  diff(g12,u);
```

This simply allows MATLAB to recongnize the elements calculated above as functions with specified variables, which again allows 'partij' to be the content of the Fisher Matrix with the given set of variables u,k,p and w.

```
pst11 = matlabFunction(b12);
pst21 = matlabFunction(e12);
pst31 = matlabFunction(h12);
pst41 = matlabFunction(c12);

part11 = @(u,k,p,w) pst11(u,k,p,w);
part21 = @(u,k,p,w) pst21(u,k,p,w);
part31 = @(u,k,p,w) pst31(u,k,p,w);
part41 = @(u,k,p,w) pst41(u,k,p,w);
```

Fisher with respect to log-Weibull PDF:

```
syms x u k
f3 = log((k./u).*(x./u).^(k-1).*exp(-(x./u).^k));
```

Elements in Fisher matrix:
b3 = Fisher(1,1)
c3 = Fisher(2,2)
e3 = Fisher(1,2)
h3 = Fisher(2,1)

```
b3(x,u,k) = diff(f3,u,2);
c3(x,u,k) = diff(f3,k,2);
d3(x,u,k) = diff(f3,u);
e3(x,u,k) = diff(d3,k);
g3(x,u,k) = diff(f3,k);
h3(x,u,k) = diff(g3,u);
```

This simply allows MATLAB to recongnize the elements calculated above as functions with specified variables, which again allows 'partij' to be the content of the Fisher Matrix with the given set of variables u,k,p and w.

```
pst13 = matlabFunction(b3);
pst23 = matlabFunction(e3);
pst33 = matlabFunction(h3);
pst43 = matlabFunction(c3);



part13 = @(x,u,k) pst13(x,u,k);
part23 = @(x,u,k) pst23(x,u,k);
part33 = @(x,u,k) pst33(x,u,k);
part43 = @(x,u,k) pst43(x,u,k);
```

Updating process

The calculation steps are identical with the following difference:
1. part assume that all wire fails
2. part that one wire fail
3. part that all wires pass the proof load

If all wires fail

The calculation steps in this chapter are given in the following order:
1) Nelder-Mead simplex to optimize the values of Theta (shape and scale facor)
2) Estimating the Fisher- and Covariance Matrix based on these values
3) The likelihood function
4) Normalizing factor
5) Predictive distribution

```
if m==n_new;
```

1) Initially, the Nelder-Mead simplex optimize the values in the variables Theta in the log-likehood function. These variables are denoted theta1 and theta2 and maximize the log-likelihood function.

```
[maxi] = fminsearch(@(x) -(sum(log(wblpdf(failed_tests,x(1),...
                            x(2))))),[1600,50]);


theta1 = maxi(1,1);
theta2 = maxi(1,2);
```

% 2) Fisher- and Covariance-matrix

elementij express the different elements in the Fisher matrix using the parameters without any numerical values.

```
element11 = @(x,u,k) part13(x,u,k);
element12 = @(x,u,k) part23(x,u,k);
element21 = @(x,u,k) part33(x,u,k);
element22 = @(x,u,k) part43(x,u,k);
```

leddij calculates the numerical values in the Fisher matrix as a sum based on elementij given above and the following parameters:

x = strength of the failed tests = failed_tests
u = scale factor = theta1
k = shape factor = theta2

```
ledd11 = -sum(element11(failed_tests,theta1,theta2));
ledd12 = -sum(element12(failed_tests,theta1,theta2));
ledd21 = -sum(element21(failed_tests,theta1,theta2));
ledd22 = -sum(element22(failed_tests,theta1,theta2));
```

The covariancematrix is defined as the inverse Fisher matrix and uncertainties_new is the standard deviation of the parameters theta1 and theta2 which describe the scale and shape parameters of a Weibull distribution.

```
Fisher_matrix = [ledd11 ledd12; ledd21 ledd22];
covmatrix_new = inv((Fisher_matrix));
uncertanties_new = sqrt(diag(covmatrix_new));
```

% 3) The log-likekilhood function given with the following parameters:

x = Wirestrength = failed_tests
d1 = scale factor = theta1 above
d2 = shape factor = theta2 above

The parameters d1 and d2 denotes the same scale and shape parameters as earlier, but requires a different annotation here due to the calculation methods used in the posterior distribution.

```
syms d1 d2
pdf_part_of_log_likelihood = @(x) (log((d2./d1).*...
    (x./d1).^(d2-1).*exp(-(x./d1).^d2)));
sum_of_the_pdf_part(d1,d2) = sum(pdf_part_of_log_likelihood(failed_tests));
```

ht simply creates a function that MATLAB recognizes, here with the variables d1 and d2. Simply known as the shape and scale parameters.

```
ht = matlabFunction(sum_of_the_pdf_part);
best = ht(d1,d2);
```

The final log-likelihood and the likelihood functions

```
log_likelihood = @(d1,d2) ht(d1,d2);
likelihood = @(d1,d2) exp(log_likelihood(d1,d2));
```

4) Normalizing factor Using Nelder-Mead simplex to calculate the theta1 and theta2 that maximize the probability of failure, where theta1 and 2 are denoted midd1 and 2.

```
fun_normalizing = @(d1,d2) priorMVN(d1,d2).*likelihood(d1,d2);
maximizing_normalizing = @(x)-fun_normalizing(x(1),x(2));
[maxi2] = fminsearch(maximizing_normalizing,[1780,50]);

midd_1 = maxi2(1,1);
midd_2 = maxi2(1,2);
```

Assuming that the curve isn't wider than +/- 500 to the center value, the preliminary integration boundaries are found. The If sentece below secures that the boundaries are positive.

```
new_theta1bot = midd_1-500;
new_theta1top = midd_1+500;
new_theta2bot = midd_2-500;
new_theta2top = midd_2+500;
if new_theta2bot<=0;
    new_theta2bot=1;
elseif new_theta2bot>0;
    new_theta2bot = midd_2-500;
end
```

Following the assumptions above, the integration range for theta1 and theta2 are found.

```
range_theta1 =new_theta1bot:new_theta1top;
steps_theta1 = numel(range_theta1);
range_theta2 =new_theta2bot:new_theta2top;
steps_theta2 = numel(range_theta2);
```

The integration boundaries are estimated with a given stopping criterion found in the loop below.

```
jobb = zeros(steps_theta2,steps_theta1);
for aa = 1:steps_theta1;
    for bb=1:steps_theta2;
        fa = fun_normalizing(midd_1,midd_2)-...
             fun_normalizing(range_theta1(aa),range_theta2(bb));
        jobb(bb,aa) = fa;
    end
end
sa = jobb;
```

The row and column containing the minimum (first) and maximum (last) values for theta1 and 2 are found using the built in function in MATLAB.

```
[row1,col1] = find(sa<fun_normalizing(midd_1,midd_2),1,'first');
[row2,col2] = find(sa<fun_normalizing(midd_1,midd_2),1,'last');
```

The final integration boundaries for theta1 and 2 are given as a square, where the slope of the 3D plot of theta1 and 2 decides the upper and lower boundaries.

```
D1_bot = new_theta1bot+col1-3;
if D1_bot<0;
    D1_bot = 0;
elseif D1_bot>=0;
    D1_bot = new_theta1bot+col1-3;
end
D1_top = new_theta1bot+col2+3;
D2_bot = new_theta2bot+row1-3;
if D2_bot <0;
    D2_bot = 0;
elseif D2_bot>=0;
    D2_bot = new_theta2bot+row1-3;
end
D2_top = new_theta2bot+row2+3;
```

The boundaries found above are implemented in MATLAB as a meshgrid using the values for D1 and D2.

```
step = 1/2;
D1 = D1_bot:step:D1_top;
D2 = D2_bot:step:D2_top;
[d1,d2] = meshgrid(D1,D2);
```

Tass is the log-likelihood function with parameters d1 and d2 used to calculate the normalizing factor (P) for the posterior and predictive distribution. The normalizing factor (P) is given by the folliwing equation:

$$P = \frac{}{\int^{\theta} f' L(\theta|\hat{x}d\theta}$$

```
tass = ht(d1,d2);
normalizing = exp(tass).*...
    (1./(2.*pi.*sigma_Uprior.*sigma_Kprior...
    *sqrt(1-rho_prior.^2))).*exp((-0.5./sqrt(1-rho_prior.^2)).*...
    ((((d1-u_old).^2)./sigma_Uprior.^2)+(((d2-k_old).^2)./...
    sigma_Kprior.^2)-(((2.*rho_prior.*(d1-u_old).*(d2-k_old))./...
    (sigma_Uprior.*sigma_Kprior)))));
normalizing_factor2 = trapz(D2,trapz(D1,normalizing,2));
P = 1/normalizing_factor2;
```

Predictive

5) predictive distribution The predictive distribution is given by the following equaiton:

$$f'''(X|\theta) = f(X|\theta)f''(\theta)$$

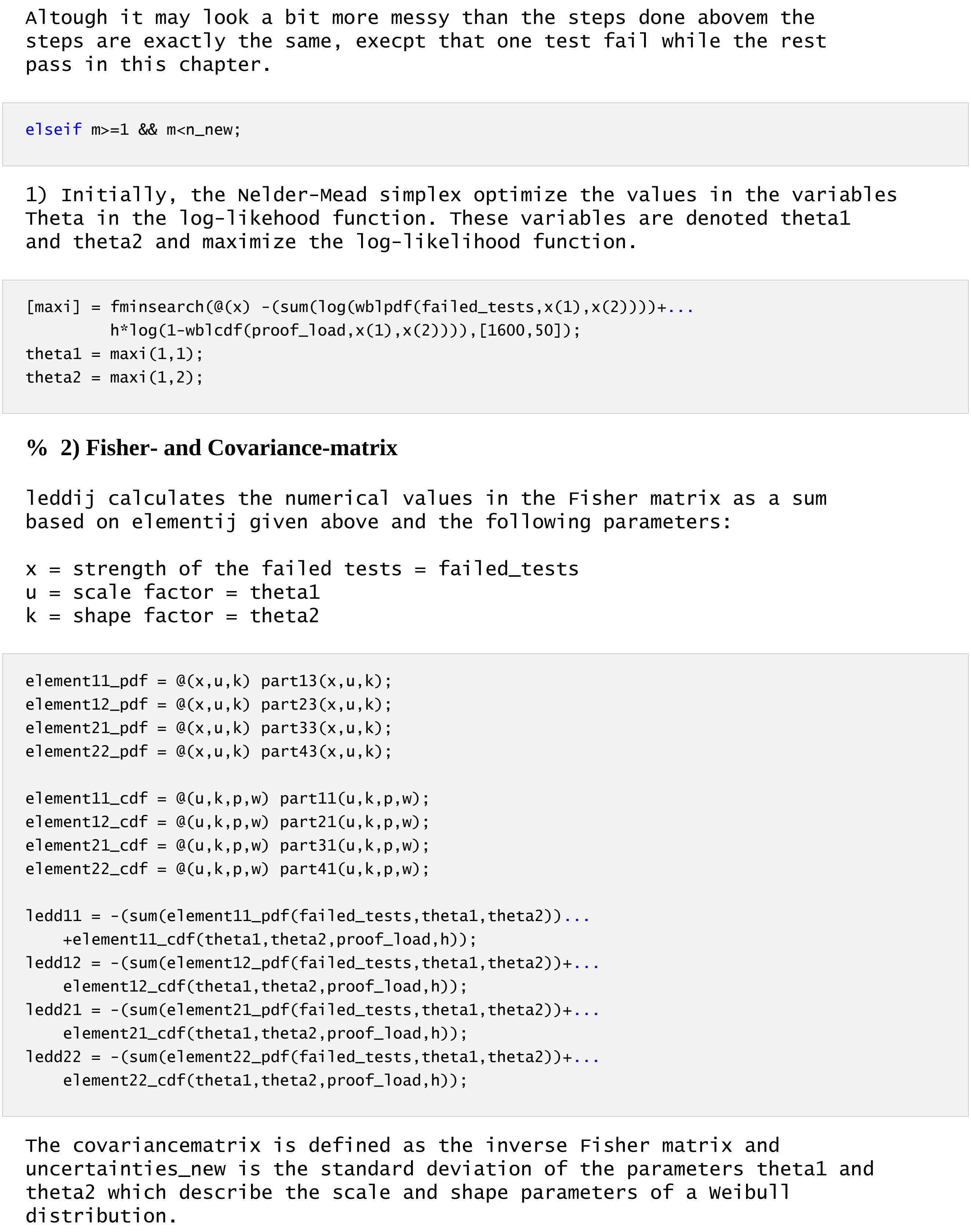
By using the integration boundaries above, the predictive distribution is estimated (predictive2), running a loop from 1 to n_limits numerical values are found and inserted into the predictive distribution.

```
predictive3 =@(x) ((d2./d1).*(x./d1).^(d2-1).*exp(-(x./d1).^d2)).*P.*...
    exp(tass).*(1./(2.*pi.*sigma_Uprior.*sigma_Kprior...
    *sqrt(1-rho_prior.^2))).*exp((-0.5./sqrt(1-rho_prior.^2)).*...
    ((((d1-u_old).^2)./sigma_Uprior.^2)+(((d2-k_old).^2)./...
    sigma_Kprior.^2)-(((2.*rho_prior.*(d1-u_old).*(d2-k_old))./...
    (sigma_Uprior.*sigma_Kprior)))));

predictive2 =@(x) trapz(D2,trapz(D1,predictive3(x),2));
predictive_distribution = zeros(n_limits,1);
for ee = 1:n_limits;
   mellom = predictive2(load_limit(ee));
   predictive_distribution(ee) = mellom;
end
```

When 1 test fail

The calculation steps in this chapter are given in the following order:
1) Nelder-Mead simplex to optimize the values of Theta (shape and scale facor)
2) Estimating the Fisher- and Covariance Matrix based on these values
3) The likelihood function
4) Normalizing factor
5) Predictive distribution



Altough it may look a bit more messy than the steps done abovem the steps are exactly the same, execpt that one test fail while the rest pass in this chapter.

```
elseif m>=1 && m<n_new;
```

1) Initially, the Nelder-Mead simplex optimize the values in the variables Theta in the log-likehood function. These variables are denoted theta1 and theta2 and maximize the log-likelihood function.

```
[maxi] = fminsearch(@(x) -(sum(log(wblpdf(failed_tests,x(1),x(2))))+...
         h*log(1-wblcdf(proof_load,x(1),x(2)))),[1600,50]);
theta1 = maxi(1,1);
theta2 = maxi(1,2);
```

% 2) Fisher- and Covariance-matrix

leddij calculates the numerical values in the Fisher matrix as a sum based on elementij given above and the following parameters:

x = strength of the failed tests = failed_tests
u = scale factor = theta1
k = shape factor = theta2

```
element11_pdf = @(x,u,k) part13(x,u,k);
element12_pdf = @(x,u,k) part23(x,u,k);
element21_pdf = @(x,u,k) part33(x,u,k);
element22_pdf = @(x,u,k) part43(x,u,k);

element11_cdf = @(u,k,p,w) part11(u,k,p,w);
element12_cdf = @(u,k,p,w) part21(u,k,p,w);
element21_cdf = @(u,k,p,w) part31(u,k,p,w);
element22_cdf = @(u,k,p,w) part41(u,k,p,w);

ledd11 = -(sum(element11_pdf(failed_tests,theta1,theta2))...
    +element11_cdf(theta1,theta2,proof_load,h));
ledd12 = -(sum(element12_pdf(failed_tests,theta1,theta2))+...
    element12_cdf(theta1,theta2,proof_load,h));
ledd21 = -(sum(element21_pdf(failed_tests,theta1,theta2))+...
    element21_cdf(theta1,theta2,proof_load,h));
ledd22 = -(sum(element22_pdf(failed_tests,theta1,theta2))+...
    element22_cdf(theta1,theta2,proof_load,h));
```

The covariancematrix is defined as the inverse Fisher matrix and uncertainties_new is the standard deviation of the parameters theta1 and theta2 which describe the scale and shape parameters of a Weibull distribution.

```
Fisher_matrix = [ledd11 ledd12; ledd21 ledd22];
covmatrix_new = inv((Fisher_matrix));
uncertanties_new = sqrt(diag(covmatrix_new));
```

% 3) The log-likekilhood function given with the following parameters:

x = Wirestrength = failed_tests
d1 = scale factor = theta1 above
d2 = shape factor = theta2 above

The parameters d1 and d2 denotes the same scale and shape parameters as earlier, but requires a different annotation here due to the calculation methods used in the posterior distribution.

```
syms d1 d2
pdf_part_of_log_likelihood = @(x) (log((d2./d1).*...
    (x./d1).^(d2-1).*exp(-(x./d1).^d2)));
sum_of_the_pdf_part(d1,d2) = sum(pdf_part_of_log_likelihood(failed_tests));
```

ht simply creates a function that MATLAB recognizes, here with the variables d1 and d2. Simply known as the shape and scale parameters.

```
ht = matlabFunction(sum_of_the_pdf_part);
cdf_part_of_log_likelihood = @(d1,d2) h.*log(exp(-(proof_load./d1).^d2));
```

The final log-likelihood and the likelihood functions

```
log_likelihood = @(d1,d2) (ht(d1,d2)+cdf_part_of_log_likelihood(d1,d2));
likelihood = @(d1,d2) exp(log_likelihood(d1,d2));
```

4) Normalizing factor Using Nelder-Mead simplex to calculate the theta1 and theta2 that maximize the probability of failure, where theta1 and 2 are denoted middl and 2.

```
fun_normalizing = @(d1,d2) priorMVN(d1,d2).*likelihood(d1,d2);
maximizing_normalizing = @(x)-fun_normalizing(x(1),x(2));
[maxi2] = fminsearch(maximizing_normalizing,[1780,50]);

midd_1 = maxi2(1,1);
midd_2 = maxi2(1,2);
```

Assuming that the curve isn't wider than +/- 500 to the center value, the preliminary integration boundaries are found. The If sentece below secures that the boundaries are positive.

```
new_theta1bot = midd_1-500;
new_theta1top = midd_1+500;
new_theta2bot = midd_2-500;
new_theta2top = midd_2+500;

if new_theta2bot<=0;
    new_theta2bot=1;
elseif new_theta2bot>0;
    new_theta2bot = midd_2-500;
end
```

Following the assumptions above, the integration range for theta1 and theta2 are found.

```
range_theta1 =new_theta1bot:new_theta1top;
steps_theta1 = numel(range_theta1);
range_theta2 =new_theta2bot:new_theta2top;
steps_theta2 = numel(range_theta2);
```

The integration boundaries are estimated with a given stopping criterion found in the loop below.

```
jobb = zeros(steps_theta2,steps_theta1);
for aa = 1:steps_theta1;
    for bb=1:steps_theta2;
        fa = fun_normalizing(midd_1,midd_2)-...
             fun_normalizing(range_theta1(aa),range_theta2(bb));
        jobb(bb,aa) = fa;
    end
end
sa = jobb;
```

The row and column containing the minimum (first) and maximum (last) values for theta1 and 2 are found using the built in function in MATLAB.

```
[row1,col1] = find(sa<fun_normalizing(midd_1,midd_2),1,'first');
[row2,col2] = find(sa<fun_normalizing(midd_1,midd_2),1,'last');
```

The final integration boundaries for theta1 and 2 are given as a square, where the slope of the 3D plot of theta1 and 2 decides the upper and lower boundaries.

```
D1_bot = new_theta1bot+col1-3;
if D1_bot<0;
    D1_bot = 0;
elseif D1_bot>=0;
```

```
    D1_bot = new_theta1bot+col1-3;
end
D1_top = new_theta1bot+col2+3;
D2_bot = new_theta2bot+row1-3;
if D2_bot <0;
    D2_bot = 0;
elseif D2_bot>=0;
    D2_bot = new_theta2bot+row1-3;
end
D2_top = new_theta2bot+row2+3;
```

The boundaries found above are implemented in MATLAB as a meshgrid using the values for D1 and D2.

```
step = 1/2;
D1 = D1_bot:step:D1_top;
D2 = D2_bot:step:D2_top;
[d1,d2] = meshgrid(D1,D2);
```

Tass is the log-likelihood function with parameters d1 and d2 used to calculate the normalizing factor (P) for the posterior and predictive distribution. The normalizing factor (P) is given by the folliwing equation:

$$P = \frac{}{\int^{\theta} f' L(\theta|\hat{x} d\theta}$$

```
tass = ht(d1,d2);
normalizing = exp(h.*log(exp(-(proof_load./d1).^d2))+tass).*...
    (1./(2.*pi.*sigma_Uprior.*sigma_Kprior...
    *sqrt(1-rho_prior.^2))).*exp((-0.5./sqrt(1-rho_prior.^2)).*...
    ((((d1-u_old).^2)./sigma_Uprior.^2)+(((d2-k_old).^2)./...
    sigma_Kprior.^2)-(((2.*rho_prior.*(d1-u_old).*(d2-k_old))./...
    (sigma_Uprior.*sigma_Kprior)))));
normalizing_factor2 = trapz(D2,trapz(D1,normalizing,2));
P = 1/normalizing_factor2;
```

Predictive

5) predictive distribution The predictive distribution is given by the following equaiton:

$$f'''(X|\theta) = f(X|\theta) f''(\theta)$$

By using the integration boundaries above, the predictive distribution is estimated (predictive2), running a loop from 1 to n_limits numerical values are found and inserted into the predictive distribution.

```
predictive3 =@(x) ((d2./d1).*(x./d1).^(d2-1).*exp(-(x./d1).^d2)).*P.*...
    exp(h.*log(exp(-(proof_load./d1).^d2))+tass).*...
    (1./(2.*pi.*sigma_Uprior.*sigma_Kprior...
    *sqrt(1-rho_prior.^2))).*exp((-0.5./sqrt(1-rho_prior.^2)).*...
    ((((d1-u_old).^2)./sigma_Uprior.^2)+(((d2-k_old).^2)./...
    sigma_Kprior.^2)-(((2.*rho_prior.*(d1-u_old).*(d2-k_old))./...
    (sigma_Uprior.*sigma_Kprior)))));
predictive2 =@(x) trapz(D2,trapz(D1,predictive3(x),2));
predictive_distribution = zeros(n_limits,1);
for ee = 1:n_limits;
   mellom = predictive2(load_limit(ee));
   predictive_distribution(ee) = mellom;
end
```

When all wires pass the proof load

The calculation steps in this chapter are given in the following order:
1) Nelder-Mead simplex to optimize the values of Theta (shape and scale facor)
2) Estimating the Fisher- and Covariance Matrix based on these values
3) The likelihood function
4) Normalizing factor
5) Predictive distribution

```
elseif m<1;
```

1) Initially, the Nelder-Mead simplex optimize the values in the variables Theta in the log-likehood function. These variables are denoted theta1 and theta2 and maximize the log-likelihood function.

```
[maxi] = fminsearch(@(x) -(h*log(1-(1-exp(-(proof_load./x(1)).^x(2))))),...
                    [1620,1200]);

theta1 = maxi(1,1);
theta2 = maxi(1,2);
```

% 2) Fisher- and Covariance-matrix

elementij express the different elements in the Fisher matrix using the parameters without any numerical values.

```
element11 = @(u,k,p,w) part11(u,k,p,w);
element12 = @(u,k,p,w) part21(u,k,p,w);
element21 = @(u,k,p,w) part31(u,k,p,w);
element22 = @(u,k,p,w) part41(u,k,p,w);
```

leddij calculates the numerical values in the Fisher matrix as a sum based on elementij given above and the following parameters:

x = strength of the failed tests = failed_tests
u = scale factor = theta1
k = shape factor = theta2

```
ledd11 = -element11(theta1,theta2,proof_load,h);
ledd12 = -element12(theta1,theta2,proof_load,h);
ledd21 = -element21(theta1,theta2,proof_load,h);
ledd22 = -element22(theta1,theta2,proof_load,h);
```

The covariancematrix is defined as the inverse Fisher matrix and uncertainties_new is the standard deviation of the parameters theta1 and theta2 which describe the scale and shape parameters of a Weibull distribution.

```
Fisher_matrix = [ledd11 ledd12; ledd21 ledd22];
covmatrix_new = (inv((Fisher_matrix)));
uncertanties_new = sqrt(diag(covmatrix_new));
```

% 3) The log-likekilhood function given with the following parameters:

x = Wirestrength = failed_tests
d1 = scale factor = theta1 above
d2 = shape factor = theta2 above

The parameters d1 and d2 denotes the same scale and shape parameters as earlier, but requires a different annotation here due to the calculation methods used in the posterior distribution.

```
log_likelihood = @(d1,d2) h*log(1-(1-exp(-(proof_load./d1).^d2)));
likelihood = @(d1,d2) exp(log_likelihood(d1,d2));
```

4) Normalizing factor Using Nelder-Mead simplex to calculate the theta1 and theta2 that maximize the probability of failure, where theta1 and 2 are denoted midd1 and 2.

```
fun_normalizing = @(d1,d2) priorMVN(d1,d2).*likelihood(d1,d2);
maximizing_normalizing = @(x)-fun_normalizing(x(1),x(2));
[maxi2] = fminsearch(maximizing_normalizing,[1780,50]);

midd_1 = maxi2(1,1);
midd_2 = maxi2(1,2);
```

Assuming that the curve isn't wider than +/- 500 to the center value, the preliminary integration boundaries are found. The If sentece below secures that the boundaries are positive.

```
new_theta1bot = midd_1-500;
new_theta1top = midd_1+500;
new_theta2bot = midd_2-500;
new_theta2top = midd_2+500;
if new_theta2bot<=0;
    new_theta2bot=1;
elseif new_theta2bot>0;
    new_theta2bot = midd_2-500;
end
```

Following the assumptions above, the integration range for theta1 and theta2 are found.

```
range_theta1 =new_theta1bot:new_theta1top;
steps_theta1 = numel(range_theta1);
range_theta2 =new_theta2bot:new_theta2top;
steps_theta2 = numel(range_theta2);
```

The integration boundaries are estimated with a given stopping criterion found in the loop below.

```
jobb = zeros(steps_theta2,steps_theta1);
for aa = 1:steps_theta1;
    for bb=1:steps_theta2;
        fa = fun_normalizing(midd_1,midd_2)-...
             fun_normalizing(range_theta1(aa),range_theta2(bb));
        jobb(bb,aa) = fa;
    end
end
sa = jobb;
```

The row and column containing the minimum (first) and maximum (last) values for theta1 and 2 are found using the built in function in MATLAB.

```
[row1,col1] = find(sa<fun_normalizing(midd_1,midd_2),1,'first');
[row2,col2] = find(sa<fun_normalizing(midd_1,midd_2),1,'last');
```

The final integration boundaries for theta1 and 2 are given as a square, where the slope of the 3D plot of theta1 and 2 decides the upper and lower boundaries.

```
D1_bot = new_theta1bot+col1-3;
if D1_bot<0;
    D1_bot = 0;
elseif D1_bot>=0;
    D1_bot = new_theta1bot+col1-3;
end
D1_top = new_theta1bot+col2+3;
D2_bot = new_theta2bot+row1-3;
if D2_bot <0;
    D2_bot = 0;
elseif D2_bot>=0;
    D2_bot = new_theta2bot+row1-3;
end
D2_top = new_theta2bot+row2+3;
```

The boundaries found above are implemented in MATLAB as a meshgrid using the values for D1 and D2.

```
step = 1/2;
D1 = D1_bot:step:D1_top;
D2 = D2_bot:step:D2_top;
[d1,d2] = meshgrid(D1,D2);
```

Tass is the log-likelihood function with parameters d1 and d2 used to calculate the normalizing factor (P) for the posterior and predictive distribution. The normalizing factor (P) is given by the folliwing equation:

$$P = \frac{}{\int^{\theta} f' L(\theta|\hat{x} d\theta}$$

```
normalizing = exp(h*log(1-(1-exp(-(proof_load./d1).^d2)))).*...
    (1./(2.*pi.*sigma_Uprior.*sigma_Kprior...
    *sqrt(1-rho_prior.^2))).*exp((-0.5./sqrt(1-rho_prior.^2)).*...
    ((((d1-u_old).^2)./sigma_Uprior.^2)+(((d2-k_old).^2)./...
    sigma_Kprior.^2)-(((2.*rho_prior.*(d1-u_old).*(d2-k_old))./...
    (sigma_Uprior.*sigma_Kprior)))));
normalizing_factor2 = trapz(D2,trapz(D1,normalizing,2));
P = 1/normalizing_factor2;
```

Predictive

5) predictive distribution The predictive distribution is given by the following equaiton:

$$f'''(X|\theta) = f(X|\theta)f''(\theta)$$

By using the integration boundaries above, the predictive distribution is estimated (predictive2), running a loop from 1 to n_limits numerical values are found and inserted into the predictive distribution.

```
predictive3 =@(x) ((d2./d1).*(x./d1).^(d2-1).*exp(-(x./d1).^d2)).*P.*...
    exp(h*log(1-(1-exp(-(proof_load./d1).^d2)))).*...
    (1./(2.*pi.*sigma_Uprior.*sigma_Kprior...
    *sqrt(1-rho_prior.^2))).*exp((-0.5./sqrt(1-rho_prior.^2)).*...
    ((((d1-u_old).^2)./sigma_Uprior.^2)+(((d2-k_old).^2)./...
    sigma_Kprior.^2)-(((2.*rho_prior.*(d1-u_old).*(d2-k_old))./...
    (sigma_Uprior.*sigma_Kprior)))));
predictive2 =@(x) trapz(D2,trapz(D1,predictive3(x),2));
predictive_distribution = zeros(n_limits,1);
for ee = 1:n_limits;
   mellom = predictive2(load_limit(ee));
   predictive_distribution(ee) = mellom;
end

end
sa = sum(predictive_distribution);
```

Mean and Standard deviation of likelihood function

The mean, standard deviation and correlation between the updated parameters are given below:

```
mu_U_new = theta1;
mu_K_new = theta2;
mu_new = [mu_U_new mu_K_new];

sigmaU2 = uncertanties_new(1,1);
sigmaK2 = uncertanties_new(2,1);

rho2 = covmatrix_new(1,2)/(sigmaU2*sigmaK2);
```

Updated Weibull parameters

The updated scale and shape parameters (u_new and k_new) are found using Nelder-Mead Simplex. Utilizing wblstat the mean and variance of the updated distribution are shown in A_new and B_new.

```
[maxi4] = fminsearch(@(x) sum(((predictive_distribution'-((x(2)./x(1))...
    .*((load_limit./x(1)).^(x(2)-1)).*exp(-((load_limit./x(1)).^x(2)))))...
    .^2)),[1780,70]);
u_new = maxi4(1,1);
k_new = maxi4(1,2);
[A_new,B_new] = wblstat(u_new,k_new);
```

Figure 1: Prior and predictive distribution

Figure 1 shows the difference between the prior and predictive distribution, where the predictive contains new test information.

```
figure
hold on;
plot(load_limit,predictive_distribution,'r','Linewidth',1.5)
plot(load_limit,prior_distribution,'b','Linewidth',1.5);
legend('Predictive distribution','Prior distribution');
xlabel('Load level');
ylabel('PDF');
hold off;
```

Figure 2: Prior, predictive and approximated predictive distribution

```
figure
hold on;
plot(load_limit,predictive_distribution,'r');
plot(load_limit,wblpdf(load_limit,u_new,k_new),'b');
plot(load_limit,wblpdf(load_limit,u_old_pred,k_old_pred),'g');
legend('Predictive distribution','approximated distribution','prior distribution');
xlabel('Load level')
ylabel('PDF')
hold off;
```

Input values - Reliability updating

The input values used in the reliability updating are presented here.

reliability_target = Target reliability
sensitivity = FORM sensitivity factor, conservatively = 1

```
reliability_target = 5;
sensitivity_factor = 1;
```

failed_wires = number of wires that fail the test
length_effect = 1, meaning that there is no reduction due to corrosion or damage in the wires
daniels_effect = 1, no reduction due to Daniels effect

```
failed_wires = 0;
length_effect = 1;
daniels_effect = 1;
```

euler_constant = Eulers constant
cov_load = Coefficient of variation for the load distribution
safety_load = partial safety factor load

```
euler_constant = 0.577215;
cov_load = 0.3;
safety_load = 1.5;
```

year_of_testing decides how many years from the structure is new to the tests are preformed, default this is 30 years.

```
step_rel = 0.01;
year_of_testing = 30;
```

Partial safety factor

The partial safety factor for the prior and posterior strength distribution, the load is distributed using a Beta-distribution and the strength using Weibull.

The characteristic values are given using the lower 5% fractile, with parameters taken from the prior predictive distribution (uncertainties from MLE are included).

```
characteristic_value_prior = wblinv(0.05,u_old_pred,k_old_pred);
characteristic_value_posterior = wblinv(0.05,u_new,k_new);
```

safety_res_prior and _posterior are the partial safety factors for the cable, with the general equation:

$$\gamma_x = \frac{x_k}{\xi_L \xi_D \mu_x (1 - \alpha_x \beta_t V_x)(1 - \rho_a ll}$$

Where:

ξ_l = Reduction factor due to length effect

ξ_D = Reduction factor due to Daniels effect

ρ_{all} = Takes into account hte number of broken wires

x_k = Characteristic wire strength

μ_x, α_x, β_t and V_x is the mean, FORM sensitivity factor, target reliability and coefficient of variation

```
safety_res_prior = characteristic_value_prior./(daniels_effect.*...
    length_effect.*A_old_pred.*(1-reliability_target....
    *sensitivity_factor.*(sqrt(B_old_pred)./A_old_pred))....
    *(1-failed_wires));

safety_res_posterior = characteristic_value_posterior./...
    (daniels_effect.*length_effect.*A_new.*(1-reliability_target....
    *sensitivity_factor.*(sqrt(B_new)./A_new)).*(1-failed_wires));
```

Prior and Posterior reliability

```
*Prior mean and standard deviation*
beta_prior = Scale parameter in Gumbell distribution, NOT related to
reliability.
```

$$\beta = \frac{F_R^{-1}(x_{resistance}}{\gamma_m \gamma_s \left(\frac{\pi}{cov_s \sqrt{6}} - \gamma - ln[-ln(x_{load})] \right)}$$

```
beta_prior = wblinv(0.05,u_old_pred,k_old_pred)/(safety_res_prior*...
    safety_load*(((pi)/(cov_load.*sqrt(6)))-euler_constant-log(-log(0.98))));
```

```
Mean and standard deviation of the prior distribution. With the
following equation:
```

$$\mu = \frac{F_R^{-1}(x_r esistance)}{\gamma_m \gamma_s} + \beta ln[-ln(x_{load})]$$

```
mu_prior = (wblinv(0.05,u_old_pred,k_old_pred)/...
    (safety_res_prior*safety_load))+beta_prior*log(-log(0.98));
sigma_prior = (pi/sqrt(6))*beta_prior;
mean_prior = mu_prior+euler_constant*beta_prior;
```

```
*Posterior mean and standard deviation*
beta_posterior = Scale parameter in Gumbell distribution, NOT related to
reliability
```

```
beta_posterior = wblinv(0.05,u_new,k_new)/(safety_res_posterior...
    *safety_load*(((pi)/(cov_load.*sqrt(6)))-euler_constant-log(-log(0.98))));
```

```
Mean and standard deviation of the prior distribution.
```

```
mu_posterior = (wblinv(0.05,u_new,k_new)/(safety_res_posterior...
    *safety_load))+beta_posterior*log(-log(0.98));
sigma_posterior = (pi/sqrt(6))*beta_posterior;
mean_posterior = mu_posterior+euler_constant*beta_posterior;
```

Prior reliability
The design equation (design_eq...) given below estimates when M = 0, i.e. when Resistance = Load.

fun_prior = Probability of failure given as a function of the variable x.
The reliability of the prior distribution is found in reliability_prior2

```
design_equation_difference_prior = (wblinv(0.05,u_old_pred,k_old_pred)...
    /(safety_res_prior*safety_load))-(mu_prior-beta_prior...
    *log(-log(0.98)));
fun_prior = @(x) (1-exp(-exp(-((x-mu_prior)./beta_prior)))...
    .*wblpdf(x,u_old_pred,k_old_pred);

P_fail_prior = integral(@(x) fun_prior(x),0,Inf);
reliability_prior2 = -norminv(P_fail_prior,0,1);
```

Posterior reliability
The design equation (design_eq...) given below estimates when M = 0, i.e. when Resistance = Load.

fun_prior = Probability of failure given as a function of the variable x.
The reliability of the posterior distribution is found in reliability_prior2

```
design_equation_difference_posterior = (wblinv(0.05,u_new,k_new)/...
    (safety_res_posterior*safety_load))-(mu_posterior-beta_posterior...
    *log(-log(0.98)));
fun_posterior = @(x) (1-exp(-exp(-((x-mu_posterior)./...
    beta_posterior)))).*wblpdf(x,u_new,k_new);

P_fail_posterior = integral(@(x) fun_posterior(x),0,Inf);
reliability_posterior2 = -norminv(P_fail_posterior,0,1);
```

Reliability

Using the time-invariant method, the reliability of the system is plotted. n_rel denotes number of years.

```
n_rel = 1:step_rel:100;
n2_rel = 1:step_rel:year_of_testing;
n3_rel = year_of_testing+step_rel:step_rel:100;
```

The probability of failure is given by the following equation:

$$P_f = (1 - (1 - P_{fail})^n)$$

```
probability_of_failur_old_over_time = (1-(1-P_fail_prior).^n_rel);
probability_of_failur_new_over_time2 = (1-(1-P_fail_prior).^n2_rel);
probability_of_failur_new_over_time3 = (1-(1-P_fail_posterior).^n3_rel);


probability_of_failur_new_over_time =
[probability_of_failur_new_over_time2';probability_of_failur_new_over_time3'];
reliability_over_time_prior = (-norminv(probability_of_failur_old_over_time));
reliability_over_time_posterior = (-norminv(probability_of_failur_new_over_time));
```

Figure 3: Time invariant reliability updating.

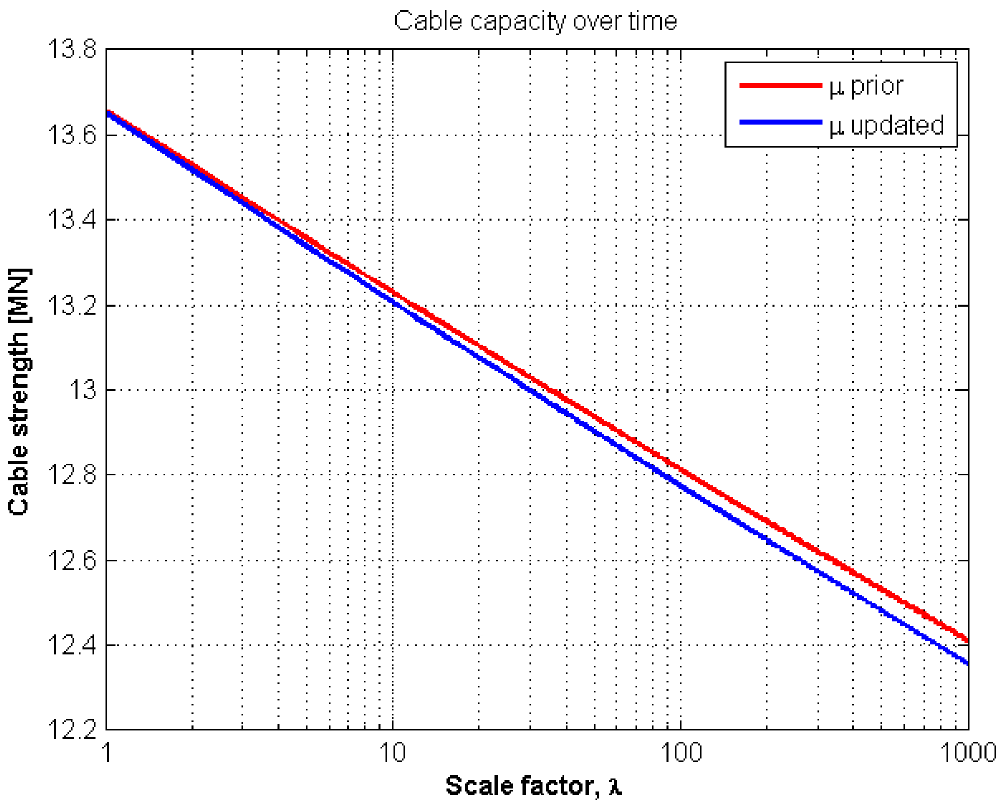
```
figure
hold on;
plot(n_rel,reliability_over_time_prior,'k','Linewidth',1.5);
plot(n_rel,reliability_over_time_posterior,'r','Linewidth',1.5)
xlabel('Lifetime in years');
ylabel('Reliability  \beta');
title('Time-invariant reliability distribution');
legend('Old reliability','updated reliability');
hold off;
```

Part 3 initial cable capacity

```
Lambda = Scale factor (NOT in relation to Daniels effect)
Wirestrength3 = Mean value of the strength
```

```
lambda = 1:1000;
Wirestrength_old = u_old_pred.*lambda.^(-1./k_old_pred).*...
    gamma(1+(1./k_old_pred));
Wirestrength_new = u_new.*lambda.^(-1./k_new).*gamma(1+(1./k_new));
cov3 = (sigma3./Wirestrength_old)*100;
fig1= figure(1);
set(fig1, 'visible', 'on')
title('Cable strength as a function of \lambda using Weibull',...
    'Color', 'k','Fontweight','bold','FontSize',11);
h=semilogx(lambda,(Wirestrength_old*n*Area)/1e6,'r',lambda,...
    (Wirestrength_new*n*Area)/1e6,'b');
New_XTickLabel = get(gca,'xtick');
set(gca,'XTickLabel',New_XTickLabel);
set(h(1),'linewidth',2); set(h(2),'linewidth',2);
title('Cable capacity over time');
xlabel('Scale factor, \lambda','Fontweight','bold');
```

```
ylabel('Cable strength [MN]','Fontweight','bold');
legend('\mu prior','\mu updated');
grid on;
```



Part 5 Calculation of strength of cable with Daniel's effect

For large bundles of wires, the strength of a cable is not seen as the sum of each wire, if the number of wires exceed 150 - 200, a reduction factor of the strength occurs. This effect is called the Daniels size effect.

w = Max range of x (Strength range)
j = Range for possible values for lambda
En5 = Mean strength value
Dn5 = Standard deviation of cable strength
The Weibull mean and variance are given by the following equations:

$$E(X) = nx_0[1 - F_Z(x_0)] + c_n$$

$$Var(X) = x_0[nF_Z(x_0)(1 - F_Z(x_0))]$$

```
w = 2000;
j = 1000;
```

```
En5 = zeros(j,1);
Dn5 = zeros(j,1);
```

Each iteration calculates the mean (En5) and standard deviation (Dn5), considering lambda as a variable.

x5 = x value range
Fz = Weibull cummulative distribution (CDF)
fz = Weibull density distribution (PDF)
dfz = Derivative of Weibull PDF

$$F_Z(z) = 1 - exp(-\lambda(z/b)^c)$$

$$f_Z(z) = \frac{\lambda * c}{b} * {\frac{x_i}{b}}^{c-1} exp[\lambda(\frac{x_i}{b})^{c-1}]$$

dummy = Equation 14 in faber, localizing x0
x05 = Calculates where the greatest x0 value for each lambda value
a5 = Correction term found in Faber et al.
cn5 = Correction term found in Faber et al.

$$x_0 = max[x(1 - F_Z(x_0))]$$

$$a^3 = \frac{f_z(x_0)^2 x_0^4}{2f_Z(x_0) + x_0 f_Z'(x_0)}$$

$$c_n = 0.966 n^{\frac{1}{3}} a$$

```
for lambda5 = 1:j;

x5 = 1:w;
Fz = (1-exp(-(lambda5.*(x5/u_old_pred).^k_old_pred)));
dummy = (x5.*(1-Fz));
dummy1 = max(dummy);
x05= x5(dummy == dummy1);

Fz5 = (1-exp(-(lambda5.*(x05/u_old_pred).^k_old_pred)));
fz5 = (((lambda5.*k_old_pred)/u_old_pred).*(x05/u_old_pred)....
    ^(k_old_pred-1).*exp(-lambda5.*(x05/u_old_pred).^k_old_pred));

dfz5 = ((lambda5.*k_old_pred.*exp(-lambda5.*(x05/u_old_pred)....
    ^k_old_pred))/k_old_pred.^2).*((k_old_pred-1).*(x05/u_old_pred)....
    ^(k_old_pred-2)-lambda5.*k_old_pred.*(x05/u_old_pred)....
    ^(2.*k_old_pred-2));

a5 = ((fz5.^2.*x05.^4)/(2.*fz5+x05.*dfz5));
cn5 = (0.966.*(n.^(1/3)).*a5.^(1/3));
Dn5(lambda5) = (x05.*sqrt(n.*Fz5.*(1-Fz5))).*(Area/1e6);
En5(lambda5) = (n.*x05.*(1-Fz5)+cn5).*(Area/1e6);
```

```
end

% In this plot, the strength is considered as a function of the aging
% factor, lamda.

figure
lambda = 1:1:1000;
z=semilogx(lambda,En5,'k');
axis([1 1000 11 13.5]);
New_XTickLabel = get(gca,'xtick');
set(gca,'XTickLabel',New_XTickLabel);
set(z(1),'linewidth',2)
xlabel('Scale factor, \lambda','Fontweight','bold');
ylabel('Cable capacity [MN]','Fontweight','bold');
legend('\mu');
title('Cable capacity with Daniels effect',...
    'Color', 'k','Fontweight','bold','FontSize',11);
grid on
```

Part 6 Reduction factor due to Daniels effekt as a function of the number wires

5) Daniels effect as a function of the number of wires

The calculations are preformed as earlier, see part 3) for more information.
While lambda was a variable in part 3), the number of wires is now considered as a variable. Otherwise, both parameters and variables are the same as in earlier calculations.

x6 = x value range
Fz = Weibull cummulative distribution (CDF)
fz = Weibull density distribution (PDF)
dfz = Derivative of Weibull PDF

dummy = Equation 14 in faber, localizing x0
x05 = Calculates where the greatest x0 value for each lambda value
a6 = Correction term found in Faber et al.
cn6 = Correction term found in Faber et al.

```
w = 2000;
lambda6 = 100;

x6 = 1:1:w;
Fz = (1-exp(-(lambda6.*(x6/u_old_pred).^k_old_pred)));
dummy = (x6.*(1-Fz));
dummy1 = max(dummy);
x06= x6(dummy == dummy1);

Fz6 = (1-exp(-(lambda6.*(x06/u_old_pred).^k_old_pred)));
fz6 = (((lambda6.*k_old_pred)/u_old_pred).*(x06/u_old_pred)....
    ^(k_old_pred-1).*exp(-lambda6.*(x06/u_old_pred).^k_old_pred));
dfz6 = ((lambda6.*k_old_pred.*exp(-lambda6.*(x06/u_old_pred)....
    ^k_old_pred))/u_old_pred.^2).*((k_old_pred-1).*(x06/u_old_pred)....
    ^(k_old_pred-2)-lambda6.*k_old_pred.*(x06/u_old_pred)....
    ^(2.*k_old_pred-2));
a6 = ((fz6.^2.*x06.^4)/(2.*fz6+x06.*dfz6));
```

```
f6 = vector ranging from 1 - l6 number of wires
N6 = Reduction of capacity due to Daniels effect
```

```
l6 = 10000;
f6 = 1:l6;
N6 = zeros(1,l6);
```

Calculating the reduction factor due to Daniels effect, where each iteration represent a number of wires, ranging from 1 - l6

```
for number = 1:l6;

En6 = (number.*x06.*(1-Fz6)...
    +(0.966.*(number.^(1/3)).*a6.^(1/3))).*(Area/1e6);
wirecapacity6 = ((lambda6.^-(1/k_old_pred))*u_old_pred).*number.*(Area/1e6);
reduction = En6/wirecapacity6;
N6(number) = reduction;

end

% Considering number of wires as the variable

figure
semilogx(f6,N6);
axis([10 l6 0.93 0.99]);
New_XTickLabel = get(gca,'xtick');
set(gca,'XTickLabel',New_XTickLabel);
xlabel('Number of wires','Fontweight','bold');
ylabel('Reduction factor due to Daniels effect','Fontweight','bold');
legend('Cable strength reduction factor');
title('Reduction factor due to Daniels effect (\lambda=constant)'...
    ,'Color', 'k','Fontweight','bold','FontSize',11);
grid on
```

Part 7 Stress strain

Stress-strain related input values

This chapter contains input values regarding the stress-strain simulation with the following input values:

mu_u7 = Mean yield strain
sigma_y7 = Standard deviation yield strain
mu_u7 = Mean failure strain (Ductile)
sigma_u7 = Standard deviation failure strain (Ductile)
mu_u72 = Brittle failure strain
sigma_u72 = Brittle standard deviation

```
mu_y7 = 2;
sigma_y7 = 0.2;
mu_u7 = 2.5;
sigma_u7 = 1.0;
mu_u72 = 1.0;
sigma_u72 = 0.5;
```

ro = Correlation factor between
max_epsilon = Strain range used in the calculation and plotting of the cable strength.
a7 = Iteration step used in the loops (see "Inner loop")
N7 = Number of simulations (see "outer loop")

```
ro = -0.45;
max_epsilon = 4;
a7 = 0.01;
N7 = 500;
```

a_7 = Lower bound for initial strain
b_7 = Upper bound for initial strain

```
a_7 = 0;
b_7 = 1;
```

Log-normally distributed strength

Both the yield and failure strength is assumed log-normally distributed with the following parameters:

$$\text{Log normal mean}(\mu) = ln(\frac{\mu^2}{\sqrt{\sigma^2 + \mu^2}})$$

$$\text{Log normal mean}(\mu) = ln(\frac{\mu^2}{\sqrt{\sigma^2 + \mu^2}})$$

my7 = Yield mean
sy7 = Yield standard deviation
mu7 = Ductile failure mean
su7 = Ductile failure SD

mu72 = Brittle failure mean
su72 = Brittle failure SD

```
my7 = log(((mu_y7)^2)/sqrt(sigma_y7^2 + (mu_y7^2)));
sy7 = sqrt(log((sigma_y7^2/mu_y7^2)+1));

mu7 = log(mu_u7^2/(sqrt(sigma_u7^2 + mu_u7^2)));
su7 = sqrt(log((sigma_u7^2/mu_u7^2)+1));

mu72 = log(mu_u72^2/(sqrt(sigma_u72^2 + mu_u72^2)));
su72 = sqrt(log((sigma_u72^2/mu_u72^2)+1));
```

j7 = Range of strain calculations o7 = Number of calculations, see "inner loop" further down

```
j7 = 0:a7:max_epsilon;
o7 = length(j7);
```

Empty vectors collecting loop values

1) The following vectors collect the strength and strain when the cable behave ductile

jobb101 = Cable strengths for each strain value after each simulation
jobb102 = Collecting cable strength for each strain value
jobb103 = Collecting maximum cable strength in each simulation
indexe1 = Collecting the strain at the point with the highest capacity

```
jobb101 = zeros(o7,N7);
jobb102 = zeros(o7,1);
jobb103 = zeros(N7,1);
indexe1 = zeros(N7,1);
```

2) The following vectors collect the strength and strain when the cable behave brittle
jobb104 = Cable strengths for each strain value after each simulation
jobb105 = Collecting cable strength for each strain value
jobb106 = Collecting maximum cable strength in each simulation
indexe2 = Collecting the strain at the point with the highest capacity

```
jobb104 = zeros(o7,N7);
jobb105 = zeros(o7,1);
jobb106 = zeros(N7,1);
indexe2 = zeros(N7,1);
```

An empty vector for collecting average initial strain for each simulation

```
average_initial_epsilon = zeros(N7,1);
```

Total number of simulations (Outer for-loop)

The outer loop contains initial assumptions and caluclations regarding the initial stress range, correlation factors and values for yield strain.

As seen from the simulation range, the outer loop decides the number of simulations preformed in the Monte Carlo method.

```
for simulation = 1:N7;
```

R7 = Random number generator (default: Uniform distributed), with range a_7-b_7.
Initial_epsilon = R7 = Used to normalize the vector length 'epsilon_value'
average_initial_epsilon(simulation) = Average initial strain

```
    R7 = (b_7-a_7)*rand(n,1)+a_7;
    initial_epsilon = R7;
    average_initial_epsilon(simulation) = ((sum(R7))/n);
```

While the initial strain is uniformly distributed, the strength of the wire is lognormally distributed with mean and variance as given above.
The simulated strength are calculated using the following steps:

c71 = Latin hybercube sample from a normal distribution
c72 = Latin hybercube sample from a normal distribution
X71 = Correlated variable pair [x71 X71]
rn71 = Normally distributed random numbers
rn72 = Normally distributed random numbers, with a correlation = ro with rn71

```
    c71 = lhsnorm(0,1,n);
    c72 = lhsnorm(0,1,n);
    X71 = c71*ro+c72*sqrt(1-ro^2);
    rn71 = normcdf(c71,0,1);
    rn72 = normcdf(X71,0,1);
```

In order to simulate the yield and failure strain of the cable, the normally random numbers calculated above are used. These simulate the strain capacity of the wires. This capacity can be shown to be lognormally distributed for a large number of wires.
In these calculations, the function logninv is used:

$$x = F^{-1}(p|\mu,\sigma) = x : F(x|\mu,\sigma) = p$$

$$p = F(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^x \frac{e^{\frac{-(\ln(t)-\mu)^2}{2\sigma^2}}}{t} dt$$

By invoking the inverse lognormal function above, the yield and failure strain is found. Both brittle and ductile cases are considered.

$$\text{Plastic elongation} = \varepsilon_{u,i} - \varepsilon_{y,i}$$

```
    epsilon_yield = logninv(rn71,my7,sy7);
    epsilon_elongation_ductile = logninv(rn72,mu7,su7);
    epsilon_failure_ductile = epsilon_elongation_ductile + epsilon_yield;

    epsilon_elongation_brittle = logninv(rn72,mu72,su72);
    epsilon_failure_brittle = epsilon_elongation_brittle + epsilon_yield;
```

Q7 decides the length of the vectors model100 and model101, these are vectors containing the stress values in each wire.
model100 = Ductile behavior
model101 = Brittle behavior

```
    Q7 = numel(c71);
    model100 = zeros(Q7,1);
    model101 = zeros(Q7,1);
```

Loop for each strain value (inner loop)

The inner loop simulates the strain in each wire, where the number of simulations is decideb by the vector o7, i.e. with the strain range is 0 - o7 (default = 4%).

```
for jj = 1:o7;
```

Q7 denotes the number of wires in the cable (Q7), where the strain values are calculated c.f. Faber et al. [1]

```
    for ii = 1:Q7;
```

epsilon_value = Assignes each wire with an initial strain an make sure that the vector for each wire is the same length as length in each simulation.
Shortly explained: The strains are simulated randomly, meaning that the length of the vector changes for each simulation, since the values change after each simulation. In order to sample these values, MATLAB demands that the length of the vectors are the same.

```
    epsilon_value = initial_epsilon(ii):a7:max_epsilon+initial_epsilon(ii);
```

The if sentences given below calculates the strain according to the equation 17 in Faber et al [1]. Shortly summarized the strain in a wire are given as three possibilites (Too short time to implement in Latex):
1) Strain < than yield strain = The strain in the wire
2) Strain between yield an failure strain = Strain in the wire are given as the yield strain
3) Strain in the wire > failure strain = The wire is
considered broken.

The upper part calculates the strain if the wires behave ductile and the lower if the wires behave brittle.

```
    if epsilon_yield(ii) >= epsilon_value(jj);
        epsilon1 = epsilon_value(jj);
    elseif epsilon_yield(ii) < epsilon_value(jj)...
           && epsilon_value(jj) < epsilon_failure_ductile(ii);
        epsilon1 = epsilon_yield(ii);
    elseif epsilon_failure_ductile(ii) <= epsilon_value(jj);
        epsilon1 = 0;
    end

    if epsilon_yield(ii) >= epsilon_value(jj);
        epsilon2 = epsilon_value(jj);
    elseif epsilon_yield(ii) < epsilon_value(jj)...
           && epsilon_value(jj) < epsilon_failure_brittle(ii);
        epsilon2 = epsilon_yield(ii);
    elseif epsilon_failure_brittle(ii) <= epsilon_value(jj);
        epsilon2 = 0;
    end
```

model100(ii) = Sampling stress for each wire (Ductile)
model101(ii) = Sampling stress for each wire (Brittle)
The strain values are given in %, dividing by 100 to compensate

```
    model100(ii) = ((epsilon1/100)*E);
    model101(ii) = ((epsilon2/100)*E);

    end
```

p7 = The capacity of the cable given as a sum of each wire (Ductile)
p72 = The capacity of the cable given as a sum of each wire (brittle)

```
    p7 = sum(model100)*Area;
    p72 = sum(model101)*Area;
```

The calculations below are identical, with the only difference that one estimates the maximum capacity (jobb103) in the ductile case and the other for brittle behavior (jobb106)

jobb101(jj,simulation) = Capacity of the cable after each strain simulation (outer loop), see Figure 1
jobb102(jj) = Capacity of the cable (inner loop)
[MaxValue,linearIndexOfMaxes] = Locates the maximum capacity in each simulation
max_capacity1 = The maximum capacity
indexe1(simulation) = Location of the maximum capacity
jobb103(simulation) = Maximum capacity value for each simulation

```
    jobb101(jj,simulation) = p7;
    jobb102(jj) = p7;
    [MaxValue,linearIndexOfMaxes] = max(jobb102(:));
    pst1 = [MaxValue,linearIndexOfMaxes];
    max_capacity1 = pst1(1,1);
    indexe1(simulation) = pst1(1,2);
    jobb103(simulation) = max_capacity1;

    jobb104(jj,simulation) = p72;
    jobb105(jj) = p72;
    [MaxValue,linearIndexOfMaxes] = max(jobb105(:));
    pst2 = [MaxValue,linearIndexOfMaxes];
    max_capacity2 = pst2(1,1);
    indexe2(simulation) = pst2(1,2);
    jobb106(simulation) = max_capacity2;

end
```

meanjobb1(simulation) = Average cable strength after each simulation (Ductile)
meanjobb2(simulation) = Average cable strength after each simulation (Brittle)

```
meanjobb1(simulation) = mean(jobb103);
meanjobb2(simulation) = mean(jobb106);

end
```

Final calculation

This chapter contains the final calculations, where the strength of a cable are estimated, based on the simulations above.

q7 = Vector with values going from 1 to total number of simulations, mainly used in the plots and as a "normalizing" factor

```
q7 = 1:N7;
```

Ductile behavior

The strength of a cable when the behavior is ductile

A7(72) finds the mean value for each "external" epsilon step by summing the values in each row, and dividing by total number of simulations.

M7 = A matrix containing the sum of each step
mean7 = Average mean value for each simulation
A7 = Mean value of all the simulations for each value of epsilon
epsilon_external_at_max_cap1 = The strain values at the maximum capacity after each simulation
epsilon_at_max_cap1 = Total strain value at maximum capacity

The mean function is multiplied by the total number of simulations (N7), to compensate for this, the vector q7 = [1 2 ... N7] are multiplied so that the number of simulations are taken into account.

epsilon_external_at_max_cap finds at what external strain the maximum capacity is after each simulations. These maximum values are used in the histogram plots (Figure 6 and 7)

```
M7 = (meanjobb1'.*N7);
mean7 = M7./q7';
A7 = (sum(jobb101,2))/N7;
epsilon_external_at_max_cap1 = j7(indexe1)';
epsilon_at_max_cap1 = epsilon_external_at_max_cap1...
                      + average_initial_epsilon;
```

meanvalue71 = Mean value
standardvalue71 = Standard deviation
cov71 = Coefficient of variation

```
meanvalue71 = mean(jobb103)/1e6;
standardvalue71 = std(jobb103)/1e6;
cov71 = (standardvalue71/meanvalue71)*100;
cap71 = [meanvalue71 standardvalue71 cov71];
```

Brittle behavior

The calculations are exactly the same as the one given above

```
M72 = (meanjobb2'.*N7);
mean72 = M72./q7';
```

```
A72 = (sum(jobb104,2))/N7;
epsilon_external_at_max_cap2 = j7(indexe2)';
epsilon_at_max_cap2 = epsilon_external_at_max_cap2...
                      + average_initial_epsilon;

meanvalue72 = mean(jobb106)/1e6;
standardvalue72 = std(jobb106)/1e6;
cov72 = (standardvalue72/meanvalue72)*100;
cap72 = [meanvalue72 standardvalue72 cov72];
```

Maximum mean value, standard deviation and coefficient of variation of ductile and britle wires

```
cap7 = [cap71;cap72];
```

Figure 1: Individual cable capacity for each simulation (Ductile)

The capacity for each simulation are shown here, as we see, the capacities are relatively similiar in the elastic range, where the main difference between the simulations occur in the plastic range

```
figure
hold on;
plot(j7,jobb101);
ylabel('Force [N]','Fontweight','bold');
xlabel('Strain external, \epsilon (%)','Fontweight','bold');
title('Individual cable, ductile behaviour','Color', 'k',...
      'Fontweight','bold','FontSize',11);
grid on;
hold off;
```

Figure 2: Individual cable capacity for each simulation (Ductile)

The capacity for each simulation are shown here, as we see, the capacities are relatively similiar in the elastic range, where the main difference between the simulations occur in the plastic range.
The main difference from Figure 1 is the plastic plateau, which is smaller when we assume that the cable behaves brittle.

```
figure
hold on;
plot(j7,jobb104);
ylabel('Force [N]','Fontweight','bold');
xlabel('Strain external, \epsilon (%)','Fontweight','bold');
title('Individual cable, brittle behaviour',...
       'Color', 'k','Fontweight','bold','FontSize',11);
```

```
grid on;
hold off;
```

Figure 3: Convergence rate based on the number of simulations (Ductile)

The capacity converge towards the exact solution roughly after 300 - 500 simulations

```
figure
hold on;
plot(q7,mean7);
xlabel('Number of simulations, N7','Fontweight','bold');
ylabel('Average force [N]','Fontweight','bold');
title('Convergance of number of simulations','Color', 'k',...
      'Fontweight','bold','FontSize',11);
legend('ductile');
hold off;
```

Figure 4: Convergence rate based on the number of simulations (Brittle)

The capacity converge towards the exact solution roughly after 250 - 300 simulations

```
figure
hold on;
plot(q7,mean72);
xlabel('Number of simulations, N7','Fontweight','bold');
ylabel('Average force [N]','Fontweight','bold');
title('Convergance of number of simulations','Color', 'k',...
      'Fontweight','bold','FontSize',11);
legend('brittle');
hold off;
```

Figure 5: Brittle and Ductile cable capacity as an average

The capacities shown here are an average of the capacities found in Figure 1 and 2

```
figure
hold on;
h7 = plot(j7,A7,'k',j7,A72,'b');
set(h7(1),'linewidth',2,'linestyle','-.')
xlabel('External strain, \epsilon (%)','Fontweight','bold');
ylabel('Force [N]','Fontweight','bold');
legend('Ductile behaviour','brittle behaviour')
title('Stress-strain relationship','Color', 'k',...
      'Fontweight','bold','FontSize',11);
```

```
grid on
hold off;
```

Figure 6: Histogram showing the maximum capacity at different strains (Ductile)

As seen in the figure, most of the simulations show a maximum capacity around an 2.7% elongation

```
figure
hold on;
hist(epsilon_at_max_cap1);
xlabel('Epsilon');
ylabel('Number of simulations');
title('For what epsilon value that give the highest capacity, ductile');
grid on;
hold off;
```

Figure 7: Histogram showing the maximum capacity at different strains (Brittle)

As seen in the figure, most of the simulations show a maximum capacity around an 2.15% elongation

```
figure
hold on;
hist(epsilon_at_max_cap2);
xlabel('Epsilon');
ylabel('Number of simulations');
title('For what epsilon value that give the highest capacity, Brittle');
grid on;
hold off;

toc;
```

APPENDIX E – MATLAB SCRIPT FOR ANALYTICAL UPDATING

Uncertain mean and known standard deviation

Normal Distribution with known Standard deviation and unknown mean

```
clc; clear all;
```

x_hat = New information (observations)
X = stress range

```
tic;
step = 0.01;
X = 1760:step:1800;
x_hat = log([1750 1760 1770 1780 1790]);
```

Initial Assumptions

A normally distributed variable X, whith parameters Theta = $(Mx,sigma_x)$ is considered. The standard deviation sigma_x is known, and the mean value (M_x) is considered a random variable.

The prior distribution has an uncertain mean and the following parameters:

$$\theta = (M_x, \sigma_X)^T$$

Where the mean M_x has the following properties:

$$M_X \ N(\mu^{'}, \sigma^{'})$$

mu_prior = mean of Mx
sigma_x = Standard deviation of theta

```
sigma_x = 0.5;
mu_prior = 7.5;
```

New tests (observations) are made with the following mean and number of new test samples

```
n_new = numel(x_hat);
mu_new = mean(x_hat);
```

The loop below simply allows the plotting of three different graphs, where P is the number of tests (n_prior), i.e. the loop plots the PDF when we have 1 test, 3 tests and 5 tests.

jobb_mean = Sampling vector for the mean
jobb_std = Sampling vector for standard deviation

```
P = 1:2:5;
k = numel(P);
jobb_mean = zeros(k,1);
jobb_std = zeros(k,1);

for ii = 1:k;
```

n_prior = number of prior tests
sigma_prior_dot = standard deviation of Mx
sigma_prior = Standard deviation of the prior distribution

$$\sigma'^2 = \frac{\sigma_X^2}{n'}$$

$$\sigma' = \sqrt{\sigma_X^2 + \sigma'^2}$$

```
n_prior = P(ii);
sigma_prior_dot = sqrt(sigma_x.^2./n_prior);
sigma_prior = sqrt(sigma_x.^2+sigma_prior_dot.^2);
```

mu_posterior = Mean of posterior distribution
sigma_posterior = Standard deviation of posterior distribution
sigma_predictive = Standard deviation of predictive distribution

$$\mathrm{mu.posterior} = \mu'' = \frac{\frac{\mu'}{n} + \frac{\bar{x}}{n'}}{\frac{1}{n} + \frac{1}{n'}}$$

$$\mathrm{sigma.posterior} = \sigma'' = \sqrt{\frac{\frac{\sigma_X}{n} * \frac{\sigma'}{n'}}{\frac{\sigma'^2}{n} + \frac{\sigma'^2}{n'}}}$$

$$\text{sigma.predictive} = \sigma''' = \sqrt{\sigma''^2 + \sigma_X^2}$$

```
mu_posterior = (((mu_prior/n_new)+(mu_new/n_prior))/((1/n_new)+(1/n_prior)));
sigma_posterior = sqrt(((sigma_x*sigma_prior)/(n_new*n_prior))...
   /((sigma_x^2/n_new)+(sigma_x^2/n_prior)));
sigma_predictive = sqrt(sigma_posterior^2+sigma_x^2);
```

Trasnferring the mean and standard deviation from lognormal to normal distribution

```
mu = exp(mu_posterior);
sigma = exp(sigma_predictive);
mu2 = exp(mu_prior);
sigma2 = exp(sigma_prior);
```

The normally distributed PDF for the prior and predictive distribution.

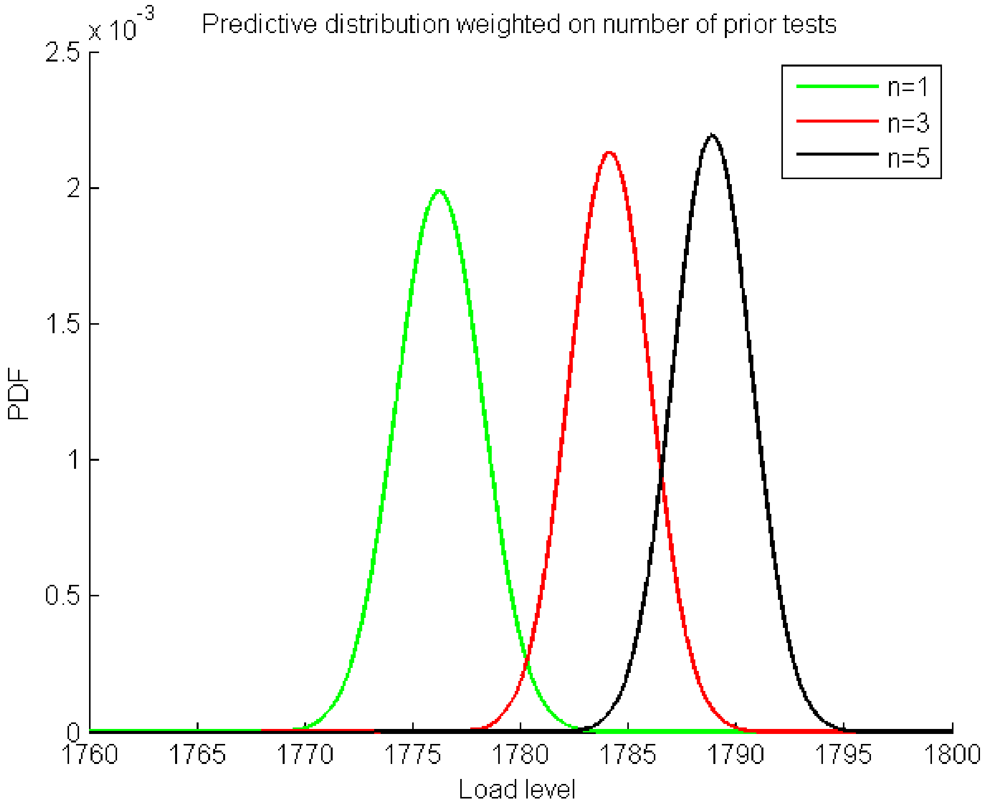
```
predictive_distribution_pdf = normpdf(X,mu,sigma);
prior_distribution_pdf = normpdf(X,mu2,sigma2);
jobb_mean(ii) = mu;
jobb_std(ii) = sigma;
end
```

Figure 1: Predictive distribution

How different number of tests affect the predictive distribution are shown below.

```
figure
hold on;
plot(X,normpdf(X,jobb_mean(1,1),jobb_std(1,1)).*step,'g','Linewidth',1.5);
plot(X,normpdf(X,jobb_mean(2,1),jobb_std(2,1)).*step,'r','Linewidth',1.5);
plot(X,normpdf(X,jobb_mean(3,1),jobb_std(3,1)).*step,'k','Linewidth',1.5)
legend('n=1','n=3','n=5');
title('Predictive distribution weighted on number of prior tests');
xlabel('Load level');
ylabel('PDF');
hold off;
toc;
```

```
Elapsed time is 1.239221 seconds.
```



Published with MATLAB® R2013a

Uncertain mean and standard deviation

Normal Distribution with unknown Standard deviation and unknown mean

```
The following example is based of an example in Annex A in the
dissertation written by our supervisor Jochen Koehler.
```

```
clc; clear all;
```

Stress range and number of test samples

X = stress range
fa = number of new tests, used to demonstrate the effect in the predictive distribution.
l = number of new tests

```
step = 0.01;
X = 1760:step:1810;
b = numel(X);
fa = 1:2:5;
l = numel(fa);
```

Prior information:

mu_prior = Prior mean
sigma_prior = Prior standard deviation
dn_prior =
tom = Samples the predictive distribution for different number of new tests (fa), see 'inner loop'

```
mu_prior = 7.5;
sigma_prior = 1.5;
dn_prior = 1;
tom = zeros(b,fa);
```

```
Warning: Size inputs must be scalar. This will error in a future release.
```

New information

z = new test samples
mu_new and mu_new2 is the log-mean and mean respectively
sigma_new and sigma_new2 is the log-SD and standard deviation

```
z = log([1750 1760 1770 1780 1790]);
mu_new = mean(z);
mu_new2 = exp(mu_new);
sigma_new = std(z);
sigma_new2 = exp(sigma_new);
```

n_new = Number of new test samples
dn_new = ??
v_new = Degrees of freedom (DOF) in the new test sample (students t-distribution)

```
n_new = numel(z);
dn_new = 1;
v_new = n_new-1;
```

Outer loop

The outer loop runs 3 iterations (default)

```
for ff = 1:1;
```

mu_prior2 and sigma_prior2 transforms the prior standard deviation and mean from a lognormal distribution to normal.

```
mu_prior2 = exp(mu_prior);
sigma_prior2 = exp(sigma_prior);
```

n_prior = Number of new tests, where the the number in the first (default) are 1, then 3 and finally 5.
v_prior = number of DOF

```
n_prior = fa(ff);
v_prior = n_prior+1;
```

Posterior mean, standard deviation and DOF

$$\text{mu.posterior} = \frac{n^{'}m^{'} + nm}{n^{'} + n}$$

$$\text{n.posterior} = n'' + n$$

$$\text{sigma.posterior} = \sqrt{\frac{(\nu' s'^2 + n'm'^2) + (\nu s^2 + nm^2) - n''m''^2}{(\nu^{'} + \delta(n^{'})) + (\nu + \delta(n)) - \delta(n''}}$$

$$\text{v.posterior} = (\nu^{'} + \delta(n^{'})) + (\nu + \delta(n)) - \delta(n^{''})$$

```
mu_posterior = ((n_prior*mu_prior+n_new*mu_new)/(n_prior+n_new));
mu_posterior2 = exp(mu_posterior);
n_posterior = n_new+n_prior;
dn_posterior = 1;
sigma_posterior = sqrt(((v_prior*sigma_prior^2+n_prior*mu_prior^2)+...
    (v_new*sigma_new^2+n_new*mu_new^2)-n_posterior*mu_posterior^2)/...
    ((v_prior+dn_prior)+(v_new+dn_new)-dn_posterior));
```

```
sigma_posterior2 = exp(sigma_posterior);
v_posterior = v_prior+dn_prior+v_new+dn_new-dn_posterior;

fungamma_prior = @(v) exp(-v).*v.^(v_prior./2);
gamma_prior = integral(@(v)fungamma_prior(v),0,Inf);
fungamma_posterior = @(v) exp(-v).*v.^(v_posterior./2);
gamma_posterior = integral(@(v)fungamma_posterior(v),0,Inf);
```

Prior and posterior distribution

The prior and posterior distribution are given as the Normal-Inverse-Gamma-2 given below

$$\text{prior} = f_{M,\Sigma}(\mu, h|m^{'}, s^{'}, n^{'}, v^{'}) = f_M(\mu|m^{'}, hn^{'})f_{\Sigma}(h|s^{'}, v^{'})$$

$$\text{prior} = f_{M,\Sigma}(\mu, h|m^{'}, s^{'}, n^{'}, v^{'}) = f_M(\mu|m^{'}, hn^{'})f_{\Sigma}(h|s^{'}, v^{'})$$

$$f_M(\mu|m^{'}, hn^{'}) = \frac{\sqrt{hn^{'}}}{\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{\mu - m^{'}}{1/\sqrt{hn^{'}}}\right)^2\right)\frac{1}{2\Gamma(\nu^{'}/2)}\left(\frac{1}{2}\nu^{'}s^{'2}h\right)^{\frac{1}{2}\nu^{'}-1}$$

$$f_{\Sigma}(h|s^{'}, v^{'}) = exp\left(-\frac{1}{2}v^{'}s^{'2}h\right)v^{'}s^{'2}$$

Where the prior is based on the prior information and posterior the updated. Both follow the Normal-Inverse-Gamma-2 distribution given above.

```
prior =@(mu,sigma) (sqrt((1./sigma.^2).*n_prior)./sqrt(2.*pi)).*...
    exp(-0.5.*(((mu-mu_prior2)./(1./sqrt((1./sigma.^2).*...
    n_prior))).^2)).*(1./(gamma_prior.*2)).*(0.5.*v_prior.*...
    sigma_prior2.^2.*(1./sigma.^2)).^(0.5.*v_prior-1).*exp(-0.5.*...
    v_prior.*sigma_prior2.^2.*(1./sigma.^2)).*v_prior.*sigma_prior2.^2;
posterior = @(mu,sigma)(sqrt((1./sigma.^2).*n_posterior)./sqrt(2.*pi)).*...
    exp(-0.5.*(((mu-mu_posterior2)./(1./sqrt((1./sigma.^2).*...
    n_posterior))).^2)).*(1./(gamma_posterior.*2)).*(0.5.*v_posterior.*...
    sigma_posterior2.^2.*(1./sigma.^2)).^(0.5.*v_posterior-1).*...
    exp(-0.5.*v_posterior.*sigma_posterior2.^2.*(1./sigma.^2)).*...
    v_posterior.*sigma_posterior2.^2;
```

Predictive distribution

The predictive distribution is found by integrating the mean and standard deviation.

fun_pred = normalPDF multiplied with the posterior distribution yields the predictive distribution as a function of mean and standard

deviation.
predictive2 = Integrating the unknowns yields the predictive function, where the variable 'x' is the wirestrength.
predictive = Arrayfun inserts different values for 'x' directly.

```
normalpdf = @(x,mu,sigma) normpdf(x,mu,sigma);
fun_pred = @(x,mu,sigma) normalpdf(x,mu,sigma).*posterior(mu,sigma);

predictive2 =@(x) integral2(@(mu,sigma)fun_pred(x,mu,sigma),0,2000,0,1000);
predictive = @(x) arrayfun(predictive2,x);

for ss = 1:b;
    sa = predictive(X(ss))*step;
    tom (ss,ff) = sa;
end

end
```

Normalization

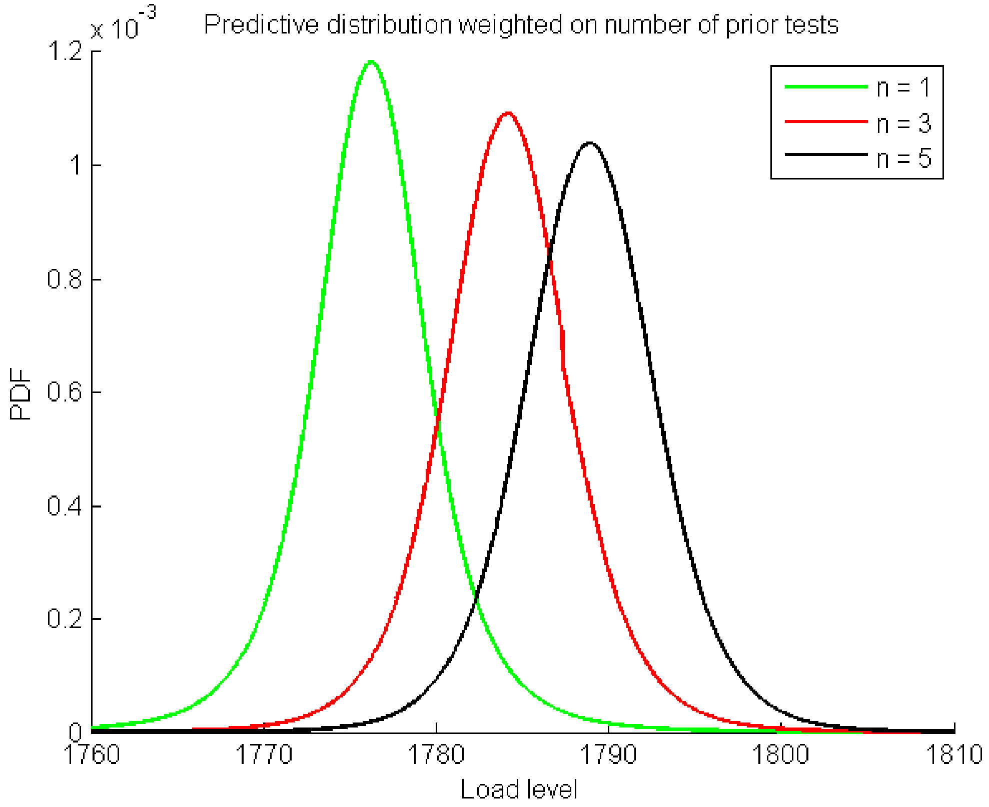
a1 - a3 calculates the normalization factor so that the area of the predictive distribution = 1.

```
a1 = 1/sum(tom(:,1));
a2 = 1/sum(tom(:,2));
a3 = 1/sum(tom(:,3));
```

Figure 1: Predictive distribution

Figure 1 shows the effects of different prior test numbers.

```
figure
hold on;
plot(X,tom(:,1)*a1,'g','Linewidth',1.5);
plot(X,tom(:,2)*a2,'r','Linewidth',1.5);
plot(X,tom(:,3)*a3,'k','Linewidth',1.5);
legend('n = 1','n = 3','n = 5');
title('Predictive distribution weighted on number of prior tests');
xlabel('Load level');
ylabel('PDF');
hold off;
```



Published with MATLAB® R2013a

APPENDIX F – MATLAB SCRIPT FOR WEIBULL PDF

```
function newpdf=wblpdf(x,A,B)
% The weibullpdf based on the shape and scale parameteres A and
B. Where
% 'x' is
newpdf = B/A.*(x/A).^(B-1).*exp(-(x/A).^B);
```