Spring 2012

Development and Validation of Probabilistic Fatigue Models Containing Out-life Suspensions

Noel S. Murray

DEVELOPMENT AND VALIDATION OF PROBABILISTIC FATIGUE MODELS CONTAINING OUT-LIFE SUSPENSIONS

by

NOEL MURRAY

(Under the Direction of Brian L Vlcek)

ABSTRACT

In the area of reliability engineering it is necessary to be confident that a component or system of components will not fail under use for safety and cost reasons. One major mechanism of failure to a mechanical component is fatigue. This is the repetitious motion of loading and unloading of the material, typically below the ultimate tensile strength of the material, which ultimately leads to a catastrophic failure. To ensure this does not happen, engineers design components based on tests to determine the life of these components. These tests are typically conducted on a bench type tester in which a sample it subjected to tension and compression, or supported in a rotational machine in which a load is applied to one end to simulate constant bending. The results from these tests tell how long it is predicted that the part will last.

This data however is not always complete. It sometimes happens that not every specimen tested actually makes it to failure; the un-failed specimens are known as suspensions. This can occur for numerous reasons. Methods currently exist for handling suspensions; however these methods require tedious hand calculations and interpolations from multiple graphs which are limited in availability.

Presented here are five methods utilizing the Monte Carlo technique in a computer simulation based on Weibull-Johnson confidence numbers that take into account suspensions. This simulation allows for data from an existing experiment to be used as inputs and either validate the findings or bring attention for more testing. The model allows for two different data sets containing suspensions to be analyzed and determine with statistical confidence whether or not there is a difference between the two populations.

INDEX WORDS: Fatigue, Monte Carlo, Weibull, confidence number, suspensions

DEVELOPMENT AND VALIDATION OF PROBABILISTIC FATIGUE MODELS CONTAINING OUT-LIFE SUSPENSIONS

by

NOEL MURRAY

B.S., Georgia Southern University, 2008

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial

Fulfillment

of the Requirements for the Degree

MASTER OF SCIENCE IN APPLIED ENGINEERING

STATESBORO, GEORGIA

2012

DEVELOPMENT AND VALIDATION OF PROBABILISTIC FATIGUE MODELS CONTAINING OUT-LIFE SUSPENSIONS

by

NOEL MURRAY

Major Professor: Brain L. Vlcek, Ph.D.

Committee: David Williams, Ph.D.
Anoop Desai, Ph.D.

Electronic Version Approved:
May 2012

DEDICATION

I would like to dedicate this to my parents, John and Debra Murray, for sticking with me through the entire process and for encouraging me to pursue a graduate degree.

I would also like to dedicate it to my advisor and mentor Dr. Brian Vlcek for helping me through all the tough spots of getting a graduate degree and for being patient with me throughout the whole process.

ACKNOWLEDGMENTS

I would like to acknowledge the department of Mechanical and Electrical Engineering Technology for giving me the opportunity to pursue this degree as well as giving me the resources that I needed.

I would also like to acknowledge the people responsible for helping me along the way in my research:

Brian L. Vlcek, PhD
Professor and Program Coordinator of Mechanical Engineering
Georgia Southern University

Robert Hendricks
Senior Technologist
NASA Glenn Research Center

Erwin Zaretsky, PE
Senior Technologist
NASA Glenn Research Center

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Background

An engineering design as defined by Norton is "the process of applying the various techniques and scientific principles for the purpose of defining a device, a process, or a system in sufficient detail to permit its realization" (Norton 2006). It is a mechanical engineer's responsibility to design safe and reliable machines for society. In engineering school students are taught deterministic equations to determine at what point a particular component of a given material will fail. In the real world, however, this is not the case. Real machines are subject to environments that cannot be incorporated into a mathematical model. The life of these machines therefore is probabilistic and not deterministic. This means there is a range of data with inherent scatter as opposed to determining a single absolute value. Engineers must rely on statistical life equations to estimate the life of the product being designed (Zaretsky, Design for life, plan for death 1994).

In designing a component or a system, it is necessary to consider how long the system will last, as well as the safety of the people that will be involved with the components. These considerations include warranty information, (need to know how long it will last to keep customers satisfied with their purchase, and not to have it too long to avoid repeat repairs); preventive maintenance schedules, (to keep the dealer and technicians informed when they need to replace or check on specific components); safety of machinery operators, or passengers in some form of transportation.

After an idea for a new mechanical system has been established, it goes through a design process that includes the preliminary design stage, the detailed design stage, and finally the documentation stage. While a system is in the detailed design stage a series of mathematical models may be made to analyze the system, such as how strong a particular component or material is, or how long that component may last. Then, experimental analysis may be conducted to back up or verify the mathematical models (Norton 2006).

Fatigue

When a material or component is loaded and unloaded hundreds or thousands of times below its ultimate yield stress, small cracks may begin to develop and accumulate. As these cracks grow and form a small network, a spall, or chunk of material breaks out leading to ultimate failure of the component. This is known as fatigue failure. In designing load-bearing components, the possibility of fatigue must be accounted for (Askeland and Phule 2006). “Fatigue failure is responsible for the majority of failures in mechanical components” (Kalpakjian and Schmid 2006).

Some examples of cyclic loads are repetitive contact of gear teeth, hot and cold heat cycles, pressurizing and depressurizing of pressure vessels, rotating shafts with an eccentric load, a spring repeatedly compressing and expanding, or repetitive loading and unloading of a beam. Most failures in mechanical systems are due to cyclic loads rather than to static loads (Norton 2006). This is the reason components cannot be designed solely on their static limits.

Fatigue Testing

One of the tests performed to determine the proficiency of mechanical components is the fatigue test. In many applications, components must be designed such that the load on the material is not great enough to cause permanent deformation. Fatigue testing involves taking samples of the same material and same shape and exposing them to cyclic loads until failure at controlled conditions—load frequency, test sample temperature, maximum stress or strain, etc. These loads can be rotational, where the samples are placed horizontally in a machine, with a force applied perpendicular to one end and then rotated rapidly, thereby causing a constant bending cycle in the middle of the sample. They can also be linear in a push-pull type tester where the sample is subjected to tension and compression repeatedly. Most of these bench type tests are accelerated tests. The samples are exposed to higher speeds, higher temperatures, and higher loads then they would experience in service or during use. The number of cycles completed until failure can be read off the display of the testing apparatus.

This method for determining component lives allows for information such as safety concerns, preventive maintenance, and warranty information. One drawback of fatigue testing is that it can quickly become time consuming and expensive.

Statistical Analysis

Fatigue is probabilistic in nature. That is, it is not possible to determine the specific life when a component will fail. All that can be determined is a range or distribution for useful life.

Fatigue data analysis includes numerous approaches such as: Weibull (Weibull 1951), Johnson (Johnson, The Statistical Treatment of Fatigue Experiments 1964), (Johnson, Theory and Technique of Variation Research 1964), Vlcek Zaretsky and Hendricks (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007), (Vlcek, Hendricks and Zaretsky, Relative Ranking of Fatigue Lives of Rotating Aluminum Shafts Using L10 Weibull-Johnson Confidence Numbers 2008). With these models, engineers have been able to provide good estimates of the fatigue lives of different materials.

Weibull (Weibull 1951) developed an equation to predict the likelihood a sample would fail. The Weibull equation will be discussed in greater detail in Chapter 2. From Weibull's work, Johnson (Johnson, The Statistical Treatment of Fatigue Experiments 1964), (Johnson, Theory and Technique of Variation Research 1964) developed a method to determine whether or not one population was longer lived than the other. The bin model, used by Vlcek et al., was to construct a Monte Carlo simulation in which Weibull slope and characteristic life were used as inputs in a program and using a random number generator to create virtual samples, the samples were ranked from 0 to 1 and then using the survivability and inputs in the Weibull equation, virtual lives were constructed. These lives were used to simulate a fatigue experiment with many data points; many more than possible with just experimentation.

Suspensions/Purpose

An unavoidable circumstance of fatigue testing is that sometimes not all of the test samples fail. When a test sample does not fail it is called a suspension. This can happen for a number of reasons. Some may happen randomly such as, a power outage, or the breaker to the machine could trip. Some may be outliers as determined by the experimenter. If it is known that a specific component will never reach more than five million cycles in its life, then the engineer may choose to shut down the fatigue test as soon as it reaches the five million mark to save time. The samples that did not fail and were stopped short are considered out-lives; the name for reasons that they are outside of the range of failures to be analyzed. There are two types of suspensions, out-life suspensions and suspensions within the data. Suspensions within the data are caused by random acts of nature (e.g., the power going out). For this work only out-life suspensions were considered. These out-lives must be considered in the analysis of the data because even though they did not fail, they bias the data.

Suspensions are a useful part of the data set and should not simply be discarded. They must be accounted for, but cannot be treated the same as a failed sample (Johnson, The Statistical Treatment of Fatigue Experiments 1964). This calls for the need for a method of incorporating suspensions into a fatigue model. Methods currently exist for handling suspensions (Johnson, The Statistical Treatment of Fatigue Experiments 1964), but they are not incorporated into any of the proven Monte Carlo simulations. This work examines five models for handling suspensions within Monte Carlo simulations.

This model was developed after known methods such as Weibull statistics (Weibull 1951), the methods of Johnson (Johnson, The Statistical Treatment of Fatigue Experiments 1964), (Johnson, Theory and Technique of Variation Research 1964), and the methods of Vlcek, Zaretsky, and Hendricks (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007), (Vlcek, Hendricks and Zaretsky, Relative Ranking of Fatigue Lives of Rotating Aluminum Shafts Using L10 Weibull-Johnson Confidence Numbers 2008). Johnson developed a method to compare two sets of data with a confidence number. This number describes whether the two sets of data are significantly different or not. The model presented here uses the idea of a confidence number in a Monte Carlo simulation to determine whether one population is better than the other. This model uses five different methods of the Monte Carlo simulation to arrive at a confidence number. When these five numbers are analyzed together a conclusion can be drawn.

Purpose

When an engineer is in the design process of a new component, it may be necessary to determine which material may be best for a particular application. This leads to using statistics to compare two or more materials and to determine which one is superior for the application. These fundamentals were the basis for this research. In some experimental fatigue data, suspension points exist and must be accounted for. Five models were developed to take these out-life suspension points into account, and give the user a probabilistic analysis of the comparison of two or more materials to determine with confidence which is better for its application.

Hypothesis

It is hypothesized that the five methods proposed and examined can be incorporated into a Monte Carlo simulation of fatigue life and confidence numbers, and that it will be possible to compare or relatively rank two sets of fatigue data containing out-life suspensions.

Summary

In designing a component or system of components, it is desired to know how long that component or system can function before failure. This influences engineers to perform accelerated fatigue tests to determine the lives of particular components. Fatigue is probabilistic by nature and therefore has scatter in the data. This leads to trying to determine whether or not one population is statistically different from another. There exist methods for analyzing this data; however, the methods are limited when suspensions are introduced. It is the purpose of this work to develop a new method of comparing data sets containing out-life suspensions with a Monte Carlo simulation.

Chapter 2

Background

Fatigue

Fatigue is a mode of failure in which cyclic loading and unloading of a part or material below its ultimate tensile strength results in the development of tiny cracks which propagate through the material, ultimately leading to the failure of that part (Norton 2006). Areas of interest to engineers include automobiles, airplanes, springs, camshafts, crankshafts, bearings, gears or any other components subjected to rotation or linear modes of cyclic motion. These components constantly undergo stresses of tension, compression, bending, vibration, thermal expansion and contraction, and other stresses.

History of Fatigue Analysis

Fatigue was first noticed in the 1800s when railroad-car axles began failing. The axles were designed with static tests and were not supposed to fail. Rankine (Rankine 1843), in 1843 postulated that the axles were “crystallizing” due to the cycle bending of the axles where the wheel and axle were joined. In 1839 an engineer named Poncelet coined the term “fatigue” stating that the material was becoming brittle at the point of failure and somehow becoming “tired” from the many oscillations. In 1870 August Wohler published his 20 years of work on his investigation of axle failures. Wohler is credited with developing the rotating bending test, the S-N curve, and defining the endurance limit (Norton 2006).

Mechanism of Fatigue Failure

Most fatigue events occur in a series of stages. It usually begins with a tiny crack on the surface due to scratching of the material which may occur if lubricant is missing for an extended period of time, or it could occur due to poor design or manufacturing such as tooling marks left over from the machining process or inclusions in the material. This crack then propagates from the scratch through the material due to the stress of the cyclic load. Once the crack propagates to a level where the material can no longer sustain the forces, catastrophic failure occurs. This type of fatigue failure is common in rotating shafts. The visible evidence of failure due to fatigue is typically shown by beach marks at the point of failure. The failure area resembles a beach with ripples extending from the point of failure toward the inside of the part. Not all points of failure, however, originate from surface cracks. Some points of failure originate inside the sample. There could be an inclusion of some kind, such as a piece of material that accidentally got mixed in the manufacturing process or an air pocket. Either of these can lead to internal cracking whereby the cracks will again propagate due to the cyclic loading ultimately leading to failure.

Figures 1 and 2 are images taken with a high resolution microscope showing beach marks. The point of failure is shown in the top left of figure 1 and the top right of figure 2.

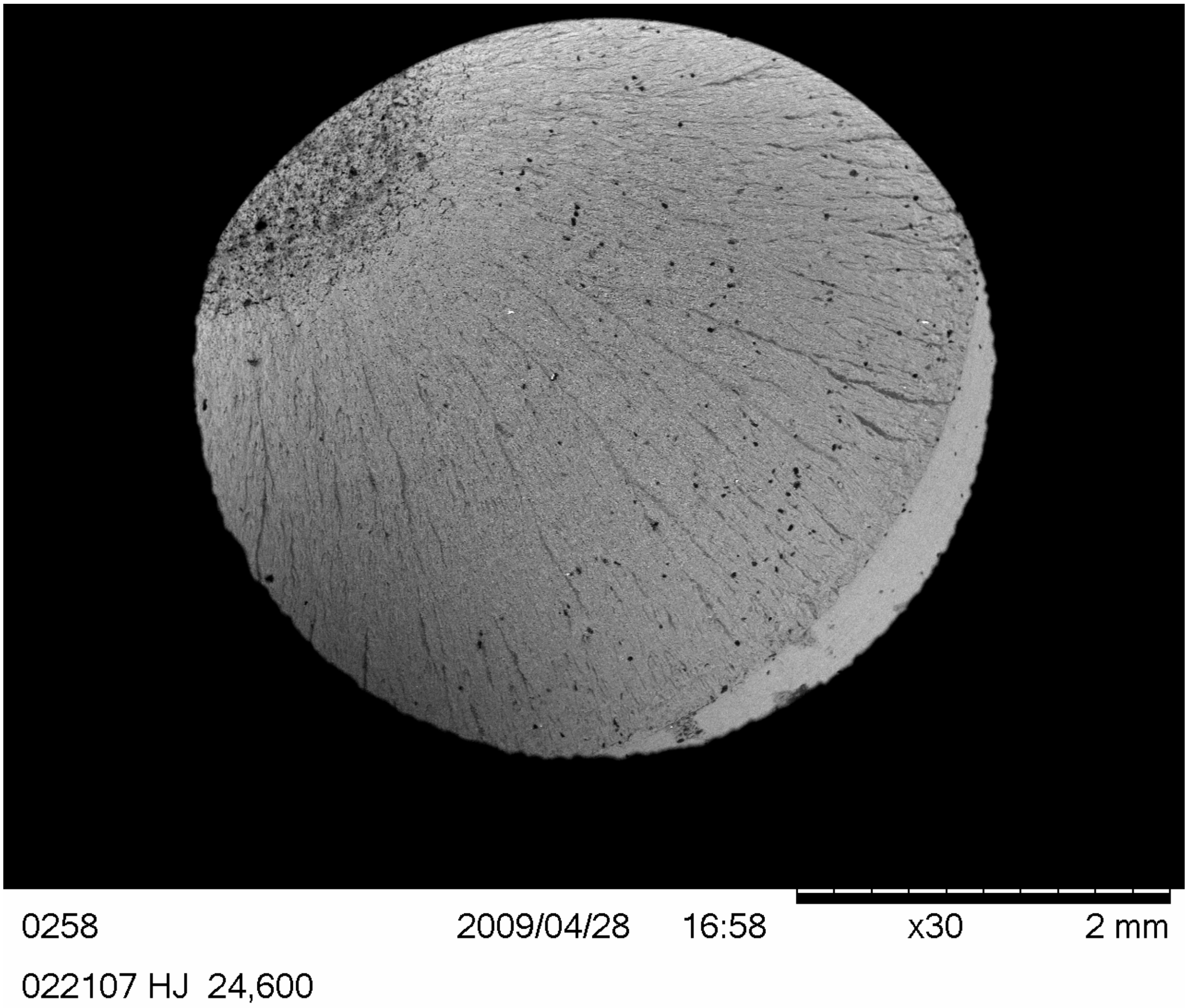


Figure 1. Unpublished image taken by Murray and Vlcek using high resolution microscope.

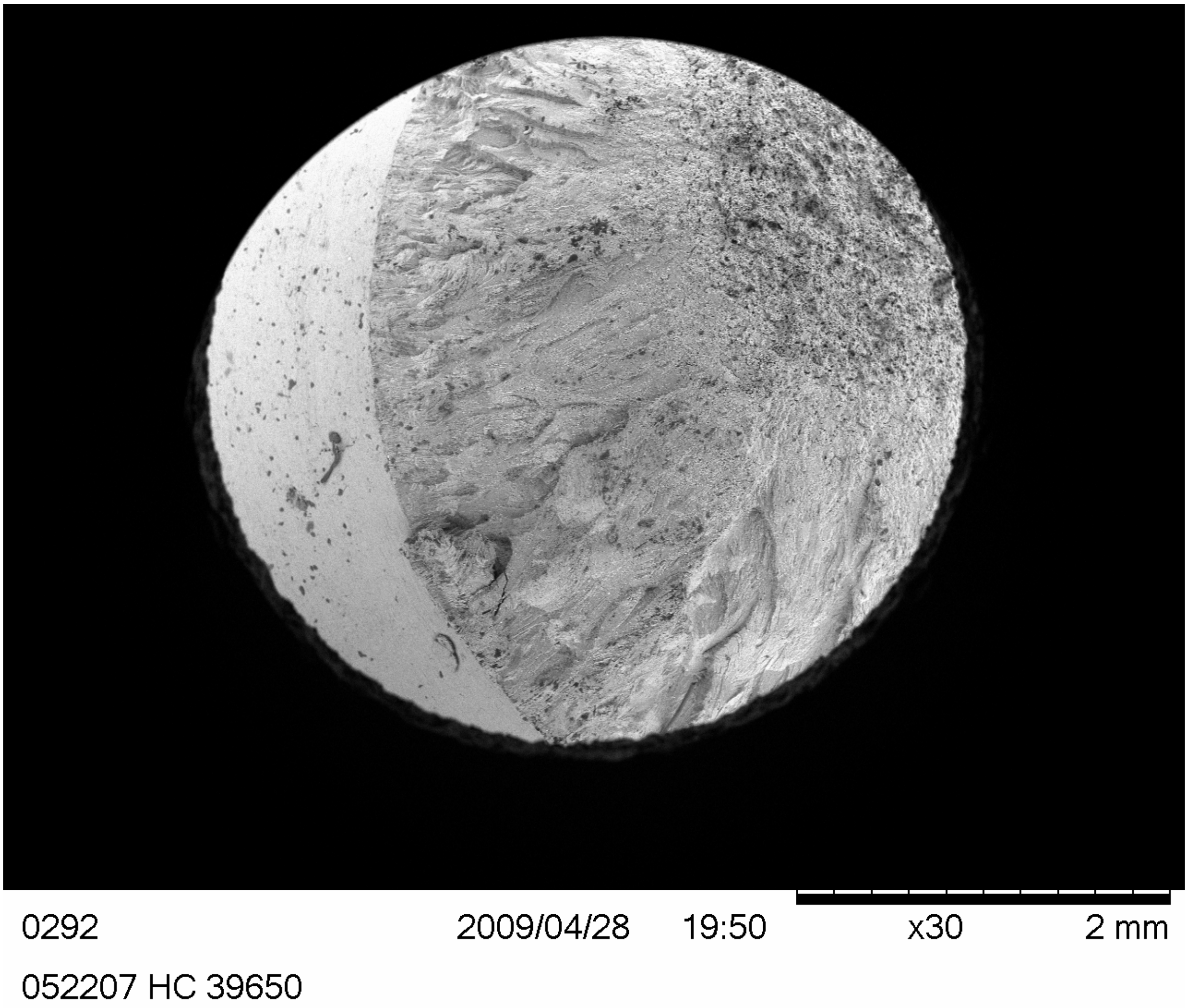


Figure 2. Unpublished image taken by Murray and Vlcek using high resolution microscope.

The fatigue life of a material is probabilistic, not deterministic (Vlcek, Zaretsky and Hendricks, Test Population Selection From Weibull-Based, Monte Carlo Simulations of Fatigue Life 2008). This causes an issue when engineers try to use methods and equations to determine when a component might fail. These methods are common in industry (Vlcek, Zaretsky and Hendricks, Test Population Selection From Weibull-Based, Monte Carlo Simulations of Fatigue

Life 2008). One way around the deterministic approach is to use factors of safety in the calculations, but then there is the risk of over designing and using too much material.

When designing a new component, it is necessary to consider factors such as secondary damage to the system, or human harm. To determine the different levels of acceptance, a statistical distribution is used to arrive at a predetermined probability of survival (Vlcek, Zaretsky and Hendricks, Test Population Selection From Weibull-Based, Monte Carlo Simulations of Fatigue Life 2008). From this distribution, the strengths or lives can be determined for values such as 90%, 95%, or 99% survival, depending on the application. As the importance of safety of the newly designed component increases, so does the necessary probability of survival (Vlcek, Zaretsky and Hendricks, Test Population Selection From Weibull-Based, Monte Carlo Simulations of Fatigue Life 2008).

One reason a system will fail due to fatigue is by excessive wear. When components come in contact with each other, they gradually wear away. To prevent this wear, it is necessary to have proper lubrication between the two surfaces. The lubrication is meant to separate the two surfaces by means of a material (liquid, gas, or even solid) that has a low resistance to shear (Oswald, et al. 2008). It is possible however that lubricant can also accelerate the means of failure. If the lubricant was to enter the crack and apply pressure, it can separate the material and cause a spall to break off.

Failure of a component due to fatigue is probabilistic in nature. There is no method to perfectly determine the fatigue life of a component. Components need to be designed not to never fail, but instead to survive a “safe” life (Zaretsky, Design for life, plan for death 1994).

Fatigue is measured by stating how many cycles or revolutions a sample of a given population, subjected to the same loads, will survive. This is typically stated as the L_{10} life which is the life at which 90 percent of samples will survive for the amount of time they were intended. Conversely, it can be said the L_{10} life is the life at which 10 percent of the population will fail. Fatigue life can also be stated as the L_{50} life at which 50 percent of the population will survive, or even the $L_{0.1}$ life which is the life at which 99.9 percent of the population will survive, for more critical components.

Failure

When a material undergoes cyclic stresses and cracks begin to form and grow, the material ultimately fractures. This is known as fatigue failure. “A failure is said to have occurred when one or more intended functions of a product are no longer fulfilled to the customer’s satisfaction” (Wasserman 2003). Minimizing failure is the primary reason for fatigue testing. “Everything is known through failure” (Tevaarwerk 2002). Without failure it would not be possible or necessary to study fatigue. If parts never failed, there would be no need for fatigue study, warranties, preventive maintenance schedules, or spare parts. If the component is in a state that can have catastrophic effects on the system, then it has failed.

Failure is what engineers try to avoid. Ideally, failure should never occur in its designated application. With proper design and maintenance, components should never fail while performing their duties. This is ideal though, so methods must be devised in case of failure during operation, such as system shut down commands.

Compromises sometimes have to be made in the design process. For example, if sufficient factors of safety were built into every component of an aircraft, the resulting weight increase would be such that the aircraft cannot achieve flight. An acceptable level of risk of failure has to be defined in view of overall system parameters to ensure effective system functionality.

Fatigue Testing

To arrive at a probability of a component or material failing it is necessary to conduct bench-top tests. These tests subject materials to different types of stresses including tension and compression, and bending. Tests are conducted at specific stress amplitudes (S) where number of cycles (N) is determined after failure. This data is then plotted on an S-N curve. From these curves it is possible to determine the endurance limit of a material.

Fatigue testing is a time consuming process. Some fatigue tests involve having a part that has failed and examining it to determine why it failed and then to redesign it so it does not happen again. Other tests involve testing the actual component before it is used in production. This allows for engineers to determine warranties and preventive maintenance schedules. The last type of fatigue testing involves using only samples of material to predict the failure of a component made from that material, as well as compare this material to similar materials to determine which is better for a specific application.

Fatigue tests are known for being lengthy, and sometimes expensive, depending on the material. Bearings and gears are designed to last for millions of cycles. It takes a very long time to test them and see how long it takes till failure. To speed up the process, a load is typically applied to the component being tested, but it could still take hours, if not days, to fail. To perform thousands of these tests would give a better idea of the failure rate of their components, but it would be too costly for the company. Because of extensive testing and costly materials, companies and engineers resort to doing the minimal, if any, amount of testing. With the data from these tests, engineers use statistical and probabilistic formulas and models to further validate their findings.

Since fatigue life is probabilistic, it is not unusual to find data sets in which the longest life to the shortest life have a ratio of 20 to 1 or even higher (Zaretsky, STLE Life Factors for Rolling Bearings 1992). This extreme ratio leads to the essential knowledge of a materials fatigue life and strength.

There are equations and analysis techniques to determine certain aspects of wear, such as crack propagations, crack thickness, lubrication thickness and properties; however, there is no definitive analysis for predicting when an application will fail (Oswald, et al. 2008). Experiments need to be conducted to determine quantitative results to make predictions using statistical methods (Oswald, et al. 2008). Reliability calculations such as bearing life are typically based on rolling element fatigue tests of the moving surfaces (Oswald, et al. 2008).

When testing a material, there are standard guidelines that should be followed. One of these guidelines is found in the ASTM Standard Practice (ASTM 1998) which discusses the correct number of specimens to test to determine an accurate S-N curve (Sutherland and Veers 2000). This standard assumes there are no run-outs or suspended items in the material; the test should be based on random samples. A summary of their recommendations are in table 1.

Table 1. ASTM Standard Practice Recommended Sample Size

Type of Test	Minimum Number of Specimens
Preliminary and Exploratory	6 – 12
Research and Development	6 – 12
Design Allowables	12 – 24
Reliability	12 - 24

Rotating Fatigue Tester

One version of the fatigue experiment is the rotating fatigue machine (figure 3).

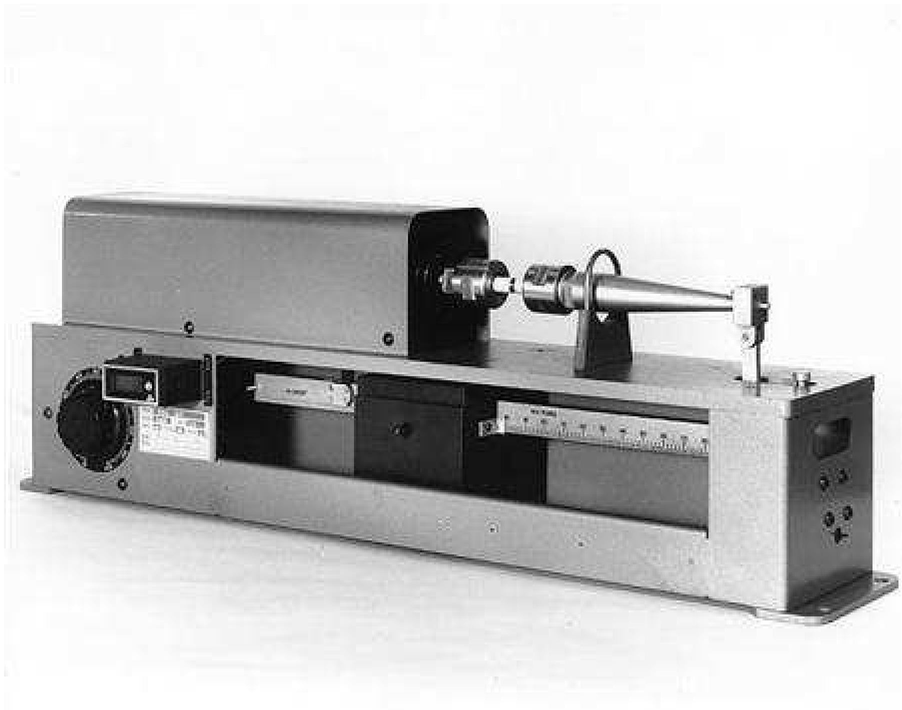


Figure 3. Bench top rotational fatigue tester (www.pci-pcmcia-express.com n.d.).

The specimen is a metal rod four inches long and one half inch in diameter. To accelerate the test, a stress concentration is introduced at the center of the specimen by removing a portion of the sample. There are different shapes that are used. Some specimens have a “V” notch machined out of them. This shape is typically used when it is desired that specimens fail quickly. Another variation is an hourglass shape. There are two different hourglass shapes used. One is a constant radius and the other has a constant diameter machined into the center of it. These take longer to fail and are more accurate. The specimen is loaded into the machine and tightened down by the use of collets. One end sits in a fixed position. The other end has the load applied to it. It is loaded by a hanging weight. This weight is adjustable to whatever is trying to be simulated. Once the load has been applied, the motor is then turned on and the specimen rotates.

The load and rotation simulate what would happen if the specimen was bent backwards and forwards repeatedly.

Servohydraulic Axial Fatigue Tester

Another popular method of performing a fatigue test is to use a servohydraulic fatigue tester (figure 4), wherein the sample is loaded vertically and subjected to tension and compression. These machines are highly programmable and capable of performing fully loaded and unloaded tests. For example, the sample can be subjected to tension and then allowed back to its rest state, or it can be compressed to the same amount of load as it was in tension. This differs from the rotational test in that the entire cross section of the material is subjected to the load.

Figure 4. Servohydraulic axial fatigue tester (www.directindustry.com n.d.).

Weibull Equation

In 1939, Waloddi Weibull developed a distribution function to aid in the statistical analysis of the fracture strength of a material. Effectively, he took a small set of fracture data and kept fitting equations to the data set until he had a linear fit. From the line, he could determine the number of cycles to failure at any probability of survival.

Weibull found that plotting the natural logarithm of the life on the abscissa and the lnln of the inverse of the probability of survivability on the ordinate resulted in a reasonably linear fit of his data. One of the most common forms of this linear fit is the 2-parameter Weibull equation which is given by equation (1)

Equation (1)

$$\ln \ln \frac{1}{S} = m \ln \frac{L_S}{L_\beta}$$

where S is the survivability, m is the Weibull slope, L_S is the life at survivability S, and L_β is the characteristic life (the life at which 63.2 percent of the samples have failed). The Weibull slope m is determined from the experimental data. It is representative of the scatter of the data points on a graph of failure probability versus life. The smaller the slope, the more scatter there is.

The Weibull equation can also be rewritten into the form:

Equation (2)

$$P(V_0) = exp\left[-\left(\frac{\sigma - \sigma_{min}}{\sigma_0}\right)^m\right]$$

This form of the equation is used to determine the probability of failure of a component or material.

Weibull's method has been under scrutiny by statisticians because of its lack of mathematical proof. However, Weibull himself was aware of this. He stated "the objection has been stated that this distribution function has no theoretical basis" (Weibull 1951). This lack of proof lies in the fact that it is hopeless to expect to be able to make predictions about random variables such as the strength properties of steels. Weibull arrived at his conclusion by choosing a function and testing it empirically until the proper results were attained. Weibull applied his method to many different circumstances where it produced satisfactory results (Weibull 1951). His examples included the yield strength of Bofors steel, size distribution of fly ash, fiber strength of Indian cotton, length of Cytroideae, fatigue life of St-37 steel, statures for adult males born in the British Isles, as well as the breadth of beans of Phaseolus Vulgaris (Weibull 1951). Weibull, however, stated that he "has never been of the opinion that this function is always valid" (Weibull 1951).

Probability paper was developed by Weibull. It is graph paper in which the abscissa is a logarithmic scale and the ordinate is a log log scale. This allows for a straight line to be fitted to plot fatigue data. Typically the fraction of the population is plotted on the ordinate and the number of cycles is plotted on the abscissa. This allows for determining the life of a component for any probability of survival (or failure).

The Weibull equation has been used by engineers for more than 80 years, particularly in the bearing industry. It will continue to be used to predict failure and lives of components. There is, however, a need for other methods of comparing data sets to justify the results of a fatigue experiment.

Comparison of Different Materials

Engineers compare two or more different components or materials to determine which is better for a specific application. Take, for instance, the main journal bearings in an automotive engine. A car manufacturer is likely to buy its bearings from a subcontractor or supplier. Before the car company makes a decision on which bearing to buy, they will test a certain amount of bearings from each company. Because of the time and cost to analyze and test the bearings, they will only test a small amount. After acquiring the lives of each bearing failed, they will perform a statistical analysis on the data set of failed bearing lives. This might include calculating the average lives of the bearings and determining the scatter in the data from the standard deviation. Next, a t-distribution test or chi-squared test might be performed to evaluate statistical differences in the data. These and other statistical methods, however, rely heavily on calculating the average first and then drawing conclusions based upon a normal distribution. In fatigue data, these methods do not take into account the large amount of scatter in the data. The data points of each bearing will typically be skewed one way or another and some data points may have a considerable amount of scatter. Likewise, some fatigue data plots have overlap, making it more difficult to determine which population is superior (figure 5). To simply take the average life does not tell the whole story.

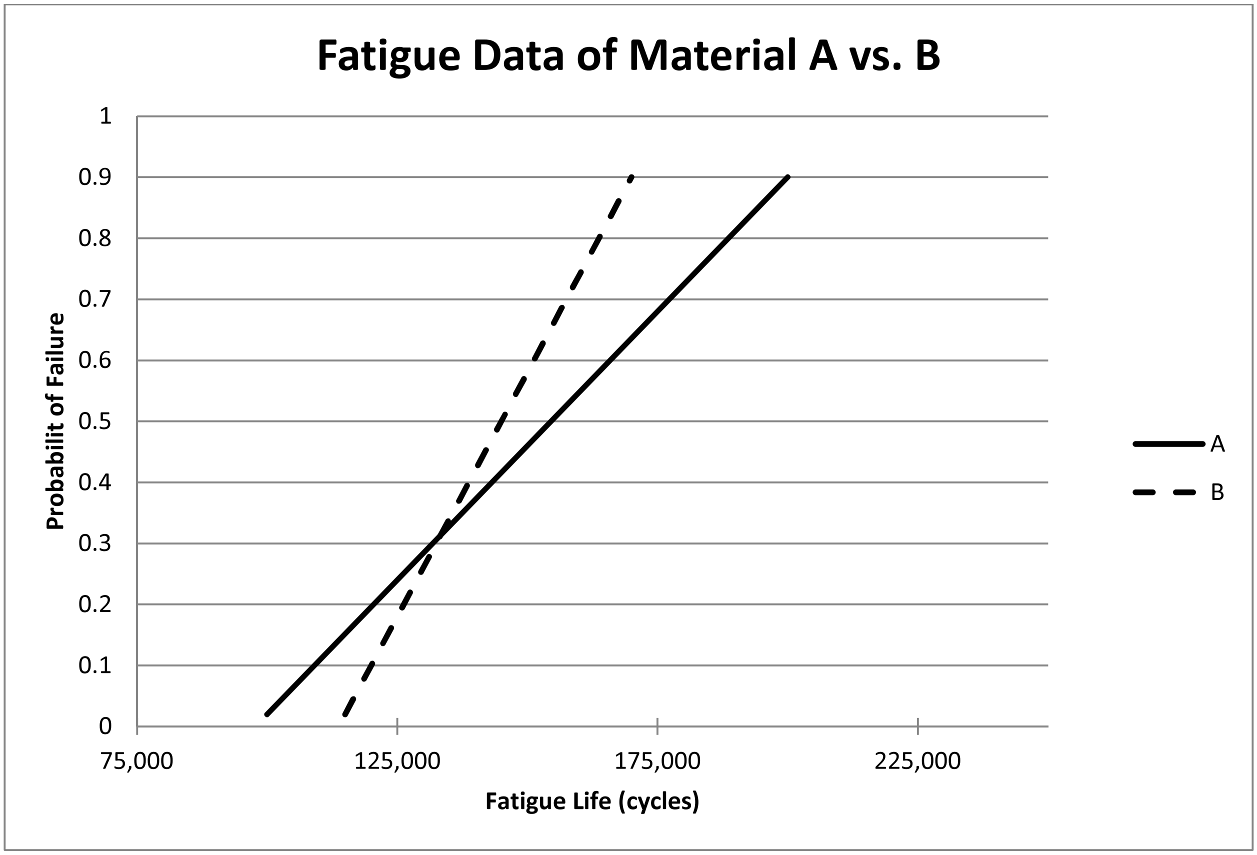


Figure 5. Graph showing overlapping fatigue data sets.

Typical statistical analysis is not enough in the area of fatigue analysis. The data acquired from fatigue tests is typically either number of cycles completed till failure, or number of hours run until failure. It could be assumed that it would be acceptable to take these numbers and average them and that this would give a good indication of the life of the material. This, however, is not the case. There exist calculations for lubrication properties and material properties, but there is a lack of mathematical methods for determining how long a component will last deterministically. It is not difficult to make new materials and understand how they work, but it is difficult to predict what they will do under stresses in a machine. This demonstrates why statistics is so important. With statistics it is possible to analyze data points with varying amounts of scatter and arrive at probabilistic conclusions.

Standard Practices

When designing a component, it is necessary to be confident whether or not it is likely to fail within a certain time period. A large number of designs use the “95/95” design values that assume there is a 95% confidence level that 95% of the components will meet or exceed the manufactures design value (Sutherland and Veers 2000). These numbers are typically determined by performing experiments and plotting the data on an S-N curve (stress-life). It is easily mistaken to get data and simply take the average of the data to determine a components limit. This statistical method cannot be used because by the definition of average, there is approximately half the material that cannot meet the standard (Sutherland and Veers 2000). This leads to the necessity of determining a “confidence limit” at which the designer has confidence that the material will meet the standard (Sutherland and Veers 2000). Since material failure is probabilistic, the design engineer is dealing with random variables. The lives at which the parts fail is the random variable, thus the designer must use a value for the strength or life that is guaranteed (Sutherland and Veers 2000).

Due to the high cost and extreme time lengths of fatigue tests, there are typically only ten or less data points to analyze. Therefore, it is essential that the maximum amount of information gained from that data is analyzed properly (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007). One method that is used far too often is to determine a mean and median of that data and use this as the basis of the results. This is inaccurate because it fails to take into account the scatter in the data (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007). One may also choose to plot the data on an S-N curve and try to draw conclusions this way. This is a better step forward; however, it is still not the whole story. The

ASTM standard (ASTM 1998) describes the use of these curves; however, it does not present a method to determine with confidence the difference between two curves (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007). The inadequacy to fully understand the data from this standard is stated by the writers of the report (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007) "As alternate fatigue models and statistical analysis are continually being developed, later revisions of this practice may subsequently present analyses that permit more complete interpretation of S-N and ε-N data."

Johnson's Method

In the 1950's, Leonard Johnson, an engineer for General Motors, developed a technique to rank materials in a fatigue test to determine which was better by the use of confidence numbers. His goal was to provide methods outside of standard statistical practices. Johnson explains that we cannot solely rely on the average of fatigue data because the average is only one number. It does not take into account the scatter of the data. Johnson even uses the quip, "a fellow who fell in love with a dimple and then made the mistake of marrying the whole girl" (Johnson, Theory and Technique of Variation Research 1964).

Johnson stressed the importance of statistics in the field of fatigue testing. Since fatigue is probabilistic in nature, then statistics is the only means of predicting when a particular component fails. He mentions how the Weibull equation developed in the 1930's is the only method to date to provide any kind of statistical analysis of fatigue data.

Johnson begins his discussion by describing how fatigue data is usually very scattered. Even with identical test conditions such as temperature, load, and samples size, there is still a large spread in the data. He uses the lives from a sample fatigue experiment to order them in integers from 1 to n, then uses these integers to rank each life from 0 to 1. The rank number is the fraction of the population of the data set that has been accounted for up to a particular value. The rank comes from the figures in his book or from the rank equation (equation 3).

Equation (3)

$$\text{Median Rank} = \frac{\text{random number} - 0.3}{\text{bin size} + 0.4}$$

The rank, as defined by Johnson, is the median rank. That is, the mid value with equal fluctuation above and below the value (Johnson, Theory and Technique of Variation Research 1964). Subsequently he is able to draw a line on Weibull probability paper to determine how the whole population would act. When plotting fatigue data on Weibull probability paper, the rank is plotted against the fatigue life. An example of a set of fatigue data that is ordered and ranked, and then graphed (not a Weibull plot) is shown in table 2 and figure 6. The graph allows for viewing the data set as a whole rather than relying on just an average.

Table 2. Sample fatigue data demonstrating median rank

Order Number	Revolutions	Median Rank
1	102,300	0.109
2	110,563	0.266
3	120,910	0.422
4	129,740	0.578
5	130,400	0.734
6	135,200	0.891

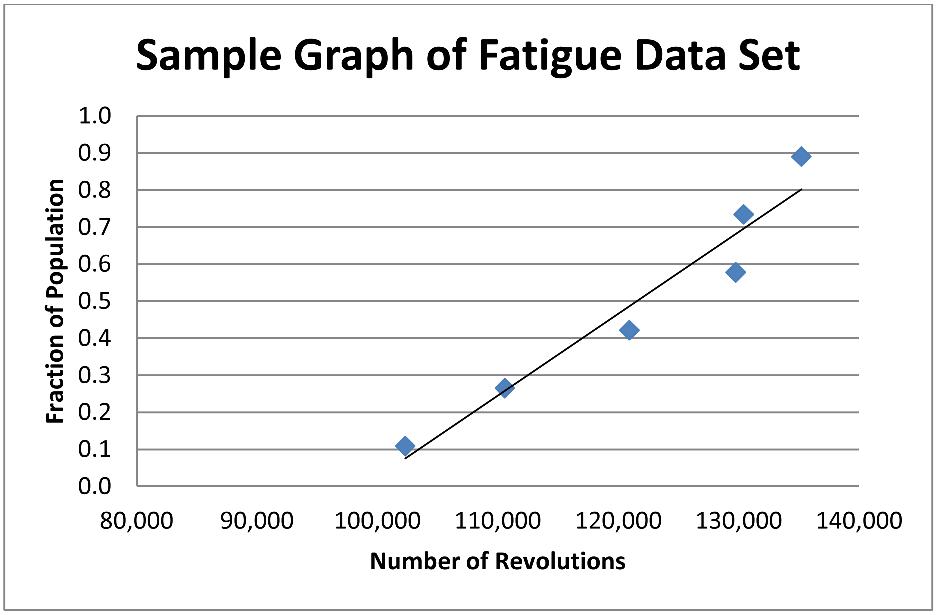


Figure 6. Graph showing least square fit of sample fatigue data

He then explains further how to determine how accurate one would expect this probability line to be with the confidence bands he developed.

Confidence Bands

With the median rank, and the knowledge of inherent fluctuations, a confidence band can be constructed. This band shows the amount of certainty there is in the true location of a point on the graph that lies within that range. Since the median rank is the average of the fluctuation at that point, it is safe to say that at that value half of the population will either be above or below that value. So to construct a 90 percent confidence band (confident that 90 percent of the population will fall within it), as in figure 7, it is necessary to know the limit of the lower 5 percent and the upper 95 percent. When these limits are known for each median rank, a confidence band can be drawn on probability paper. Even with these bands though, there is still a need to determine whether or not one material is statistically different from another when two data sets are compared.

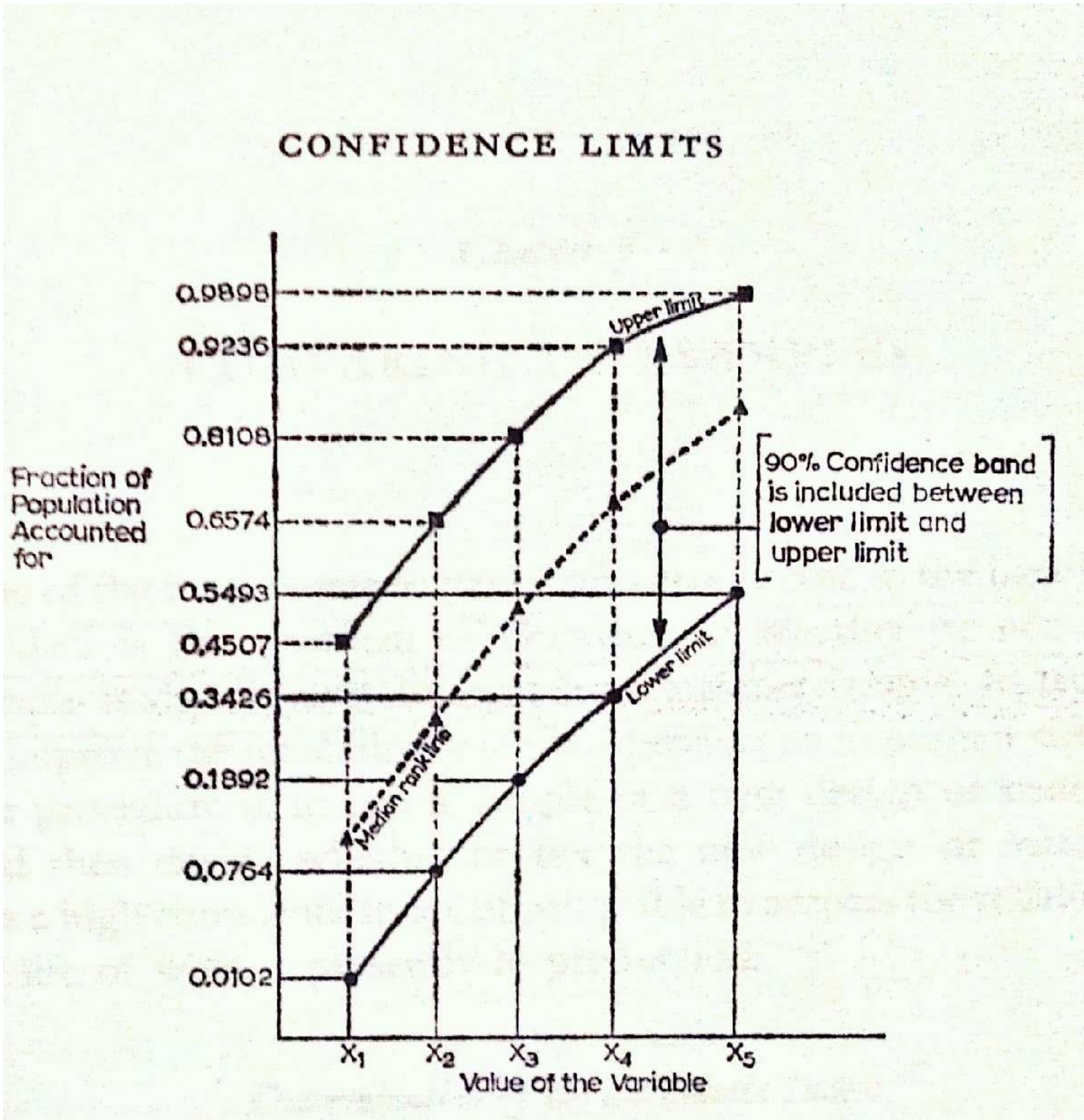


Figure 7. Graph of confidence bands (Johnson, Theory and Technique of Variation Research 1964)

Confidence Numbers

Confidence numbers are another useful tool developed by Johnson (Johnson, The Statistical Treatment of Fatigue Experiments 1964), (Johnson, Theory and Technique of Variation Research 1964). A confidence number can be arrived at to determine whether or not one population of fatigue data is significantly different from another population. A confidence number is the number of times a specific variable of a fatigue test will be better in one population than the other if the experiment and comparison were repeated 100 times. For example, if the L_{10} life of material A was higher than the L_{10} life of material B, 92 out of 100 times, then the confidence number would be 92. It is accepted that a confidence value of 90 or higher is statistically significant (Johnson, The Statistical Treatment of Fatigue Experiments 1964). His argument was that, given two sets of fatigue data and plotting them on Weibull paper yields an estimate of the lives of that population. If the two plots have equal slopes and lie relatively close to one another on the graph paper, then how is one to know whether or not they are really different. Another case could be where the two plots overlap. Say material A has a longer L_{10} life, but material B has a longer L_{50} life, again how to determine whether or not statistically different.

Johnson addresses comparing or ranking the fatigue life of two materials with his method of determining a confidence index (Johnson, Theory and Technique of Variation Research 1964). Johnson developed confidence curves from which a confidence number, as a function of life ratio, Weibull slope (scatter in the data) and total degrees of freedom (populations sample size), could be graphically read. To arrive at this confidence index, both data sets must first be plotted on Weibull probability paper with the percent of population failed on the ordinate and the lives on the abscissa. From these lines the Weibull slope of each set can be determined by taking the tangent of the angle the line makes with the horizontal and either the L_{10} life or mean life can be read.

Parameters to Determine Confidence Number

The total degrees of freedom of the set are the degrees of freedom of the first set (n-1) multiplied by the degrees of freedom of the second set (n-1).

Equation (4)

$$DOF = (n_1-1) \times (n_2-1)$$

The degrees of freedom is how many choices are allowed when choosing something. For example, if there is a box with 5 gears, the first choice could be any of the 5; the next choice could be any of the remaining 4, and so on until there is one left. When one is left there is no choice to be made, the final one has to be chosen. This is why the degrees of freedom is one minus the size of the population.

The life ratio is calculated by dividing the larger life of the two sets by the smaller life. This can be the mean life or the L_{10} life. With the life ratio, Weibull slope, and total degrees of freedom, a confidence number can be determined by Johnson's figures. An example of one of Johnson's figures is illustrated in figure 8.

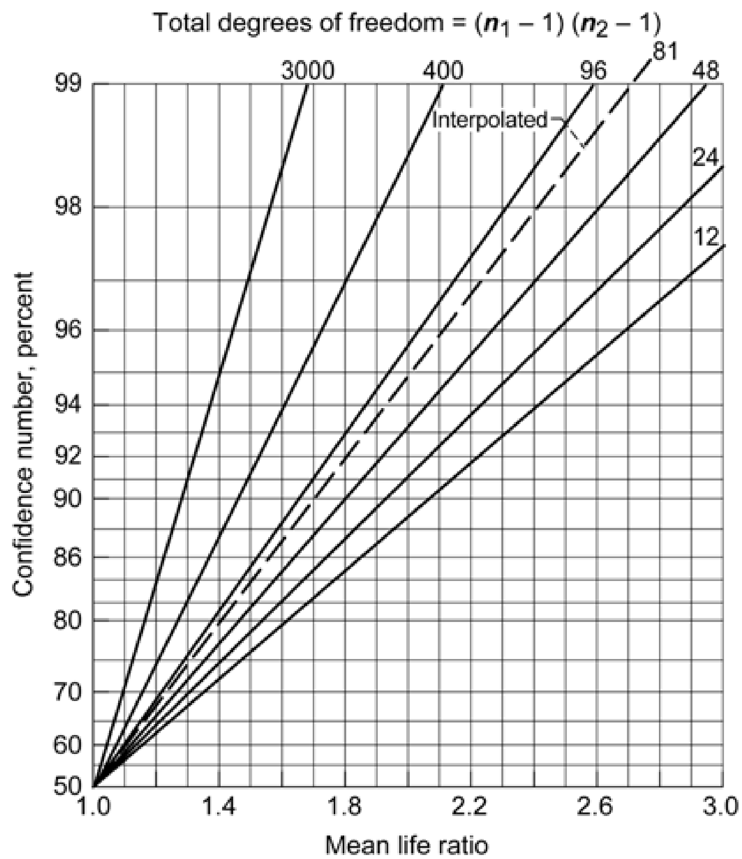


Figure 8. One of Johnson's figures for determining confidence number (Johnson, The Statistical Treatment of Fatigue Experiments 1964).

If the Weibull slopes are not equal, then the confidence number must be determined by calculating the average of the two confidence numbers found from the respective graphs associated with each slope.

Suspensions

Failure testing, like many things in research, does not always go as planned. There are situations called suspensions in which, for whatever reason, the test was stopped and the sample was not allowed to fail. Reasons for suspensions include, a power outage, someone trips over the power cord and causes the machine to shut off, or the test is stopped at a predetermined time. When a sample is stopped at a predetermined time this is known as an out-life suspension. It is considered an out-life because it is outside of the range of the samples that failed. Obviously, unforeseen things can happen and turn a machine off. These, however, are not lost data. They are taken into account in the analysis by their own method.

Also, sometimes tests are stopped when the component runs for hours and hours. It is possible that sometimes the tester does not want to wait for a part to fail when it has already far exceeded what it was suppose to. If a part is designed to last 10 million cycles and there is a batch of 20, if 15 fail within the 10 million and the other 5 begin to reach 20 million, the test may be cut short to save time because these are obviously outliers. Suspensions are common in fatigue testing and they should not be discounted as confounding data. Everything is considered in fatigue analysis.

Johnson gives three examples of causes of suspended items: (Johnson, Theory and Technique of Variation Research 1964)

1) A special need to terminate a test before all the original N specimens have failed, such as shortage of time or testing equipment.
2) A failure of a different nature from the one being tested, e.g., a large bore in a bearing would make it impossible to continue running that particular bearing until it exhibits pitting fatigue as originally intended.
3) A desire to make an analysis before the test has been completed.

Johnson's Analysis Method of Suspensions

Johnson (Johnson, The Statistical Treatment of Fatigue Experiments 1964), (Johnson, Theory and Technique of Variation Research 1964) derived a method for ranking fatigue data sets that included suspensions. This allowed for a more accurate Weibull plot. These suspended items are referred to as suspensions.

Johnson defined a test in which all samples failed as a complete test and one where not all samples failed as an incomplete test. The number of samples failed out of tested in an incomplete test is known as the failure index. According to Johnson, it is not correct to treat an incomplete test the same as a complete test because the suspended samples contain information which could affect the predicted positions of the actual failures of the complete population. Recall that his method involves assigning integer values to each life so that they can then be ranked. To account for suspended samples, this ordering and ranking needs to change. These integer values become fractional values. The integer values no longer hold because, if a suspended sample has a value between two failed samples, it is uncertain whether that sample was going to fail or not before the sample with the longer life, or after it. Thereby its order number could be before or after the failed sample. Johnson developed a new method to order suspended samples. Rather than ordering the samples using the median rank equation (equation 4), Johnson defines changing the increment using equation 5 when suspensions are present in the data set.

Equation (5)

$$\text{new increment} = \frac{\text{(n+1)-(previous mean order number)}}{\text{1+(number of items beyond present suspended set)}}$$

Once the new order numbers are assigned to the lives, then the ranks can be found in the tables. Since the tables are for order numbers of integers, it is necessary to interpolate the correct value for the new fractional order numbers. These ranks can then be used the same as for a complete test to determine a confidence number of the two populations.

As an example of this method consider the fatigue data set in table 3.

Table 3. Sample fatigue data set to illustrate Johnson's suspension incremental method		
Number	Life (hours)	
1	12	Failed
2	25	Failed
3	26	Suspended
4	33	Failed
5	47	Suspended
6	52	Failed
7	71	Failed
8	79	Suspended
9	83	Failed
10	95	Failed

This data set contains ten samples, three suspensions and seven failures. Using equation 5 the new mean order number can be calculated followed by the new median rank. The new mean order numbers and median ranks are shown in table 4.

Table 4. New mean order numbers and median ranks for sample fatigue data				
Number	Life (hours)		Mean order number	Median Rank
1	12	Failed	1.00000	0.0670
2	25	Failed	2.00000	0.1632
3	26	Suspended	---	---
4	33	Failed	3.12500	0.2715
5	47	Suspended	---	---
6	52	Failed	4.43750	0.3978
7	71	Failed	5.75000	0.5241
8	79	Suspended	---	---
9	83	Failed	7.50000	0.6924
10	95	Failed	9.25000	0.8608

"Neglecting suspensions and assuming a complete test of magnitude equal to failure index amounts to assigning too high a population rank to each failed item. This causes the Weibull plot to be shifted upward, making the life estimate more conservative than necessary"

(Johnson, The Statistical Treatment of Fatigue Experiments 1964). This would lead to over designing components and again wasting material and money.

Limitations of Johnson's Method

The use of Johnson's method for determining the difference of two populations has fallen out of wide spread use due to the difficulty of his interpolating between his figures and the lost literature on how he arrived at his method (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007).

The problem with Johnson's method is that he has a limited number of Weibull slopes and degrees of freedom of which his charts allow for determining confidence numbers. His methods and equations for coming up with these charts has been lost. His method is not easily used for many points of data and it can be quite time consuming.

Monte Carlo Methods

The incorporation of Weibull-Johnson Monte Carlo simulations to the statistical analysis field gave another dimension: another means of calculating and predicting the outcomes of components. The application of a Monte Carlo simulation to fatigue data to determine statistical reliability and confidence numbers has been demonstrated in the simulation of fatigue lives of bearings (Vlcek, Zaretsky and Hendricks, Test Population Selection From Weibull-Based, Monte Carlo Simulations of Fatigue Life 2008), (Vlcek, Hendricks and Zaretsky, Determination of Rolling-Element Fatigue Life From Computer Generated Bearing Tests 2003), (McBride 2011).

Generic Monte Carlo Simulation

A Monte Carlo simulation is a mathematical process that combines user inputs and random variable input(s) to a mathematical equation to simulate possible outcomes. This random process is repeated many times to establish trends, if not absolute magnitude (Rubinstein 1981). A flowchart showing the basic steps of a Monte Carlo simulation is shown in figure 9.

The term Monte Carlo was first introduced by von Neumann and Ulam during World War II as a secret code word at Los Alamos. It was in reference to the gambling casinos in Monte Carlo, Monaco (Rubinstein 1981). According to Haldar (Haldar and Mahadevan 2000), the Monte Carlo simulation is comprised of six elements: "1) defining the problem in terms of all the random variables; 2) quantifying the probabilistic characteristics of all the random variables in terms of their PDFs or PMFs and the corresponding parameters; 3) generating the values of these random variables; 4) evaluating the problem deterministically for each set of realizations of all the random variables, that is, numerical experimentation; 5) extracting probabilistic information from N such realizations; and 6) determining the accuracy and efficiency of the simulation."

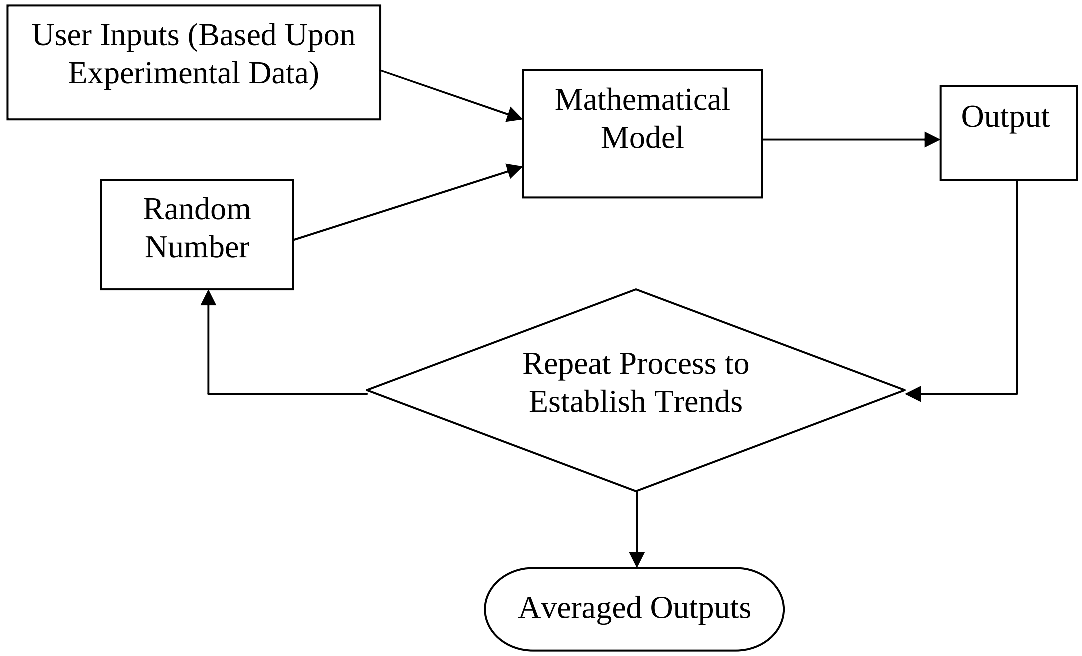


Figure 9. Flow chart of a basic Monte Carlo simulation.

Computer simulations are sometimes considered a “last resort” method. With the current rise in technology and advances in computer systems, however, computer simulations have become one of the most widely used and accepted tools for analysis (Rubinstein 1981).

The life of mechanical components is not deterministic. Data, design curves, and formulas exist for calculating parameters with deterministic variables, however, this is not necessarily the case when the lives of the components are probabilistic. In the case of studying the robustness of planetary gears, a Monte Carlo simulation was used to account for this probabilistic nature (Enguo, Lei and Yanyum 2010). The deterministic methods of determining the robustness of the planetary gears was employed and then compared to the Monte Carlo method. The deterministic approach gave values that resulted in poor analysis of the robustness of the materials which lead to premature values. When the Monte Carlo method was applied, the results gave a better prediction of the life of the gears, which resulted in less cost as well as better components (Enguo, Lei and Yanyum 2010).

Monte Carlo Method used for Pattern Recognition

"The design, analysis, and verification and validation of a spacecraft rely heavily on Monte Carlo simulations" (Restrepo and Hurtado n.d.). Space travel is a very expensive endeavor and requires many engineering man hours to ensure the safety and reliability of a spacecraft. With the incredibly high expenses, it is not possible to test every system and determine all the possible outcomes. Testing, therefore, is limited. Monte Carlo simulations allow engineers to input different variables and let the program run to generate different random outcomes (Restrepo and Hurtado n.d.). This simulation, run long enough, could potentially display most of the possible outcomes of disasters and allow the engineers to design accordingly. There is, however, one problem with this method that was addressed in (Restrepo and Hurtado n.d.). With the enormity of possibilities of outcomes it can become very difficult to go through all the data. Restrepo and Hurtado have developed a plan to attack large amounts of data from a Monte Carlo simulation using pattern recognition. "Given enough time the Monte Carlo approach allows analysts to identify most of the individual design variables that influence certain system failures" (Restrepo and Hurtado n.d.). It typically, however, is not an individual parameter that causes a failure; it is a series of complex anomalies that lead to system failure (Restrepo and Hurtado n.d.). It is this series of events that an engineer must try to predict to aid in the design process of the system. "The goal of a Monte Carlo simulation is to understand all critical design sensitivities that may prevent the design from meeting requirements" (Restrepo and Hurtado n.d.).

Monte Carlo simulations can give an extreme amount of data. This leaves the engineer to sift through great quantities of data. One drawback to this is that the engineer needs some kind of already known intuition about the system to make judgment calls. Also, it is required to have a sound knowledge in the area of the research. Restrepo and Hurtado (Restrepo and Hurtado n.d.) have devised a method to escape this problem.

To evaluate the accuracy of more sophisticated statistical techniques, or to verify a new technique, simulation is routinely used to independently evaluate the underlying probability of failure (Haldar and Mahadevan 2000). “In the simplest form of the basic simulation, each random variable in a problem is sampled several times to represent its real distribution according to its probabilistic characteristics. Using many simulation cycles gives the overall probabilistic characteristics of the problem, particularly when the number of cycles N tends to infinity. The simulation technique using a computer is an inexpensive way (compared to laboratory testing) to study the uncertainty in the problem” (Haldar and Mahadevan 2000). The most commonly used simulation for this purpose is the Monte Carlo simulation. Engineers have been using this tool because of its ease of use and its accuracy of results. A strong background in statistics and probability is not needed to develop a Monte Carlo simulation. These are the reasons why engineers use Monte Carlo simulations for evaluating the risk and reliability of complicated systems (Haldar and Mahadevan 2000).

Preventive Maintenance

Preventive maintenance schedules are crucial to ensuring machines and systems operate properly and that no harm is done. Preventive maintenance can be defined as “a fundamental, planned maintenance activity designed to improve equipment life and avoid any unplanned maintenance activity” (Wireman 2008). With the analysis of fatigue data it is possible to determine these preventive maintenance schedules as well as warranty information.

The Space Shuttle is one example of the critical importance to ensure human safety and machine reliability. The space shuttle was designed to function for 100 flights per shuttle without maintenance or inspection (Oswald, et al. 2008). One system, in particular, is the body flap actuators on the space shuttle. There are four actuators on each shuttle (two per wing). After several flights of the shuttle, the bearings of these actuators were inspected for wear. Due to the varying degrees of wear, analysis had to be performed to determine proper timing of removal and replacement of these bearings (Oswald, et al. 2008).

This analysis was performed and the objectives of the research were: "a) experimentally duplicate the operating conditions of the space shuttle body flat actuator input shaft ball bearings; b) generate, under these simulated conditions, a statistical data base codifying bearing wear; c) determine the usable life of the actuator bearings based on a two-parameter Weibull distribution function for the bearings using strict-series system reliability; and d) compare these results to field data from the space shuttle fleet (Oswald, et al. 2008)." The statistical methods to analyze this data was both Weibull (Weibull 1951) and Johnson (Johnson, The Statistical Treatment of Fatigue Experiments 1964), (Johnson, Theory and Technique of Variation Research 1964). These methods have been in use by NASA for over 50 years in the area of failure analysis of bearings and gears in which a large database now exists (Oswald, et al. 2008).

Using the probabilistic method on the actuators, it was predicted that the bearing would fail after 20 missions. This was in close agreement to the actual failure at 22 missions (Oswald, et al. 2008). Of 116 missions between 1981 and 2006, it was reported that only one actuator bearing had to be replaced due to excessive wear (Oswald, et al. 2008).

During the experiment, one of the tests conducted to determine the life of a bearing included six bearings in which sudden death testing was used that resulted in three of the six bearings failing (Oswald, et al. 2008). The analysis used to incorporate the suspended items was that of Johnson (Johnson, The Statistical Treatment of Fatigue Experiments 1964).

Along with testing for failure comes the knowledge of how to prevent failure in the first place. During failure testing, engineers learn methods to prevent failure. Different tolerances and lubrication methods lead to longer failure lives. The only way to prevent or postpone failure is to maintain the machine. Proper maintenance schedules need to be written and enforced. Failure tests, proper lubrication, and maintenance schedules are one way to prevent failure.

Summary

Fatigue has been studied since the mid 1800's when train axles started breaking unexpectedly. It turned out that this phenomenon was probabilistic and could not be precisely predicted. In the 1930's Weibull developed a method to analyze this probabilistic occurrence. Although this method has been criticized by mathematicians it has worked for years and engineers still use it. Leonard Johnson developed a way of ranking two populations of fatigue data. From his method it is possible to statistically determine whether or not one population is better than the other.

Recently, with the advent of computers, Monte Carlo simulations have been developed to analyze fatigue data. A Monte Carlo simulation uses a limited number of predetermined inputs and random numbers to simulate possible outcomes. This means of comparing fatigue data has been demonstrated to work.

Chapter 3

Method

Introduction

The purpose of this research is to compare fatigue data sets containing out-life suspensions via a Monte Carlo simulation. Existing methods for handling such suspensions can be tedious, relying heavily upon graphical interpretation and interpolation of design curves (Johnson, The Statistical Treatment of Fatigue Experiments 1964). This current project takes advantage of technological computing power to handle a statistically significant number of simulations.

A model was developed to simulate experimental fatigue data that contains suspensions. A Monte Carlo simulation was written in Visual Basic, to interface with Microsoft Excel, to simulate fatigue lives using a "bin" model developed by Vlcek, Hendricks and Zaretsky (Vlcek, Hendricks and Zaretsky, Determination of Rolling-Element Fatigue Life From Computer Generated Bearing Tests 2003). Different suspension models were evaluated. The models differed in how the failure index (number of failed samples or total number samples tested) was modeled prior to determining the L_{10} life for the data set. A relative ranking counting method similar to that demonstrated by McBride, Vlcek, and Hendricks (McBride 2011) was used to determine the relative confidence numbers associated with each suspension method. The simulation was repeated a statistically significant number of times (10,000). For validation of the model, the simulated confidence numbers were compared to those graphically determined for two experimental data sets that were available in the literature (Townsend, Zaretsky and

Anderson, Comparison of Modified Vasco X-2 with AISI 9310 - Preliminary Report 1977), (Townsend and Zaretsky, Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels 1980).

Monte Carlo Simulation - Weibull Equation, Confidence Numbers, and "Bin" Model

The model was generated by combining three statistical methods already in existence. The first was the use of the Weibull equation (equation 1) to determine a life. Using a random number generator, a number between 1 and 1000 was generated. This number represented the order in which the sample failed (this is known as the "bin" model and will be discussed in more detail in the next section). From this the order number was converted to a rank using equation 3. This number subtracted from 1 gave the survivability (S) of the sample. Then, using the Weibull slope and characteristic life as determined from experimental data, a virtual life can be solved for algebraically.

The second method incorporated into the model was developed by Johnson (Johnson, The Statistical Treatment of Fatigue Experiments 1964). Johnson developed a way to statistically predict whether or not one material was better than another. He did this through confidence numbers. If two materials were analyzed, and a confidence number of 90 or higher was determined, then it implies that there is a statistical difference between the two materials. Although his graphical method of calculating confidence numbers was not incorporated into this model, the idea behind a confidence number was—how many times out of one hundred, one probabilistic value was greater than another. Johnson also developed a method for incorporating suspensions into the determination of a confidence number, and demonstrated the importance of taking suspensions into consideration instead of just dismissing them.

The third method incorporated into this model was that of Vlcek et al. (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007), (Vlcek, Hendricks and Zaretsky, Relative Ranking of Fatigue Lives of Rotating Aluminum Shafts Using L10 Weibull-Johnson Confidence Numbers 2008), (Vlcek, Hendricks and Zaretsky, Determination of Rolling-Element Fatigue Life From Computer Generated Bearing Tests 2003), (McBride 2011). Vlcek et al. developed what is referred to as a virtual bin model. The bin model is a virtual bin of 1000 specimens that are assumed to have been tested to failure. The lives of these samples are not known at this time, however, the order in which they failed is known, and each sample has an order number from 1 to 1000 associated with the order in which it failed. With this order number, it is possible to determine the survivability of the sample using the median rank equation (equation 3). If the Weibull slope (m) and characteristic life (L_β) are known, then the simulated life is the only unknown in the Weibull equation (equation 1), and can be algebraically solved for.

Another method used that was developed by Vlcek et al. was that of numerically counting simulated lives to arrive at a confidence number. Since a confidence number is defined as the number of times out of 100 one material will be better than another, with a Monte Carlo simulation generating 100 virtual lives it is possible to count how many times the life of one material is greater than the other.

The last method used by Vlcek et al. in this simulation was the development of curve fit equations, which were derived from Johnson's figures, to calculate a confidence number. These equations were incorporated into the simulations, to use the variables generated, to calculate a confidence number. This added one more dimension for comparison of confidence numbers. Ultimately three confidence numbers were able to be compared to determine whether or not one material was better than the other.

Bin Model Monte Carlo Simulation

The objective of the Monte Carlo technique in this application is to acquire virtual lives from random numbers between 1 and 1000 which represent samples in a bin ordered in which they failed. These numbers were converted to lives by use of the rank equation (equation 3) and the Weibull equation (equation 1).

These random numbers represent the order in which the gears would have failed had all 1,000 been tested. For example, if the random number generator used by the simulation generates the number 784, this would represent the 784th sample to fail in the bin of 1,000. Since the true life of the 784th sample that failed is not known, it will be sequentially ordered against the other samples in its bin. Sample 1 will have the shortest life, and sample 1,000 will have the longest life.

The random number generator itself does not have a method for determining how many of a certain number it picks. For example, it is possible that the number 256 gets generated twice. This is not desired because it is not possible to have two samples with the order number 256. This issue was addressed to ensure unique random numbers. No two samples can have the same rank. A Visual Basic Module in Excel was written to ensure that all random numbers or parts pulled from the bin were unique in a test set. This module can be seen in Figure 10.

```
'Bin A
'generating random number
alpha = Cells(9, 2)
r = 7
num1 = 1
For loop1 = 1 To alpha
    Cells(r, 4).Select
here2:
    ActiveCell.FormulaR1C1 = "=randbetween(1,1000)"

'Checking for duplicate
s = 6
For randcheck = 1 To num1
    If Cells(r, 4) = Cells(s, 4) Then GoTo here2
    s = s + 1
Next randcheck
    r = r + 1
    num1 = num1 + 1
Next loop1
```

Figure 10. Subroutine to generate random number and make sure all numbers are unique

It should be stated now that in Visual Basic, any text following a (‘) is a comment and not used in the program. They are for organizational purposes only.

The virtual test samples were ranked on a scale from 0 to 1. Zero being the lowest life and one being the longest life. To rank the samples equation 3 was used

Equation (3)

$$\text{Median Rank} = \frac{\text{random number} - 0.3}{\text{bin size} + 0.4}$$

Where random number is the number the program chooses and bin size is 1000. After ranking the samples from 0 to 1, the rank is subtracted from 1 to get the survivability S of the sample (equation 7).

Equation (7)

$$S = \text{Rank} - 1$$

Lives are assigned to the virtual samples based on the experimental data using the Weibull equation.

Equation (1)

$$\ln \ln \frac{1}{S} = m \ln \frac{L_S}{L_\beta}$$

The rank from the simulation results in survivability S. The Weibull slope m and characteristic life L_β come from previously available experimental data. The virtual life at a survivability probability S can be calculated after plugging these three variables into equation 1. The lives, still in order from shortest lived to longest lived, were then ranked again. This time they were ranked according to the test population size. If 20 samples were used in the experiment, then in the ranking equation, 20 was used for the population size n, and instead of a random number, 1 to the sample size was used; in this case, from 1 to 20 (equation 8).

Equation (8)

$$\text{Rank} = \frac{\text{number} - 0.3}{20 + 0.4}$$

The survivability S, or percent of samples that survived, is again solved for by using equation 7. The lnln (1/S) is plotted as a function of the ln (L). lnln (1/S) is plotted on the ordinate and ln (L) is plotted on the abscissa. As a result, fatigue data plots as a relatively straight line on Weibull paper, and a fitted curve is determined using a least squares fit. The Weibull slope, characteristic life, L_{10} life, L_{50} life, and mean life are results of this plot. Figure 11 is a flowchart of the simulation based on a bin model.

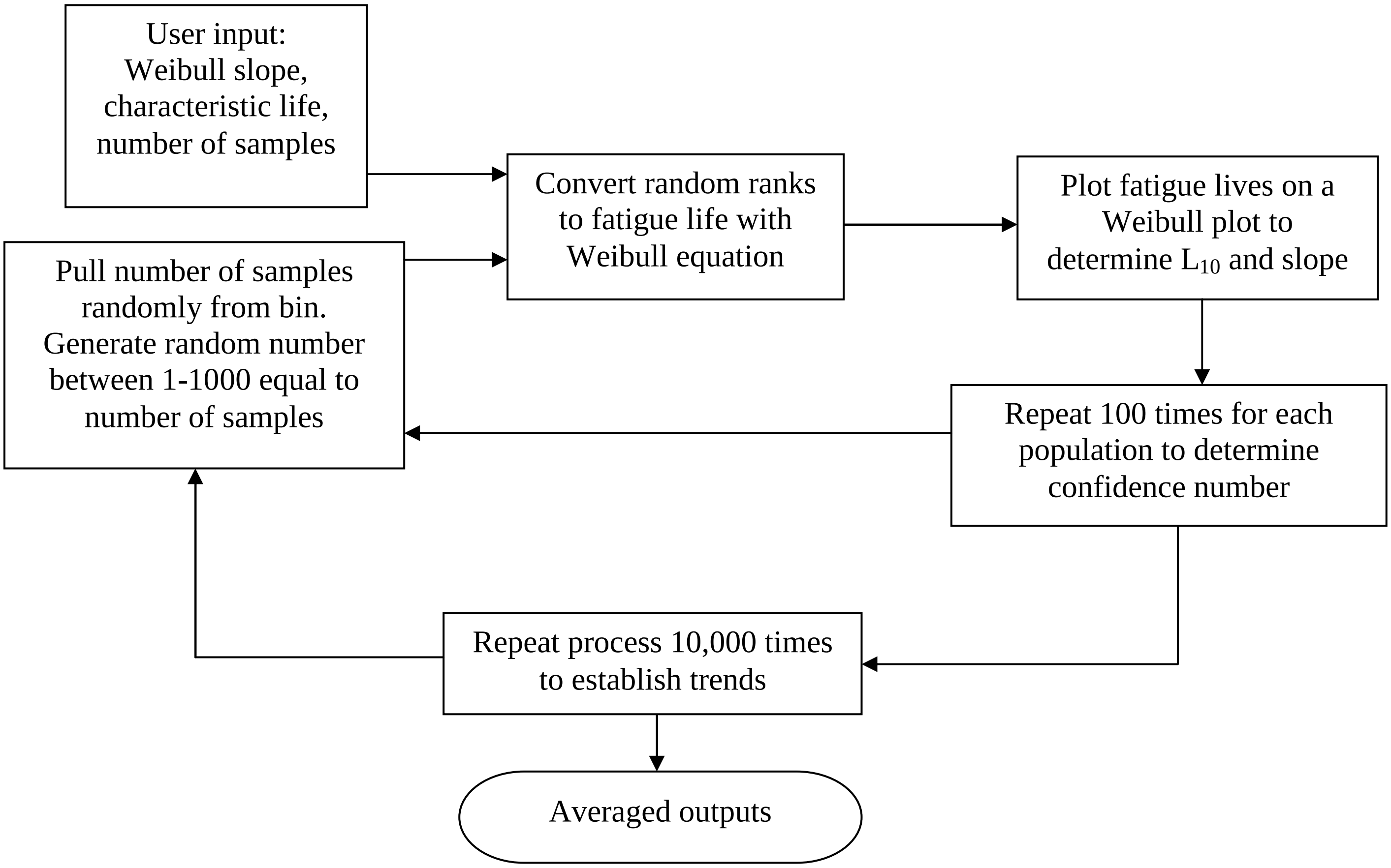


Figure 11. Flowchart of Monte Carlo simulation based on bin model

Numerically Counting Confidence Numbers

A confidence number determined by the simulation (numerical counting method) was accomplished by generating 100 L_{10} lives for two different simulated materials and counting the number of times the L_{10} life associated with one population was greater than the other out of 100 comparisons. Since the program generates virtual L_{10} lives these lives were counted to see how many times population A has a longer life than population B. The number of times out of 100 that A is better than B and the number of times out of 100 that B is better than A was outputted to the summary table. The greater of the two numbers is the confidence number. If this number is greater than 90 then it states that there is a significant difference between the two populations. A flowchart of the simulation determining a confidence number by the numerical counting method is shown in figure 12.

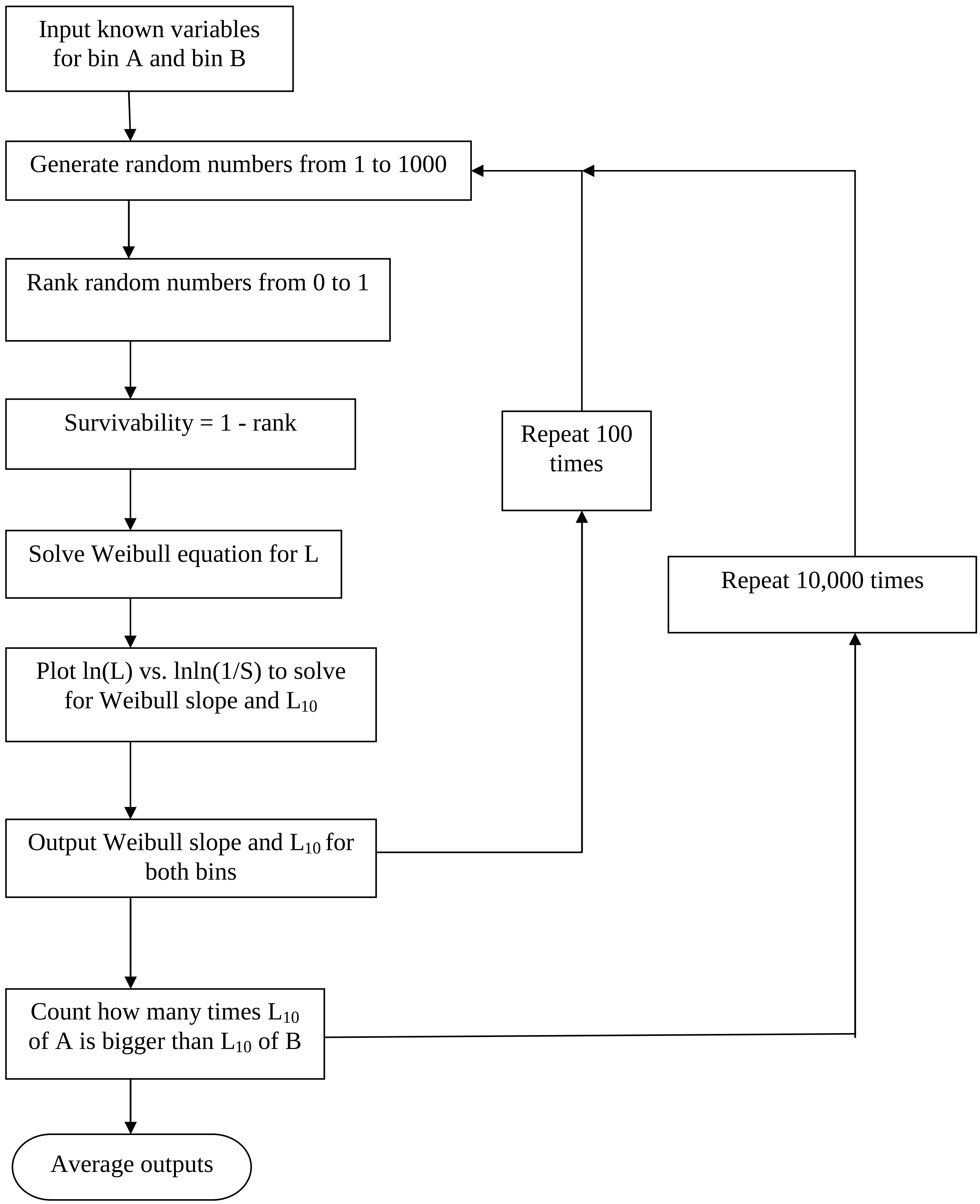


Figure 12. Flowchart of Monte Carlo simulation counting method.

It was determined which was greater by dividing the L_{10} life of bin A by the L_{10} life of bin B and then counting the number of times the result was greater than or less than one. If the result is greater than one then L_{10} of A was greater, and vice versa for B. This counting method is shown in figure 13.

```
olumn)
```

Figure 13. Subroutine for numerically counting a confidence number

A screen shot of the output of the numerical counting method is shown in figure 14.

150			
151	L10A/L10B	0.32	0.43
152			
153	L10A>L10B	7	B>A
154	L10B>L10A	93	

Figure 14. Screen shot of output of counting confidence numbers

This confidence number model works well for experimental data in which all fatigue specimens in the original experiment failed. However for the purpose of this research a new model was developed to use this Monte Carlo simulation to generate fatigue data that has suspensions in it and then determine confidence numbers. There are five new models presented.

Five Suspension Models

In this study, five different out-life suspension models were created and compared, to determine which best simulated fatigue data sets, with out-life suspensions, was better. There are five versions of the program; each simulates a different suspension model. The five different methods were designed to represent possible outcomes of the actual experimental since the details of why the suspensions occurred are not known.

Simulation Method 1: No out-life suspensions

Method 1 is a simulation of what confidence number would have been determined had all the specimens failed—i.e. there were no suspensions. For example, if 20 samples of material A were to be tested against 20 samples of material B and only 6 samples of material B failed for whatever the reason, the following would be conducted. 20 lives of A and 20 lives of B would be generated. All of these lives would then be used to determine a confidence number and a base line. This is theoretically what could have happened had all the specimens in the experiment failed. This serves as a baseline for comparison. While the method for determining the individual fatigue lives and the L_{10} lives was demonstrated by Vlcek, Hendricks and Zaretsky (Vlcek, Hendricks and Zaretsky, Determination of Rolling-Element Fatigue Life From Computer Generated Bearing Tests 2003) and McBride (McBride 2011), the Visual Basic programming solution to this simulation was unique to this study. The full Visual Basic program for simulating Method 1 can be found in Appendix A.

A screen shot of the inputs for method one can be seen in figure 15.

	A	B
1		
2		
3	A m_A	2.2
4	B m_B	1.4
5		
6	$L_{B,A}$	175,220,000
7	$L_{B,B}$	698,580,000
8		
9	size A	20
10	size B	20
11		
12	Trials	100
13	Trials*	2000
14		

Figure 15. Sample of the inputs for method 1

In this figure, A m_A is the slope of the experimental data set for material A and B m_B is the slope of the experimental data set for material B. $L_{B,A}$ is the characteristic life for the experimental data for material A and $L_{B,B}$ is the characteristic life for the experimental data for material B. Size A and size B are the sizes of the populations for materials A and B. Trials is the number of times to run the loop to calculate a confidence number. Remember the simulation runs 100 times to count how many times (out of 100) one material is better than the other. Trials* is the number of times to run the entire program to establish trends. All five methods were run a total of 10,000 times.

In this figure, and in most of the simulations, the program was broken up into five blocks of 2,000 because of the lack of RAM on the computers used.

The part of the simulation that changes from method to method is how the slopes are calculated. After the virtual lives have been generated and a new survivability has been calculated then virtually plotting the lnln(1/S) against the ln(L) will give the Weibull slope of the simulated fatigue data. The code to calculate the slope is in figure 16.

```
'slope A
MA = Application.WorksheetFunction.LinEst(Range(Cells(7, 10),
Cells(alpha + 6, 10)), Range(Cells(7, 11), Cells(alpha + 6,
11)), True, True)
Cells(16, 2) = MA
MAa = Cells(16, 2)
Slope1A = MAa

'slope B
MB = Application.WorksheetFunction.LinEst(Range(Cells(7, 18),
Cells(alpha2 + 6, 18)), Range(Cells(7, 19), Cells(alpha2 + 6,
19)), True, True)
Cells(24, 2) = MB
MBb = Cells(24, 2)
Slope1B = MBb
```

Figure 16. Visual Basic code for calculating the Weibull slope of bins A and B

Figure 17 is a screen shot of the outputs. The shaded numbers are the numbers used to calculate the slope of the virtual data.

1	Bin A	Rank A	S_A	L_A	New Rank A	S_A new	lnln(1/S)$_A$	ln(L)$_A$	Bin B	Rank B	S_B	L_B	lew Rank I	S_B new	lnln(1/S)$_B$	ln(L)$_B$
1	5	0.00	1.00	15,340,235	0.03	0.97	-3.35	16.55	27	0.03	0.97	53,010,024	0.03	0.97	-3.35	17.79
2	71	0.07	0.93	53,419,972	0.08	0.92	-2.44	17.79	35	0.03	0.97	64,110,578	0.08	0.92	-2.44	17.98
3	121	0.12	0.88	68,968,962	0.13	0.87	-1.95	18.05	107	0.11	0.89	146,978,212	0.13	0.87	-1.95	18.81
4	182	0.18	0.82	84,392,019	0.18	0.82	-1.61	18.25	138	0.14	0.86	178,538,032	0.18	0.82	-1.61	19.00
5	270	0.27	0.73	103,516,841	0.23	0.77	-1.34	18.46	143	0.14	0.86	183,516,002	0.23	0.77	-1.34	19.03
6	337	0.34	0.66	116,877,991	0.28	0.72	-1.12	18.58	145	0.14	0.86	185,500,133	0.28	0.72	-1.12	19.04
7	407	0.41	0.59	130,367,480	0.33	0.67	-0.92	18.69	187	0.19	0.81	226,448,076	0.33	0.67	-0.92	19.24
8	445	0.44	0.56	137,635,527	0.38	0.62	-0.75	18.74	272	0.27	0.73	307,415,141	0.38	0.62	-0.75	19.54
9	605	0.60	0.40	169,326,964	0.43	0.57	-0.59	18.95	309	0.31	0.69	342,729,647	0.43	0.57	-0.59	19.65
10	625	0.62	0.38	173,567,107	0.48	0.52	-0.44	18.97	313	0.31	0.69	346,568,710	0.48	0.52	-0.44	19.66
11	701	0.70	0.30	190,746,731	0.52	0.48	-0.30	19.07	325	0.32	0.68	358,118,969	0.52	0.48	-0.30	19.70
12	708	0.71	0.29	192,437,720	0.57	0.43	-0.16	19.08	365	0.36	0.64	397,068,343	0.57	0.43	-0.16	19.80
13	722	0.72	0.28	195,888,873	0.62	0.38	-0.03	19.09	394	0.39	0.61	425,862,734	0.62	0.38	-0.03	19.87
14	777	0.78	0.22	210,543,538	0.67	0.33	0.11	19.17	469	0.47	0.53	503,394,419	0.67	0.33	0.11	20.04
15	804	0.80	0.20	218,577,410	0.72	0.28	0.24	19.20	544	0.54	0.46	587,189,733	0.72	0.28	0.24	20.19
16	830	0.83	0.17	227,040,457	0.77	0.23	0.38	19.24	599	0.60	0.40	654,325,646	0.77	0.23	0.38	20.30
17	875	0.87	0.13	244,124,925	0.82	0.18	0.53	19.31	601	0.60	0.40	656,880,287	0.82	0.18	0.53	20.30
18	880	0.88	0.12	246,284,881	0.87	0.13	0.70	19.32	686	0.69	0.31	775,049,907	0.87	0.13	0.70	20.47
19	890	0.89	0.11	250,812,846	0.92	0.08	0.91	19.34	760	0.76	0.24	899,492,732	0.92	0.08	0.91	20.62
20	951	0.95	0.05	288,800,815	0.97	0.03	1.22	19.48	870	0.87	0.13	1,160,533,004	0.97	0.03	1.22	20.87

Figure 17. Screen shot of method 1 outputs

Using this new Weibull slope, and the virtual data, a characteristic life for this data set can be calculated. Using the new Weibull slope and characteristic life in the Weibull equation, with S set as 0.9, a L_{10} life for this virtual data set can be calculated. This is done for both bins. These L_{10} lives are then counted to see how many times out of 100 the L_{10} life of one population is better than the other population.

Simulation Method 2: Failure Index Forced to Match a Known Failure Index

Method 2 simulates the L_{10} lives that would be obtained if all the specimens that failed in the original experiment were equal to the total number tested. For example, if there were 20 samples of material A and 20 samples of material B to be tested, and 20 of material A failed, yet only 6 of material B failed, then 20 lives for material A and 6 lives for material B were generated. This method again does not directly incorporate suspensions because it assumes all samples failed. While the reduced population size is used, it is still the first 6 failures that were randomly generated. The full Visual Basic program for simulating Method 2 can be found in Appendix B.

A screen shot of the inputs for method 2 can be seen in figure 18.

	A	B
1		
2		
3	A m_A	2.2
4	B m_B	1.4
5		
6	$L_{B,A}$	175,220,000
7	$L_{B,B}$	698,580,000
8		
9	size A	20
10	size B	6
11		
12	Trials	100
13	Trials*	2000

Figure 18. Sample of the inputs for method 2

The slopes and characteristics lives and trial numbers do not change. The only change is the amount of lives generated determined by the size of the populations entered. In figure 18 it can be seen that the population of B is lower than in method 1.

1	Bin A	Rank A	S_A	L_A	New Rank A	S_A new	lnln(1/S)$_A$	ln(L)$_A$	Bin B	Rank B	S_B	L_B	lew Rank I	S_B new	lnln(1/S)$_B$	ln(L)$_B$
1	49	0.05	0.95	44,857,216	0.03	0.97	-3.35	17.62	116	0.12	0.88	156,283,349	0.11	0.89	-2.16	18.87
2	60	0.06	0.94	49,336,869	0.08	0.92	-2.44	17.71	307	0.31	0.69	340,812,013	0.27	0.73	-1.18	19.65
3	78	0.08	0.92	55,856,941	0.13	0.87	-1.95	17.84	716	0.72	0.28	822,433,162	0.42	0.58	-0.60	20.53
4	127	0.13	0.87	70,614,082	0.18	0.82	-1.61	18.07	872	0.87	0.13	1,166,806,959	0.58	0.42	-0.15	20.88
5	136	0.14	0.86	73,019,074	0.23	0.77	-1.34	18.11	954	0.95	0.05	1,554,557,802	0.73	0.27	0.28	21.16
6	255	0.25	0.75	100,417,734	0.28	0.72	-1.12	18.42	977	0.98	0.02	1,793,215,037	0.89	0.11	0.79	21.31
7	328	0.33	0.67	115,118,196	0.33	0.67	-0.92	18.56								
8	351	0.35	0.65	119,600,211	0.38	0.62	-0.75	18.60								
9	421	0.42	0.58	133,044,585	0.43	0.57	-0.59	18.71								
10	495	0.49	0.51	147,262,563	0.48	0.52	-0.44	18.81								
11	526	0.53	0.47	153,319,702	0.52	0.48	-0.30	18.85								
12	586	0.59	0.41	165,380,019	0.57	0.43	-0.16	18.92								
13	613	0.61	0.39	171,011,789	0.62	0.38	-0.03	18.96								
14	725	0.72	0.28	196,641,157	0.67	0.33	0.11	19.10								
15	780	0.78	0.22	211,404,078	0.72	0.28	0.24	19.17								
16	799	0.80	0.20	217,037,564	0.77	0.23	0.38	19.20								
17	833	0.83	0.17	228,072,474	0.82	0.18	0.53	19.25								
18	873	0.87	0.13	243,278,685	0.87	0.13	0.70	19.31								
19	968	0.97	0.03	306,455,796	0.92	0.08	0.91	19.54								
20	975	0.97	0.03	316,083,032	0.97	0.03	1.22	19.57								

Figure 19. Screen shot of method 2 outputs

In figure 19 are the outputs for method 2 and the shaded areas are the numbers used to calculate the new Weibull slope. Using this Weibull slope an L_{10} life can be calculated for each population.

Simulation Method 3: Specified Failure Index for Cut-Off Out-Life

The third method generates random lives for *all* specimens yet it only uses the number of actual failed specimens to calculate the confidence number. For example, if there were 20 specimens of material A and 20 specimens of material B, yet only 6 of B failed, the program would generate 20 lives for A and 20 lives for B and then use all 20 lives for the calculations of A and only the shortest lived 6 lives for B. This represents what would happen if it was assumed that the 14 longest lived samples in population B were suspended. The way this was accomplished in the actual program is all 20 lives for bins A and B were generated as normal to get a full distribution. When the values were calculated for the Weibull slope of bin B, only the first 6 lives were picked for the calculation. The following calculations for L_{10}, L_{50}, characteristic life, and mean were then based on only those first 6 samples. The full Visual Basic program for simulating Method 3 can be found in Appendix C.

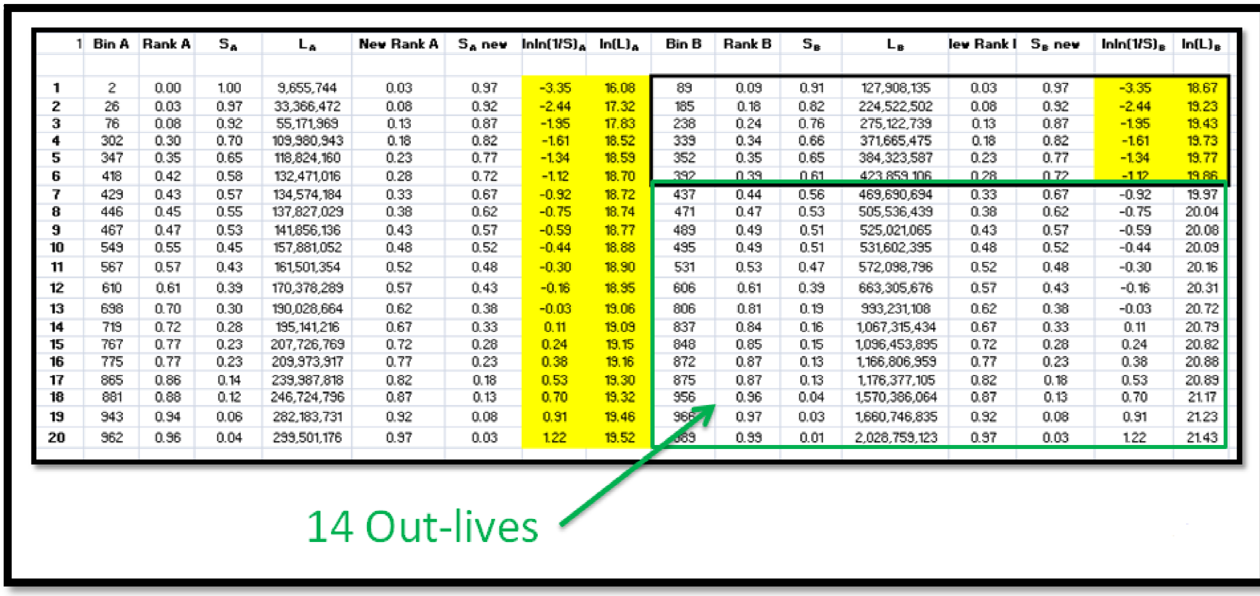
A screen shot for method 3 can be seen in figure 20.

	A	B
1	**1**	**2**
2		
3	**A m_A**	2.2
4	**B m_B**	1.4
5		
6	**$L_{B,A}$**	175,220,000
7	**$L_{B,B}$**	698,580,000
8		
9	**size A**	20
10	**size B**	20
11	**cut off B**	6
12	**Trials**	100
13	**Trials***	2000

Figure 20. Sample of the inputs for method 3

The slopes, characteristic lives, and trials again do not change. This time the sizes of the populations are set to the total number of samples. A cut-off is specified to tell the simulation to only use the first 6 generated lives in the calculations of the Weibull slopes and L_{10} lives.

In figure 21 is a screen shot of the outputs of method 3.



1	Bin A	Rank A	S_A	L_A	New Rank A	S_A new	$\ln\ln(1/S)_A$	$\ln(L)_A$	Bin B	Rank B	S_B	L_B	lew Rank I	S_B new	$\ln\ln(1/S)_B$	$\ln(L)_B$
1	2	0.00	1.00	9,655,744	0.03	0.97	-3.35	16.08	89	0.09	0.91	127,908,135	0.03	0.97	-3.35	18.67
2	26	0.03	0.97	33,366,472	0.08	0.92	-2.44	17.32	165	0.16	0.82	224,522,502	0.08	0.92	-2.44	19.23
3	76	0.08	0.92	55,171,969	0.13	0.87	-1.95	17.83	238	0.24	0.76	275,122,739	0.13	0.87	-1.95	19.43
4	302	0.30	0.70	109,980,943	0.18	0.82	-1.61	18.52	339	0.34	0.66	371,665,475	0.18	0.82	-1.61	19.73
5	347	0.35	0.65	118,824,160	0.23	0.77	-1.34	18.59	352	0.35	0.65	384,323,587	0.23	0.77	-1.34	19.77
6	418	0.42	0.58	132,471,016	0.28	0.72	-1.12	18.70	392	0.39	0.61	423,859,106	0.28	0.72	-1.12	19.86
7	429	0.43	0.57	134,574,184	0.33	0.67	-0.92	18.72	437	0.44	0.56	469,690,694	0.33	0.67	-0.92	19.97
8	446	0.45	0.55	137,827,029	0.38	0.62	-0.75	18.74	471	0.47	0.53	505,536,439	0.38	0.62	-0.75	20.04
9	467	0.47	0.53	141,856,136	0.43	0.57	-0.59	18.77	489	0.49	0.51	525,021,065	0.43	0.57	-0.59	20.08
10	549	0.55	0.45	157,881,052	0.48	0.52	-0.44	18.88	495	0.49	0.51	531,602,395	0.48	0.52	-0.44	20.09
11	567	0.57	0.43	161,501,354	0.52	0.48	-0.30	18.90	531	0.53	0.47	572,098,796	0.52	0.48	-0.30	20.16
12	610	0.61	0.39	170,378,289	0.57	0.43	-0.16	18.95	606	0.61	0.39	663,305,676	0.57	0.43	-0.16	20.31
13	698	0.70	0.30	190,028,664	0.62	0.38	-0.03	19.06	806	0.81	0.19	993,231,108	0.62	0.38	-0.03	20.72
14	719	0.72	0.28	195,141,216	0.67	0.33	0.11	19.09	837	0.84	0.16	1,067,315,434	0.67	0.33	0.11	20.79
15	767	0.77	0.23	207,726,769	0.72	0.28	0.24	19.15	848	0.85	0.15	1,096,453,895	0.72	0.28	0.24	20.82
16	775	0.77	0.23	209,973,917	0.77	0.23	0.38	19.16	872	0.87	0.13	1,166,806,959	0.77	0.23	0.38	20.88
17	865	0.86	0.14	239,987,818	0.82	0.18	0.53	19.30	875	0.87	0.13	1,176,377,105	0.82	0.18	0.53	20.89
18	881	0.88	0.12	246,724,796	0.87	0.13	0.70	19.32	956	0.96	0.04	1,570,386,064	0.87	0.13	0.70	21.17
19	943	0.94	0.06	282,183,731	0.92	0.08	0.91	19.46	966	0.97	0.03	1,660,746,835	0.92	0.08	0.91	21.23
20	962	0.96	0.04	299,501,176	0.97	0.03	1.22	19.52	969	0.99	0.01	2,028,759,123	0.97	0.03	1.22	21.43

Figure 21. Screen shot of method 3 outputs

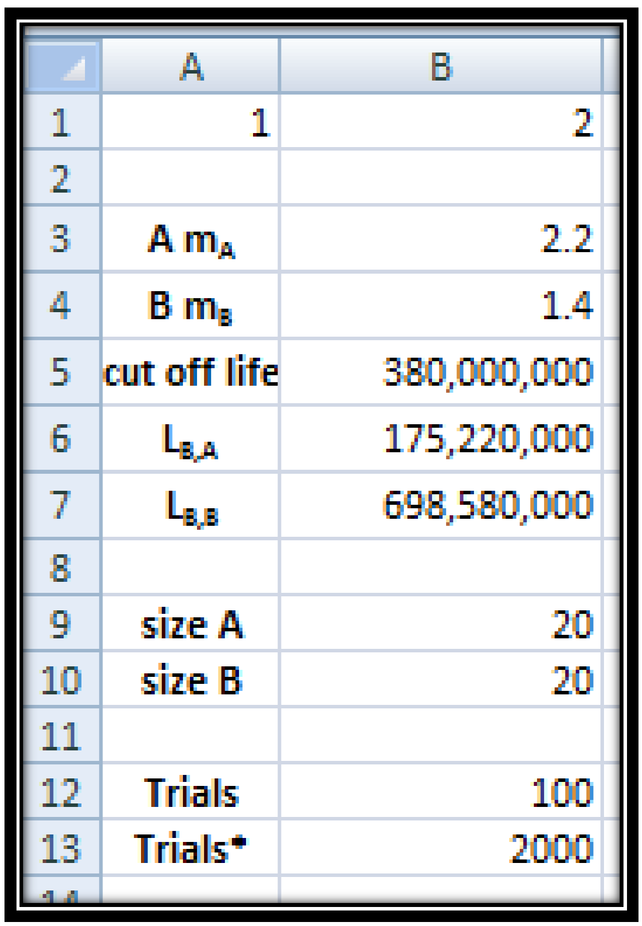
In figure 21 it shows that 20 lives were generated for both populations, but only the shortest lived 6 lives of bin B are used in the calculation of a Weibull slope and L_{10} life. The longest lived 14 are the out-life suspensions.

Simulation Method 4: Specified Cut-Off Out-Life

In method 4 the total number of specimens attempted is generated but then only the ones that reach a specified cut-off life are used for calculated the L_{10} life. For example, bin A has 20 samples and bin B has 20 samples. The experimenter determines that there is no need for a specimen to run more than 380 million cycles. If the longest life of bin A was 255,900,000 then all of bin A would get used in the calculations for a confidence number. In bin B, however, the seventh longest life out of the 20 was 390,000,000 then only the shortest six lives would get used in calculating the L_{10} life.

As in the original program the user inputs the size of bin A and bin B. The program goes through and generates 20 lives for bin A. When calculating the lives of bin B, the program begins by generating 20 random numbers as usual, however, when it begins to convert the random numbers into lives, when the program computes a life greater than the designated life it ends the loop and jumps to the next command. Since the program has to go over the designated life before it can terminate, it then needs to be told to only use the lives previous to the last one generated. The program copies the usable data (that is the lives generated minus the last one) to open cells to the right of the original numbers in the spreadsheet and then performs calculations using this copied data to ensure the correct numbers are used. The full Visual Basic program for simulating Method 4 can be found in Appendix D.

In figure 22 is a screen shot of the inputs for method 4.



	A	B
1	1	2
2		
3	A m_A	2.2
4	B m_B	1.4
5	cut off life	380,000,000
6	$L_{B,A}$	175,220,000
7	$L_{B,B}$	698,580,000
8		
9	size A	20
10	size B	20
11		
12	Trials	100
13	Trials*	2000

Figure 22. Sample of the inputs for method 4

The Weibull slopes, characteristic lives and trials are entered the same as before. The sizes of the populations are set as the total number attempted. This time the cut-off that is entered is a specific life. In this particular case the cut-off life for bin B is 380 million. In figure 23 is a screen shot of the outputs of method 4.

	Bin A	Rank A	S_A	L_A	Jew Rank	S_A new	$\ln\ln(1/S)_A$	$\ln(L)_A$	Bin B	Rank B	S_B	L_B	New Rank B	S_B new	$\ln\ln(1/S)_B$	$\ln(L)_B$
1	92	0.09	0.91	60,432,110	0.03	0.97	-3.35	17.92	11	0.01	0.99	27,426,821	0.03	0.97	-3.35	17.13
2	109	0.11	0.89	65,564,964	0.08	0.92	-2.44	18.00	23	0.02	0.98	47,138,750	0.08	0.92	-2.44	17.67
3	233	0.23	0.77	95,772,536	0.13	0.87	-1.95	18.38	148	0.15	0.85	188,469,169	0.13	0.87	-1.95	19.05
4	247	0.25	0.75	98,743,341	0.18	0.82	-1.61	18.41	155	0.15	0.85	195,365,609	0.18	0.82	-1.61	19.09
5	307	0.31	0.69	110,975,812	0.23	0.77	-1.34	18.52	173	0.17	0.83	212,926,630	0.23	0.77	-1.34	19.18
6	309	0.31	0.69	111,372,767	0.28	0.72	-1.12	18.53	186	0.19	0.81	225,485,520	0.28	0.72	-1.12	19.23
7	323	0.32	0.68	114,136,778	0.33	0.67	-0.92	18.55	205	0.20	0.80	243,705,293	0.33	0.67	-0.92	19.31
8	382	0.38	0.62	125,577,869	0.38	0.62	-0.75	18.65	246	0.25	0.75	282,719,164	0.38	0.62	-0.75	19.46
9	408	0.41	0.59	130,558,762	0.43	0.57	-0.59	18.69	263	0.26	0.74	298,861,575	0.43	0.57	-0.59	19.52
10	446	0.45	0.55	137,827,029	0.48	0.52	-0.44	18.74	325	0.32	0.68	358,118,969	0.48	0.52	-0.44	19.70
11	471	0.47	0.53	142,625,670	0.52	0.48	-0.30	18.78	402	0.40	0.60	433,905,966	0.52	0.48	-0.30	19.89
12	475	0.47	0.53	143,396,029	0.57	0.43	-0.16	18.78	429							
13	539	0.54	0.46	155,889,700	0.62	0.38	-0.03	18.86	433							
14	591	0.59	0.41	166,411,654	0.67	0.33	0.11	18.93	538							
15	681	0.68	0.32	186,030,083	0.72	0.28	0.24	19.04	562							
16	713	0.71	0.29	193,659,364	0.77	0.23	0.38	19.08	705							
17	758	0.76	0.24	205,255,434	0.82	0.18	0.53	19.14	830							
18	909	0.91	0.09	260,346,888	0.87	0.13	0.70	19.38	892							
19	924	0.92	0.08	269,018,852	0.92	0.08	0.91	19.41	921							
20	943	0.94	0.06	282,183,731	0.97	0.03	1.22	19.46	955							

10 Out-lives

Figure 23. Screen shot of method 4 outputs

From figure 23 it can be seen that 20 lives are generated for bin A. For bin B however all 20 random numbers are generated, but when the random numbers started getting calculated into lives the program stops when it generates a life over the specified life. The program then knows to use one minus the generated lives in the calculation of the Weibull slopes and L_{10} life. This is shown by the shaded numbers in figure 23.

Simulation Method 5: Specified Cut-Off Out-Life and Failure Index

Method 5 is a hybrid of methods 3 and 4. Method 5 operates the same way except that it forces the bin to use the specified predetermined number of samples failed or failure index *and* a specified life. To relate to the previous examples, it would force bin B to have 6 failed samples between 1 and 380 million cycles. The user inputs the number of lives to be generated; in this case the numbers are 20 for bin A and 6 for bin B. The program will only generate 20 lives for bin A and 6 lives for bin B. The program generates the lives for bin A as usual; however, when the lives are being calculated for bin B the same procedure as method 4 was used. This time, however, if the lives get cut below the specified number (i.e. 6) then the loop starts over. For example, if only 4 lives are generated because the fourth life has reached the limit then the loop will start over until there are 6 lives within the specified life range. The full Visual Basic program for simulating Method 5 can be found in Appendix E.

In figure 24 is a screen shot of the inputs for method 5.

	A	B
1	1	2
2		
3	A m_A	2.2
4	B m_B	1.4
5	cut off life	380,000,000
6	$L_{B,A}$	175,220,000
7	$L_{B,B}$	698,580,000
8		
9	size A	20
10	size B	6
11		
12	Trials	100
13	Trials*	2000

Figure 24. Sample of the inputs for method 5

The Weibull slopes, characteristic lives, and trial numbers are again still the same. The forced failure index is designated by the size of the bins and the cut-off life is also specified, in this case the cut-off life is 380 million. In figure 25 is a screen shot of the outputs for method 5.

1	Bin A	Rank A	S_A	L_A	Jew Rank	S_A new	$\ln\ln(1/S)_A$	$\ln(L)_A$	Bin B	Rank B	S_B	L_B	Jew Rank	S_B new	$\ln\ln(1/S)_B$	$\ln(L)_B$
1	7	0.01	0.99	18,031,000	0.03	0.97	-3.35	16.71	74	0.07	0.93	111,412,540	0.11	0.89	-2.16	18.53
2	55	0.05	0.95	47,357,271	0.08	0.92	-2.44	17.67	135	0.13	0.87	175,538,480	0.27	0.73	-1.18	18.98
3	172	0.17	0.83	82,028,927	0.13	0.87	-1.95	18.22	185	0.18	0.82	224,522,502	0.42	0.58	-0.60	19.23
4	192	0.19	0.81	86,705,903	0.18	0.82	-1.61	18.28	207	0.21	0.79	245,615,922	0.58	0.42	-0.15	19.32
5	215	0.21	0.79	91,866,459	0.23	0.77	-1.34	18.34	245	0.24	0.76	281,769,735	0.73	0.27	0.28	19.46
6	267	0.27	0.73	102,900,965	0.28	0.72	-1.12	18.45	333	0.33	0.67	365,849,697	0.89	0.11	0.79	19.72
7	343	0.34	0.66	118,046,770	0.33	0.67	-0.92	18.59								
8	371	0.37	0.63	123,463,157	0.38	0.62	-0.75	18.63								
9	401	0.40	0.60	129,219,441	0.43	0.57	-0.59	18.68								
10	426	0.43	0.57	134,000,550	0.48	0.52	-0.44	18.71								
11	625	0.62	0.38	173,567,107	0.52	0.48	-0.30	18.97								
12	646	0.65	0.35	178,128,550	0.57	0.43	-0.16	19.00								
13	647	0.65	0.35	178,348,848	0.62	0.38	-0.03	19.00								
14	668	0.67	0.33	183,046,949	0.67	0.33	0.11	19.03								
15	674	0.67	0.33	184,416,206	0.72	0.28	0.24	19.03								
16	679	0.68	0.32	185,567,123	0.77	0.23	0.38	19.04								
17	771	0.77	0.23	208,844,186	0.82	0.18	0.53	19.16								
18	868	0.87	0.13	241,204,848	0.87	0.13	0.70	19.30								
19	891	0.89	0.11	251,282,281	0.92	0.08	0.91	19.34								
20	917	0.92	0.08	264,821,238	0.97	0.03	1.22	19.39								

Figure 25. Screen shot of method 5 outputs

From figure 25 it can be seen that the program generates 20 lives for bin A and forces bin B to have 6 lives that fall between 1 and 380 million. Again the shaded numbers are used in calculating the Weibull slope and L_{10} lives.

All of these simulations were run using real data from past experiments in the literature (Townsend, Zaretsky and Anderson, Comparison of Modified Vasco X-2 with AISI 9310 - Preliminary Report 1977), (Townsend and Zaretsky, Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels 1980) as inputs to the Monte Carlo simulation and to validate the model. The data came from an experiment where a comparison of helicopter gear materials was conducted. The helicopter gears were of particular interest because in the original experiment there were suspensions and a graphically determined confidence number using Johnson's (Johnson, The Statistical Treatment of Fatigue Experiments 1964) method.

The data from the original experiments was run through each of the five methods for 10,000 cycles for statistical accuracy. The inputs of the simulation can be seen in tables 5 and 6.

Table 5. Input parameters used for rolling contact data set simulation			
	Weibull Slope	Characteristic Life	Failure Index
Modified Vasco X-2	2.2	175,220,000	20 out of 20
AISI 9310	1.4	698,580,000	6 out of 20

Table 6. Input parameters used for gear fatigue data set simulation				
		Weibull Slope	Characteristic Life	Failure Index
AISI 9310	----	2.3	61,190,000	30 out of 30
Modified Vasco X-2	Boeing Vertol	1.0	364,460,000	12 out of 26
	NASA	0.53	55,859,000	18 out of 21
	Curtis-Wright	2.1	9,636,000	19 out of 19

Algebraic Approximation of Johnson's Confidence Numbers

There was another method of calculating confidence numbers that was incorporated into the simulation. This method was developed by Vlcek et al. (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007). In the method a number of equations were developed to algebraically solve for a confidence number. This method eliminated the need for graphical interpretation of the results, and was incorporated into the simulation to give another confidence number to compare the Monte Carlo counting method to.

Equations 9 through 15 are calculated to determine a confidence number using the mean lives. To begin, the degrees of freedom (DOF) must be found. The degrees of freedom represents the number of times a part can be randomly chosen. If there are 20 objects, you can randomly choose 19 because the 20th item is fixed and not random. Degrees of freedom is defined as

Equation (9)

$$DOF = n - 1$$

In the case of two bins the degrees of freedom will be

Equation (10)

$$DOF = (n_1 - 1) \times (n_2 - 1)$$

The mean life ratio (MLR) at 99%, is determined by

Equation (11)

$$MLR_{@99\%} = (A_o \ln(DOF) + B_o)^2 + 1$$

Where

Equation (12)

$$A_o = \frac{-0.0844}{m} - 0.05584$$

And

Equation (13)

$$B_o = \frac{1.2796}{m} + 0.6729$$

Once the mean life ratio at a confidence number of 99-percent has been established it can then be used to find D.

Equation (14)

$$D = \frac{3.912}{MLR_{@99\%} - 1}$$

Next the mean life ratio was determined for the experimental values, the bigger mean life divided by the smaller mean life. Once these variables have been calculated, a confidence number can be determined using equation 15:

Equation (15)

$$C = 1 - 0.5e^{\left(-D\left(MLR_{exp}-1\right)\right)}$$

Another method to determining a confidence number is to use the L_{10} life dependent equations. These begin with:

Equation (16)

$$A = \exp\left(\frac{a_2}{m} + b_2\right)$$

Equation (17)

$$B = a_1 \ln(m) + b_1$$

Where $a_1 = 0.29574$, $a_2 = 4.5286$, $b_1 = -0.45228$, and $b_2 = 0.3152$

Once A and B have been calculated the fitted L_{10} life ratio must be determined:

Equation (18)

$$\text{Fitted } L_{10} \text{ life ratio} = (A)\, DOF^{B}$$

Followed by:

Equation (19)

$$a = \frac{a_0}{\ln(L_{10}\text{Life Ratio})}$$

where

Equation (20)

$$a_o = (E_2 \, Ln(1-0.99) - E_1)^{0.5}$$

where $E_1 = 2896.3$ and $E_2 = -3595.9$

The confidence number based on L_{10} lives can now be calculated.

Equation (21)

$$C_{L10} = 1 - \exp\{[(a \ln(x_o))^2 + E_1]/E_2\}$$

These two methods make it possible to calculate a confidence number of an experiment without the need to use Johnson's figures (Johnson, The Statistical Treatment of Fatigue Experiments 1964). They were incorporated into the simulation and used the lives of the generated random numbers to calculate confidence numbers.

The program code was then modified to determine confidence numbers utilizing the equations proposed by Vlcek, Zaretsky, and Hendricks. These values were then compared to the confidence numbers generated by the Monte Carlo counting method. There are two methods of curve fit equations. The first uses the mean life ratio to calculate a confidence number and the second uses the L_{10} life ratio to calculate a confidence number at that respective probability of failure. The degrees of freedom for the bins was first calculated. The first confidence number calculated is the mean life ratio confidence number. The confidence number based on the Weibull slope for bin A was calculated and then the confidence number based on the Weibull slope for bin B was calculated. The average of these two is then the mean life ratio confidence number. The code for determining a confidence number based on mean life ratio is shown in figure 26.

```
'CONFIDENCE INTERAL FOR A

    'Ao
    Anot = (-0.0844 / Cells(16, 2)) - 0.05584
    'Bo
    Bnot = (1.2796 / Cells(16, 2)) + 0.6729
    'lnDOF
    lnDOF =
Application.WorksheetFunction.Ln(Cells(40, 2))
    'MLR at 99
    MLR99 = (Anot * lnDOF + Bnot) ^ 2 + 1
    Cells(41, 2) = MLR99
    'D
    Dvegas = 3.912 / (MLR99 - 1)
    'MLRexp
    If Cells(21, 2) > Cells(29, 2) Then
    MLRexp = Cells(21, 2) / Cells(29, 2)
    Else
    MLRexp = Cells(29, 2) / Cells(21, 2)
    End If
    'C
    Cvegas = 1 - 0.5 * Exp(-Dvegas * (MLRexp -
1))
    Cells(42, 2) = Cvegas
```

Figure 26. Subroutine for calculating confidence number based on mean life ratio

The L_{10} ratio confidence numbers were calculated next. Again the confidence number based on the Weibull slope for bin A was calculated followed by the confidence number based on the Weibull slope for bin B, then the average of those two values was calculated do determine a final confidence number. The code for determining a confidence number based on L_{10} lives is shown in figure 27.

```
'L10 dependent confidence numbers

'A
AAA = Exp((4.5286 / Cells(16, 2)) + 0.3152)
'ln(m)
lnmA = Application.WorksheetFunction.Ln(Cells(16,
2))
'B
BBB = 0.29574 * lnmA + (-0.45228)
'L10LR
L10LRA = AAA * Cells(40, 2) ^ BBB
'ao
litanot = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRA = Application.WorksheetFunction.Ln(L10LRA)
'a
lita = litanot / lnL10LRA
    'L10exp
    If Cells(18, 2) > Cells(26, 2) Then
    xnot = Cells(18, 2) / Cells(26, 2)
    Else
    xnot = Cells(26, 2) / Cells(18, 2)
    End If
'ln(xo)
lnxo = Application.WorksheetFunction.Ln(xnot)
'CL10
CL10 = 1 - Exp(((lita * lnxo) ^ 2 + 2896.3) / -
3595.9)
Cells(46, 2) = CL10
```

Figure 27. Subroutine for calculating confidence number based on L_{10} life ratio

After the calculations were performed the Weibull slopes, L_{10} lives, L_{50} lives, characteristic lives, mean lives, degrees of freedom, mean life ratio confidence numbers and L_{10} life ratio confidence numbers were outputted again to the first worksheet on the spreadsheet (figure 28) where they were then copied from sheet one to the “Summary” sheet. The summary sheet is used to store all the numbers until the program is done running the desired number of trials. The final confidence numbers came from averaging the numbers on the summary page.

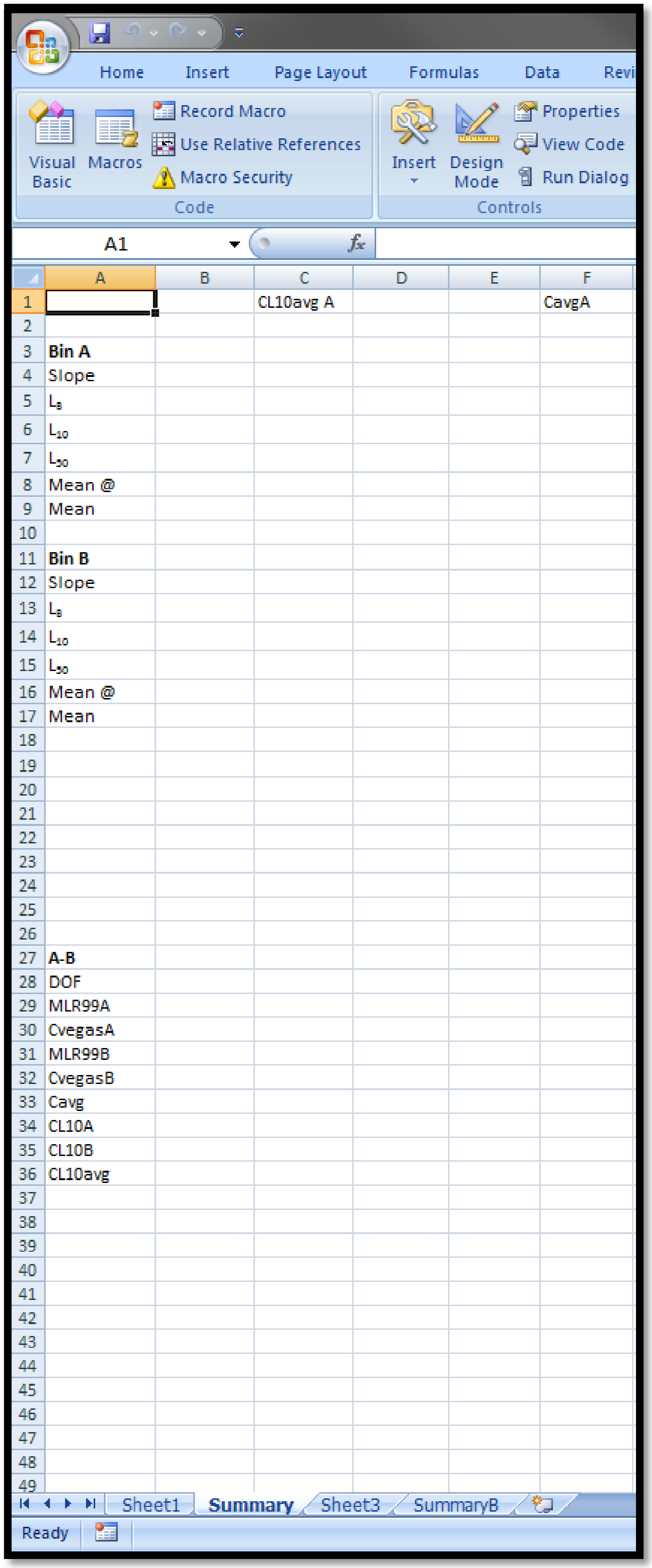


Figure 28. Screen shot of output cells.

In addition to holding all the values from the calculations, the summary page is where the numerical counting takes place for the Monte Carlo method of determining a confidence number. All the L_{10} lives are counted and determined how many of bin A were greater than bin B.

The Visual Basic Macro of the Simulation before Suspensions Incorporated

The model was developed using Visual Basic for Applications in Microsoft Excel. Excel was the program of choice because of its ability to handle and organize large amounts of data. It was also possible to interface with the Visual Basic code through the spreadsheet.

Visual Basic in Excel was developed to make handling small tasks easier. One method of doing this is by using a macro in Excel. A macro was constructed, for simplicity, by clicking the "Record Macro" button and following a sequence of steps for whatever task needs to be completed. The macro is assigned a shortcut key, ctrl+(a letter). For example, if "Record Macro" was pressed, then the user clicked on an empty cell, and entered the command to take the average of a series of numbers, and then highlight a series of numbers and pressed enter, then clicked stop recording, the user would be able to repeat this process by simply hitting the macro short key.

Once a quick "record" macro has been constructed it is then possible to access the macro and edit it using Visual Basic code. This is how this model was constructed.

The program was constructed over a number of weeks in small modules. Each module would be run, and then compared to hand calculations to be sure the code was correct and then the next module would be added on until the entire program ran as one unit. The initial set up was on the first page of the spread sheet as shown in figure 29.

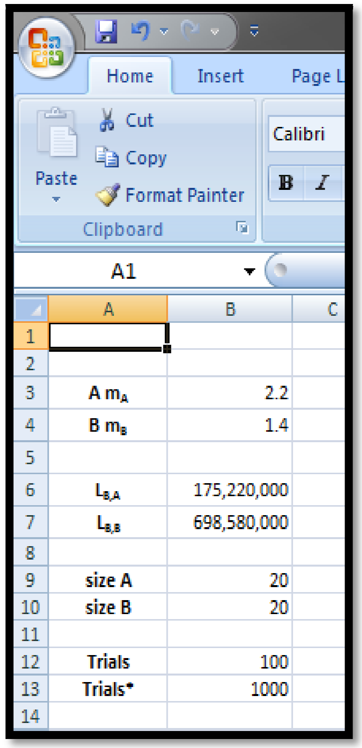


Figure 29. Screen shot of simulation inputs.

There were cells assigned to values that would be inputted by the user. These inputs included the Weibull slope for each bin, the characteristic life for each bin, the size of each bin, and the number of trials to be run.

First the set of random numbers that represented the number in which the sample failed was generated. The number generator in Visual Basic generated a random number between 1 and 1000. To ensure that no two same random numbers appeared in either of the sets generated, a sub-routine was written that compared each randomly pulled number to those already selected for the data set (figure 10). If found to equal one of the previously pulled numbers, the number in question was discarded, another pulled, and uniqueness again established.

The random numbers were then ordered from smallest to largest to be ranked. To rank them equation 3 was used. This rank was then turned in to a survivability value by subtracting it from 1. This survivability was then incorporated into the Weibull equation (equation 1), along with the Weibull slope and characteristic life as reported in the literature to determine a life. The code for determining a rank, then survivability, then a life is shown in figure 30.

```
For loop5 = 1 To alpha
'Rank A
    Cells(r1, c1) = (Cells(r1, c2) - 0.3) / (1000 + 0.4)
'S of A
    Cells(r1, c4) = 1 - Cells(r1, c1)
    SA = Cells(r1, c4)
'Life of A
        A1 = Application.WorksheetFunction.Ln(1 / SA)
        B1 = Application.WorksheetFunction.Ln(A1)
        CA = Exp(B1 / Cells(3, 2)) * Cells(6, 2)
        Cells(r1, c6) = CA
    r1 = r1 + 1
Next loop5
```

Figure 30. Subroutine for determining a life from random number.

Once the lives were established, a new rank was calculated based upon the size of the pulled population rather than 1000, this time from 1 to population size. Once the rank was established again, the survivability associated with each life was calculated. The lnln(1/S) and the ln(L) were calculated to determine a Weibull slope for the generated data. The code for determining the new rank, new survivability, lnln(1/S), and ln(L) is shown in figure 31.

```
For loop7a = 1 To alpha
'rank for 1 to sample size
        Z1 = (Cells(r11, 3) - 0.3) / (alpha + 0.4)
        Cells(r11, 8) = Z1

'S for sample size
        Z2 = 1 - Cells(r11, 8)
        Cells(r11, 9) = Z2

'lnln(1/S) for sample size
        Z5 = 1 / Z2
        Z3 = Application.WorksheetFunction.Ln(Z5)
        Z4 = Application.WorksheetFunction.Ln(Z3)
        Cells(r11, 10) = Z4
    r11 = r11 + 1
Next loop7a

'lnA
r12 = 7
For loop8 = 1 To alpha
     Cells(r12, 11) =
     Application.WorksheetFunction.Ln(Cells(r12, 7))
r12 = r12 + 1
Next loop8
```

Figure 31. Subroutine for calculating rank, survivability, lnln(1/S), and ln(L).

All calculations were performed for both bin A and bin B. The complete code can be found in the appendix. After all the previous calculations have been made, the spreadsheet is set up to begin finding the Weibull slope, L_{10}, L_{50}, and mean lives of the current data. These numbers were outputted to cells B16 to B29 (See figure 32).

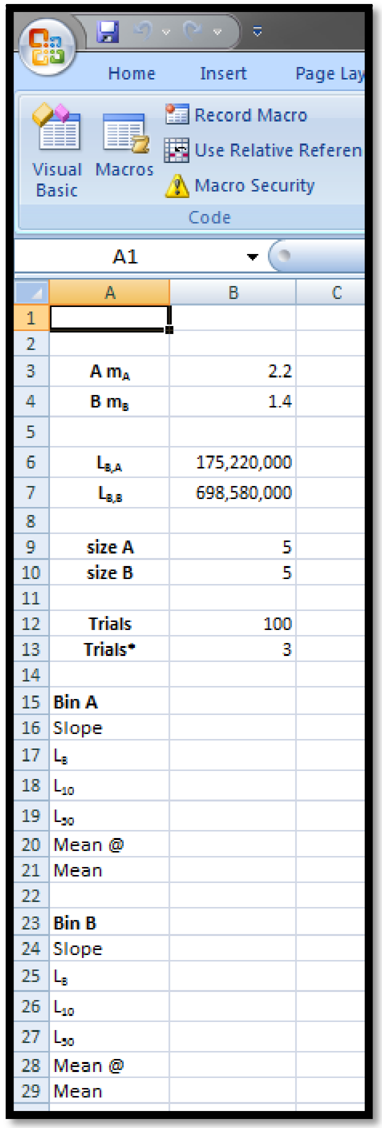


Figure 32. Screen shot of inputs and outputs.

The slope was found by virtually plotting the lnln(1/S) by the ln(L) using the Application.Worksheet.LinEst command in Visual Basic. The full code can be viewed in figure 33.

```
'slope A
MA = Application.WorksheetFunction.LinEst(Range(Cells(7, 10),
Cells(alpha + 6, 10)), Range(Cells(7, 11), Cells(alpha + 6,
11)), True, True)
Cells(16, 2) = MA
MAa = Cells(16, 2)
Slope1A = MAa
```

Figure 33. Code for determining Weibull slope of new virtual data

The L_{10} and L_{50} lives were calculated using the Weibull equation with S values of 0.9 and 0.5 respectively. The mean was found using the Gamma function method using the code in figure 34.

```
'Mean A Gamma function method
musb = (MAa + 1) / MAa
a1a = Application.WorksheetFunction.GammaLn(musb)
LmeanA = LBa * Exp(a1a)
Cells(21, 2) = LmeanA
cheeseA1 = LmeanA
```

Figure 34. Code for determining mean by Gamma function method.

The curve fit equations were then calculated. All confidence numbers and the averages of the Weibull slopes, L_{10} and L_{50} lives were then displayed on the “SummaryB” sheet as shown in figure 35.

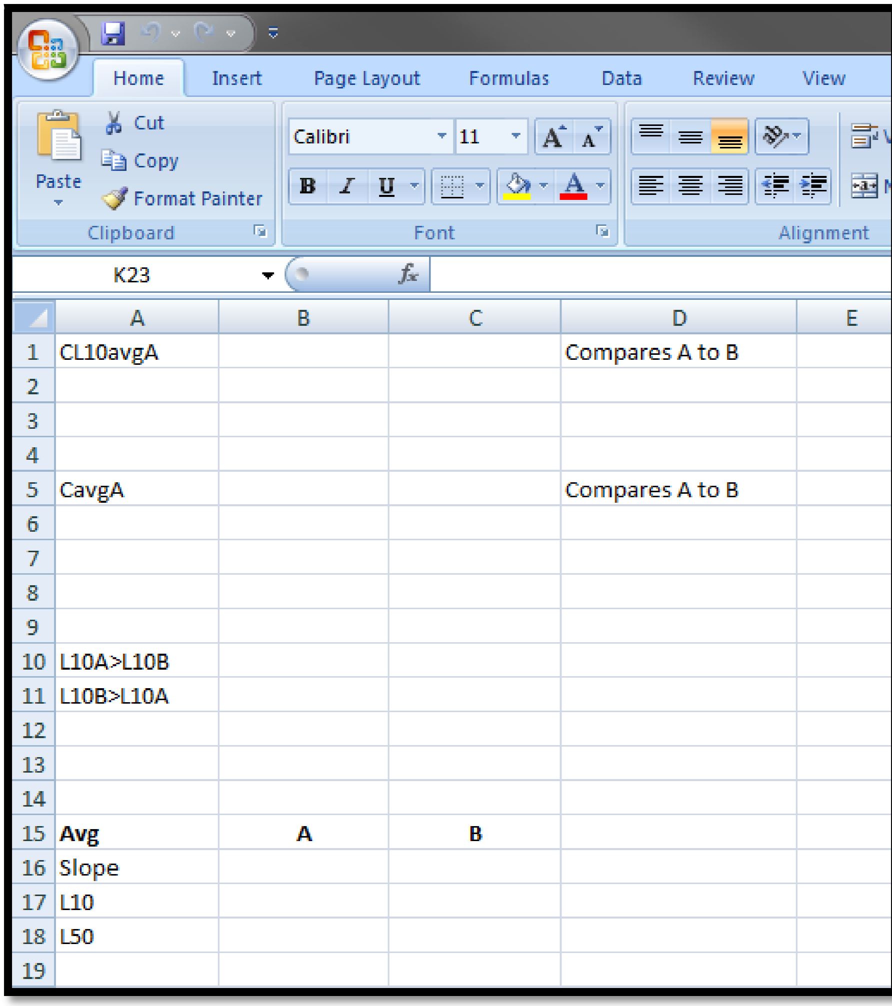


Figure 35. Screen shot of SummaryB sheet.

Time and Memory Used in Runs

Each simulation was set up in two main loops. One loop would run 100 times in order to count how many times out of 100 the L_{10} of one material was greater than the L_{10} of the other material. The other loop was how many times the simulation would do this. Over all, it was performed 10,000 times. Due to limitations of the RAM of the computers used, it was not possible to set up the simulation to just run 10,000 times. For methods 1-4 the simulation was broken down into 5 segments each of which ran 2,000 runs. Depending on the capabilities of the computer these runs took on average one hour. The runs were split up between my home PC, my laptop, and the computers in the Engineering Building at Georgia Southern University.

Method 5, however, took significantly longer. It was written into the program to restart a run if the lives generated did not fall into the designated cutoffs. Because of the randomness of the simulation, it took many trials to get to the total 10,000 runs. The method 5 simulations were broken down into runs of 50, 100, 500, 1000, and 2000. The runs comparing AISI 9310 gears to NASA Modified Vasco X-2 gears were able to finish within 5 to 10 hours running 2000 at a time. The Boeing Vertol runs were split into 25 sections, each one running 400 runs. Each one of these was also able to finish in 5 to 10 hours. The AISI 9310 vs. Modified Vasco X-2 rolling contact simulations were broken down into runs of 50 and 100. Depending on the computers these could take between 12 and 24 hours to run.

All data totaled to over 6GB of memory. The 2,000 runs were on the 80MB range while the 50 runs were in the 2MB range.

Summary of the Methodology

The general methodology is as follows.

1) A macro was constructed using Microsoft Visual Basic employing the Monte Carlo “bin” technique of generating random values to be calculated into lives by the use of the Weibull equation (equation 1).
2) Hand calculations were performed to ensure each code module was making the correct calculations.

3) These lives were counted to determine how many times the life of one population was larger than the other, this gave the confidence number based upon the counting method.
4) This was done with five different methods. There were five methods to incorporate suspensions that were present in the original data. Each method has a different way of determining a confidence number from the way the random numbers are generated.
5) In each method confidence numbers due to the curve fit equations by Vlcek, Zaretsky, and Hendricks were also computed for comparison.
6) This process was simulated 10,000 times for each method, for statistical certainty.
7) After all the confidence numbers from the simulations were computed they were compared to existing fatigue data, which contained confidence numbers determined from Johnson curves, to validate them.

Chapter 4

Results and Discussion

Introduction

A computer simulation was written in Microsoft Visual Basic using Excel to statistically determine the difference between two materials or components involved in a fatigue experiment, including suspensions, with confidence. The program was modified from previous Monte Carlo models as well as derived from previous statistical models in which all specimens have failed. The simulation modeled fatigue data with out-life suspensions. There are five methods presented to determine which material or component is better. The following are results of the simulation that were compared to published results (Townsend, Zaretsky and Anderson, Comparison of Modified Vasco X-2 with AISI 9310 - Preliminary Report 1977), (Townsend and Zaretsky, Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels 1980), to validate the findings.

Each method represents what could have happened in the original experiment. As mentioned earlier, some fatigue tests may be cut short by the experimenter due to their lengthy lives. This is taken into account in methods 3, 4, and 5. These methods force the program to use only the specified number of lives in the calculation of the confidence number.

Suspensions

The method of calculating confidence numbers in the papers is not known. It is also not known whether or not the suspensions are out-lives or are contained within the data. It is assumed, however, that the lives are out-lives due to the trends shown in the graphs in the literature (figure 36). The Monte Carlo simulation of out-life suspensions methods presented were designed with this assumption.

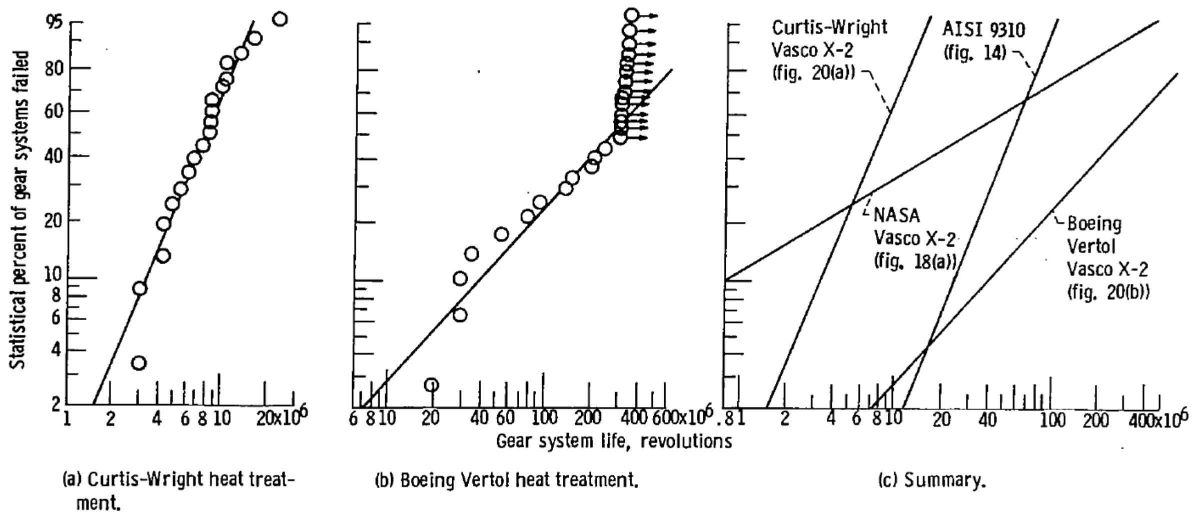


Figure 36. Surface pitting fatigue life of CVM modified Vasco X-2 spur gears heat treated to different specifications. Pitch Diameter 8.39 centimeters; speed, 10,000 rpm; lubricant, synthetic paraffinic oil; gear temperature, 350 K; maximum Hertz stress, $1.71x10^9$ N/m^2.

The differing reasons for a test having suspensions are also taken into account in the five methods. If a piece of previous work is revisited with this program, it is not always clear why the suspensions occurred. This is another reason for the five different methods presented here.

The only assumption necessary is that there is fatigue data present and within that data are suspensions. It is not necessary to know how or why there are suspensions in the data.

In all 5 methods, the program records how many times out of 100 the L_{10} life of Modified Vasco X-2 is greater than AISI 9310 as well as how many times out of 100 the L_{10} life of AISI

9310 is greater than Modified Vasco X-2. The confidence number that is reported is always the bigger of the two numbers.

The Five Methods

Method 1

In method 1, it is assumed that the experiment underwent the ideal case in which all samples tested actually failed—i.e. no suspensions. Method 1 is the original program written to determine which material is better if all samples failed. It does have its place in the analysis of data containing suspensions though. It hypothesizes the view of what could have happened had all the specimens failed.

From this method, trends can be observed as to what could have potentially happened had all samples failed. This could show that, perhaps, there would have been no difference in the data or on the other hand could show that there would have been a significant difference. If a large difference is observed in the two samples from this method, and not in the other methods, this could cause concern and possibly mean the experiment should be performed again with no suspensions to ensure accurate results.

In this case, it means that all 20 of Modified Vasco X-2 failed, and all 20 of AISI 9310 failed. The result of this test gave a Monte Carlo confidence number of 92. This number, however, is on the wrong side of the statistically different boundary. It states that, if there were no suspensions in the original experiment, then there may be a statistical difference between the two materials. The curve fit equations, however, from the simulation reported confidence numbers of 83 for the L_{10} equation. These numbers fall in accordance with the original data. These results are summarized in table 7.

Table 7. Results of method 1 model rolling contact fatigue test	
Confidence numbers generated by simulation	Method 1
L_{10} Curve fit equation	83
Monte Carlo	92

Method 1 which assumes all samples tested have reached failure offers a confidence number of 77 for AISI 9310 vs. Boeing Vertol Modified Vasco X-2. In the original experiment, the confidence number was 80. These agree with each other in stating that there is no statistical difference between the two materials. The simulation generates a confidence number of 100 for AISI 9310 vs. NASA Modified Vasco X-2 which relates to the confidence number of 99 which was reported in the original paper. This states that there is a statistical difference between AISI 9310 and NASA's heat treated Modified Vasco X-2. The simulation also generated a confidence number of 100 for AISI 9310 vs. Curtis-Wright Modified Vasco X-2 which also relates to the confidence number of 99 as reported in the original paper. The results of method 1 are summarized in table 8.

Table 8. Results of method 1 model gear fatigue test			
Confidence Numbers	Monte Carlo Method 1	Monte Carlo Curve Fit	Experimental/Graphical (Townsend and Zaretsky)
AISI vs. Boeing Vasco	77	76	80
AISI vs. NASA	100	96	99
AISI vs. Curtis-Wright	100	100	99

Method 2

The failure index as reported in the literature is used as the inputs for sample size in method 2. Instead of inputting the number of specimens tested, the number of failed specimens is entered. This represents the actual number of specimens failed. This is a description of the actual number of failures in the original experiment. As in method 1, trends can be observed as to what could have happened had there been no suspensions in the data. Another conclusion that can be drawn from this method is what could potentially happen if outliers are left out of the analysis of data.

In this method only 6 samples of AISI 9310 were tested as opposed to 20. The inputs were 20 samples of Modified Vasco X-2 and 6 samples of AISI 9310. This method presupposes the correct number of failed samples in relation to the original experiment. The Monte Carlo confidence number was 78. This falls in accordance with the original experiment; there is no statistical difference between the two materials. The L_{10} life curve fit confidence number was 79. This also follows the experiment, no statistical difference between the two materials. These results are summarized in table 9.

Table 9. Results of method 2 rolling contact fatigue test	
Confidence numbers generated by simulation	Method 2
L_{10} Curve fit equation	79
Monte Carlo	78

Method 2 again assumes that the number of samples failed equaled the number of samples tested. In the original experiment all 30 of the 30 samples tested of AISI 9310 failed.

Only 12 of the 26 samples of Boeing Vertol Modified Vasco X-2 failed and only 18 of the 21 samples of NASA Modified Vasco X-2 failed. There were no suspensions in the Curtis-Wright data so it was not used in any of the remaining methods.

The inputs of method 2 were 30 samples of AISI 9310 with, 12 samples of Boeing Vertol Modified Vasco X-2 and 18 samples of NASA Modified Vasco X-2. The Monte Carlo confidence number comparing AISI 9310 and Boeing Vertol Modified Vasco X-2 was 68. This was low compared to the original value of 80 from the paper but is still in agreement that they are not statistically different. The Monte Carlo confidence number comparing AISI 9310 and NASA Modified Vasco X-2 was 100. This again agrees with the confidence number from the paper which was 99. This shows there is a statistical difference between the AISI 9310 gear and then NASA heat treated Modified Vasco X-2 gear. These results are summarized in table 10.

Table 10. Results of method 2 gear fatigue test

Confidence Numbers	Monte Carlo Method 2	Monte Carlo Curve Fit	Experimental/Graphical (Townsend and Zaretsky)
AISI vs. Boeing Vasco	68	76	80
AISI vs. NASA	100	95	99

It is assumed that the confidence number generated by the L_{10} curve fit equation for method 2 should most closely agree with the graphical confidence number because the curve fit equations were developed (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007), (Vlcek, Hendricks and Zaretsky, Relative Ranking of Fatigue Lives of Rotating Aluminum Shafts Using L10 Weibull-Johnson Confidence Numbers 2008) based off the graphical method presented by Johnson (Johnson, Theory and Technique of Variation Research 1964), (Johnson, The Statistical Treatment of Fatigue Experiments 1964). Method 2 represents how the confidence number would be calculated graphically. The only lives given are those reported and the degrees of freedom is also the same.

Method 3

In method 3, suspension out-lives were intentionally generated. It is designed to closely represent what happened in the original experiment in that the total attempted lives were generated but only uses the number of actual failed to calculate the confidence number.

To begin this method the amount of specimens tested for each material, and the amount of specimens that actually failed were inputted. The input of actual failures is used as the cutoff point when calculating confidence numbers from the generated random lives. As mentioned in the method, the program generated random numbers between 1 and 1000 to represent the number in which failure occurred in the particular "bin" of components. These numbers were then ranked, survivability was determined, and the Weibull slope and characteristic life were put into the Weibull equation (equation 1) to calculate a virtual life. These lives are then used in the fitted equations and in the Monte Carlo counting simulation to determine confidence numbers. The number of random numbers chosen was determined by the input of the number of specimens tested. To incorporate the suspensions into this, it was necessary to generate lives for all specimens tested and then select a certain few to be used in the calculation of the confidence number. The number of samples failed as inputted, is programmed into the simulation as the cutoff point. The program will generate lives for the number of specimens tested and then only use the lowest lived lives, as specified by the user, to calculate the confidence numbers.

For comparison of Modified Vasco X-2 and AISI 9310, 20 random lives for Modified Vasco X-2 and 20 lives for AISI 9310 were generated. When lives are used for calculating confidence numbers, however, it only uses the lowest valued 6 lives for AISI 9310. All 20 lives of Modified Vasco X-2 were used. This allows the program to generate 20 lives for AISI 9310 in which they are spread out from minimum to maximum and then only the lowest 6 are used, leaving the other 14 as out-lives.

The six life cutoff point was used because in the paper (Townsend, Zaretsky and Anderson, Comparison of Modified Vasco X-2 with AISI 9310 - Preliminary Report 1977) there were only 6 failures and the assumption was made that any specimens over those 6 were out-lives due to the graph in the original paper. It is assumed from this graph (figure 37) that both the Modified Vasco X-2 and AISI 9310 samples stopped in the range of 380,000,000 cycles.

There was a discrepancy in the original paper (Townsend, Zaretsky and Anderson, Comparison of Modified Vasco X-2 with AISI 9310 - Preliminary Report 1977), according to the table the L_{10} lives of the materials were 63 million cycles and 140 million cycles, however, according to the graph of this data (figure 37), it shows L_{10} lives of 6.3 and 14 million cycles. There was a decimal error somewhere but this did not affect the results. For this work, it was assumed that the L_{10} lives were 63 and 140 million cycles.

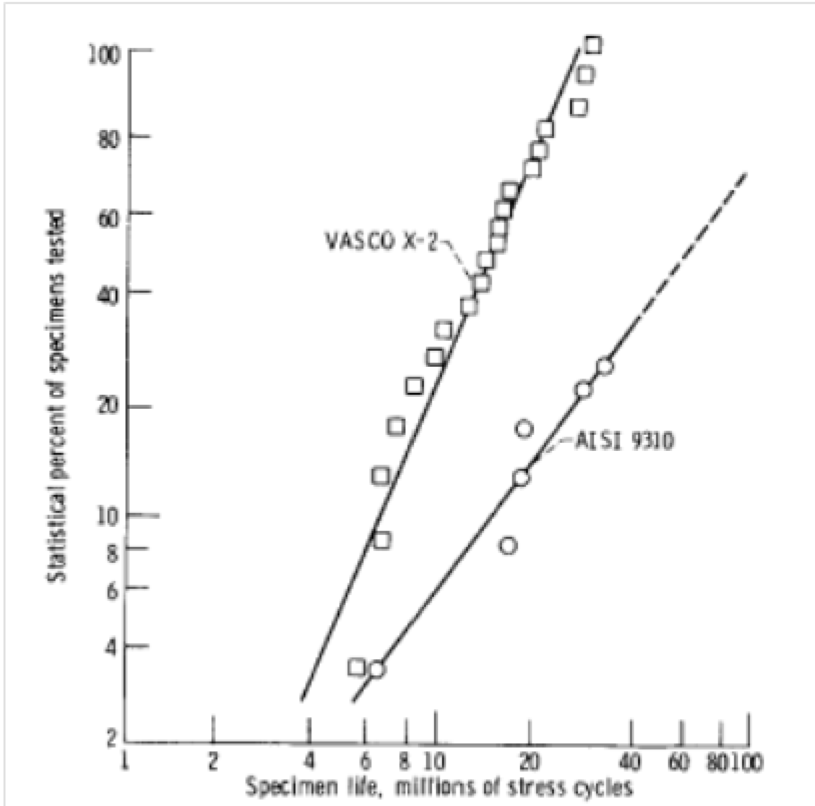


Figure 37. Comparison of rolling-element fatigue lives of AISI 9310 and Vasco X-2 in rolling contact tester (Townsend, Zaretsky and Anderson, Comparison of Modified Vasco X-2 with AISI 9310 - Preliminary Report 1977)

Method 3 is more indicative of what actually happened in the original experiment if the suspensions were out-lives. Based on the data, it appears that the experiment was cut short once a certain maximum life was reached. Method 3 simulates what would happen if every time the experiment was run, that only the 6 lowest lives of AISI 9310 were used to calculate the confidence number. For method 3, the Monte Carlo confidence number was 92. The confidence number for the curve fit equations was 79 for the L_{10} life curve fit. These results are summarized in table 11.

Table 11. Results of method 3 model rolling contact fatigue test	
Confidence numbers generated by simulation	Method 3
L_{10} Curve fit equation	79
Monte Carlo	92

For the first simulation of the second paper (Townsend and Zaretsky, Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels 1980), 30 lives for AISI 9310 were generated and all 30 lives were used in the calculations and 26 lives for Boeing Vertol Modified Vasco X-2 were generated but only the lowest 12 lives were used for the confidence number calculations. Similarly, for the second simulation, 30 lives for AISI 9310 were generated and used and 21 lives for NASA Modified Vasco X-2 were generated but only the lowest 18 were used for the confidence number calculations. The confidence number generated by the simulation for AISI 9310 vs. Boeing Vertol Modified Vasco X-2 was 77 which closely agrees with the original confidence number of 80 from the paper (Townsend and Zaretsky, Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels 1980). This states that there is no statistical difference between the two materials. The confidence number generated by the simulation for AISI 9310 vs. NASA Modified Vasco X-2 was 100 which agrees with the confidence number of 99 from the original paper (Townsend and Zaretsky, Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels 1980). This states that there is a statistical difference between the two materials. These results are summarized in table 12.

Table 12. Results of method 3 gear fatigue test			
Confidence Numbers	Monte Carlo Method 3	Monte Carlo Curve Fit	Experimental/Graphical (Townsend and Zaretsky)
AISI vs. Boeing Vasco	77	74	80
AISI vs. NASA	100	95	99

Method 4

Method 4 works under the same principle as method 3. It again uses an input as a cutoff point to determine how many lives to use in the calculation of the confidence numbers. This method differs from Method 3 in that instead of using a number of samples as the cutoff point, it uses a specified life as the cutoff. The program essentially does what an experimenter would do, i.e. stop a run once it reaches a certain number of cycles. This method allows for a closer representation of the original experiment. If the suspensions in the original experiment were caused due to the experimenter stopping the test because they reach a maximum life this simulation should closely resemble the results. As in method 3, the user inputs the total number of specimens tested and then inputs the life at which it should use as the cutoff. The program will then generate the total number of lives and only use the lives within the cutoff window for the calculation of the confidence numbers.

In the original paper, it was not stated why only 6 of the specimens failed. It was deduced from one of the original graphs that the specimens were probably stopped at a certain life because it was not necessary to carry on (out-life suspensions).

For the comparison of Modified Vasco X-2 and AISI 9310, 20 random lives were generated for Modified Vasco X-2 and 20 lives for AISI 9310. Once these lives were generated, a set cutoff life was used to determine the confidence number. This represents what would happen if the experiment was cut short due to the sample reaching a maximum life. The cutoff point used was 380 million lives. This life was chosen because it was the maximum life as shown in the original paper.

The only lives used to compute the confidence numbers were between 1 and 380 million. This could be anywhere between 2 samples and all 20 samples, as long as it falls within that window. There had to be at least 2 lives to use the plotting function in Visual Basic to determine a Weibull slope. The Monte Carlo confidence number generated by method 4 was 92; the L_{10} curve fit equation confidence number was 79. The curve fit confidence number agrees with the original paper, however the Monte Carlo confidence number does not. These results are summarized in table 13.

Table 13. Results of method 4 rolling contact fatigue test

Confidence numbers generated by simulation	Method 4
L_{10} Curve fit equation	79
Monte Carlo	92

Again for method 4, a specified life was used as the cut off criteria for the simulation. Lives were generated for the amount of attempted samples and then the defined life was used as the cut off to perform the calculations to determine the confidence number. The cut off life was 400 million.

For this simulation, 30 lives were generated and used for the AISI 9310 gears. This was compared to the 26 lives of Boeing Vertol Modified Vasco X-2, and the 21 lives of NASA Modified Vasco X-2.

The Monte Carlo confidence number calculated comparing AISI 9310 to Boeing Vertol Modified Vasco X-2 was 76, which compares closely to the original confidence number of 80. The Monte Carlo confidence number calculated comparing AISI 9310 with NASA Modified Vasco X-2 was 100, compared to the original confidence number of 99. Both of these confidence numbers agree with the original experiment in that there is no statistical difference between AISI 9310 gears and the Boeing Vertol heat treated Modified Vasco X-2 gears, and there is a statistical difference between AISI 9310 gears and the NASA heat treated Modified Vasco X-2 gears. These results are summarized in table 14.

Table 14. Results of method 4 gear fatigue test

Confidence Numbers	Monte Carlo Method 4	Monte Carlo Curve Fit	Experimental/Graphical (Townsend and Zaretsky)
AISI vs. Boeing Vasco	76	75	80
AISI vs. NASA	100	95	99

Method 5

Method 5 was designed to be a hybrid of methods 3 and 4. As in methods 3 and 4, the number of lives generated was the number of specimens tested. In method 5, the cutoff points from both methods 3 and 4 were used together. The inputs specified were how many suspended lives to generate as well as what range of lives to fall in.

In the comparison of Modified Vasco X-2 and AISI 9310, 20 lives for Modified Vasco X-2 were generated and 6 lives for AISI 9310 were generated but would force a cutoff life of 380 million cycles. It would force 6 lives that fall between 1 and 380 million cycles for AISI 9310. This method was thought to be the one that would most closely resemble the original experiment; however, the results proved that to not be the case. The Monte Carlo confidence number generated was 52. The L_{10} curve fit confidence number was 74.

Table 15. Results of method 5 rolling contact test	
Confidence numbers generated by simulation	Method 5
L_{10} Curve fit equation	74
Monte Carlo	52

The curve fit confidence number was in close agreement with the experimental value of 80 but the Monte Carlo confidence number was significantly off. This is believed to have happened because the sample size of 6 is so small compared to the life range of 1 to 380 million cycles. The scatter in the data was extreme and reasonable outcomes were rare.

In comparing AISI 9310 with Boeing Vertol and NASA Modified Vasco X-2, the cutoff sample numbers were 12 and 18 respectively and the cutoff life was 400 million cycles for both cases. The Monte Carlo confidence number generated for AISI 9310 against Boeing Vertol Modified Vasco X-2 was 52 and the L_{10} curve fit confidence number was 75. Again, this poor Monte Carlo confidence number is due to the low number of samples and wide range of lives generated. The Monte Carlo confidence number generated for AISI 9310 against NASA Modified Vasco X-2 was 100 and the L_{10} curve fit confidence number was 96. These numbers were in excellent agreement with the original confidence number of 99.

Table 16. Results of method 5 gear fatigue test			
Confidence Numbers	Monte Carlo Method 5	Monte Carlo Curve Fit	Experimental/Graphical (Townsend and Zaretsky)
AISI vs. Boeing Vasco	52	75	80
AISI vs. NASA	100	96	99

Summary of Simulation Results

The numbers generated were in close agreement with the original data. In table 17 is a summary of the results of the simulation for the comparison of Modified Vasco X-2 with AISI 9310 rolling contact fatigue test.

Summary of Rolling Contact Fatigue Test

Table 17. Summary of results of methods 1-5 for rolling contact fatigue test vs. experimental confidence number							
Material		Method					Experimental
		1	2	3	4	5	
Modified Vasco X-2 vs. AISI 9310	Curve fit	83	79	79	79	74	84
	MC	92	78	92	92	52	

Table 17 includes the confidence number using the L_{10} life curve fit equations, the generated Monte Carlo (counting) confidence number, and the confidence number determined graphically. The curve fit confidence numbers were calculated using the equations developed by Vlcek, Zaretsky, and Hendricks (Vlcek, Hendricks and Zaretsky, Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters 2007), (Vlcek, Hendricks and Zaretsky, Relative Ranking of Fatigue Lives of Rotating Aluminum Shafts Using L10 Weibull-

Johnson Confidence Numbers 2008). The Monte Carlo confidence numbers are the numbers generated by using the random life generator and then counting from run to run which L_{10} life is greater.

Summary of Gear Fatigue Test

In table 18 is a summary of the Monte Carlo, L_{10} curve fit, and graphical confidence numbers for comparing AISI 9310 to the three (Boeing Vertol, NASA, Curtis-Wright) heat treatments of Modified Vasco X-2 gear steels.

Table 18. Summary of results of methods 1-5 for gear fatigue test vs. experimental confidence number

Material	Heat Treatment		Method 1	Method 2	Method 3	Method 4	Method 5	Experimental
AISI 9310	----	----	----	----	----	----	----	----
Modified Vasco X-2	Boeing Vertol	Curve fit	76	76	74	75	75	80
		MC	77	68	77	76	52	
	NASA	Curve fit	96	95	95	95	96	99
		MC	100	100	100	100	100	
	Curtis-Wright	Curve fit	100	----	----	----	----	99
		MC	100	----	----	----	----	

The only materials to have suspensions in their tests were the NASA Modified Vasco X-2 and the Boeing Vertol Modified Vasco X-2. The two other materials had all of their samples fail. Even though all of the Curtis-Wright Modified Vasco X-2 samples failed the data was still run through the simulation to validate it because the simulation was original written to run data with no suspensions. This means the Curtis-Wright data was only used in method 1 where all samples of each material failed. This is why the boxes are blank for methods 2, 3, 4 and 5.

Comparison of Weibull Slopes, L_{10}, and L_{50} Lives

In calculating the confidence numbers, Weibull slopes and L_{10} lives were generated in the simulation. These values were recorded for each run and then averaged to compare to the original values published in the literature. Table 19 summarizes the Weibull slopes, L_{10}, and L_{50} lives generated by all five methods as well as the graphically determined values from the original experiment comparing Modified Vasco X-2 to AISI 9310 rolling contact fatigue samples.

Table 19. Summary of Weibull slope, L_{10}, and L_{50} numbers generated by simulation vs. experimental numbers

Material		Method					Experimental
		1	2	3	4	5	
Modified Vasco X-2	Weibull Slope	2.11	2.11	2.11	2.11	2.11	2.2
	L10	60,027,376	60,016,270	60,026,020	77,706,402	60,020,752	63,000,000
	L50	148,353,772	148,345,740	148,358,838	148,378,744	148,364,754	148,000,000
AISI 9310	Weibull Slope	1.35	1.42	1.52	1.58	1.95	1.4
	L10	133,989,948	144,470,418	140,761,638	140,230,434	70,829,292	140,000,000
	L50	541,389,859	549,152,686	748,699,326	1,120,168,002	195,337,596	570,000,000

In table 20 are the values comparing AISI 9310 gear steels to the three (Boeing Vertol, NASA, Curtis-Wright) heat treatments of Modified Vasco X-2.

Table 20. Summary of Weibull slope, L_{10}, and L_{50} numbers generated by simulation vs. experimental numbers								
Material	Heat Treatment		Method					Experimental
			1	2	3	4	5	
AISI 9310	----	Weibull Slope	2.22	2.22	2.22	2.22	2.22	2.30
		L10	22,096,686	22,096,554	22,093,585	22,101,670	22,096,324	23,000,000
		L50	52,179,303	52,179,988	52,176,856	52,180,906	52,181,944	52,000,000
Modified Vasco X-2	Boeing Vertol	Weibull Slope	0.96	0.96	0.99	0.98	1.15	1.00
		L10	37,461,804	39,423,526	38,739,860	38,320,327	27,559,164	38,400,000
		L50	256,415,183	260,420,049	309,002,024	276,309,799	134,956,578	253,000,000
	NASA	Weibull Slope	0.51	0.51	0.51	0.51	0.54	0.53
		L10	982,878	1,031,184	1,022,719	1,016,736	977,826	800,000
		L50	30,456,126	30,863,060	31,958,843	30,845,819	24,037,944	27,600,000
	Curtis-Wright	Weibull Slope	2.02	----	----	----	----	2.10
		L10	3,141,356	----	----	----	----	3,300,000
		L50	8,097,877	----	----	----	----	8,000,000

Summary of Comparison of Modified Vasco X-2 with AISI 9310 – Preliminary Report

The first experiment the program was compared to was performed in 1977 by Dennis Townsend, Erwin Zaretsky, and Neil Anderson (Townsend, Zaretsky and Anderson, Comparison of Modified Vasco X-2 with AISI 9310 - Preliminary Report 1977). In this experiment Townsend, Zaretsky, and Anderson were concerned which material would be more suitable for gears in helicopter transmissions.

With advances being made in the helicopter industry, the gears in the transmissions were reaching extreme temperatures (above 250° F). This was exceeding the limits of the gear material currently available. This material was AISI 9310 steel. The new gear material they decided to test was Modified Vasco X-2. This material was originally used as a tool steel. In order to make the Vasco X-2 suitable for helicopter transmission gears, the carbon content was lowered and then the material was case hardened leaving a softer core. The objectives of their research were to a) determine the performance of spur gears made from the new material modified Vasco X-2, b) compare the fatigue lives of the old material AISI 9310 against the new material modified Vasco X-2, and c) to compare the hot hardness retention of the two materials. This was going to be accomplished by testing the spur gears with different heat treatments, rolling contact fatigue tests, and hardness tests.

The specific test this current research was concerned with was the rolling element test. In this test a 3 inch long rod with 0.375 inch diameter was inserted into the rolling contact test apparatus (figure 38). This tester consisted of two rolling discs of 7.5 inch diameter made of AISI M-50 steel which were heat treated to the same hardness as the samples. The samples were placed in between the two discs and a load was applied until the sample was able to turn both discs. Once the discs and sample were in thermal equilibrium the maximum load (700,000 psi) was applied. The specimen would be rotated at 12,500 rpm until failure occurs. The tester would shut down automatically by means of a vibration detector. The lives of the specimens, denoted by number of rotations until failure, were used to calculate a confidence number to determine which material was more suitable for the application. In the experiment 20 samples of Modified Vasco X-2 were tested and all 20 of them failed. AISI 9310 also had 20 samples reported as tested, but only 6 reached failure.

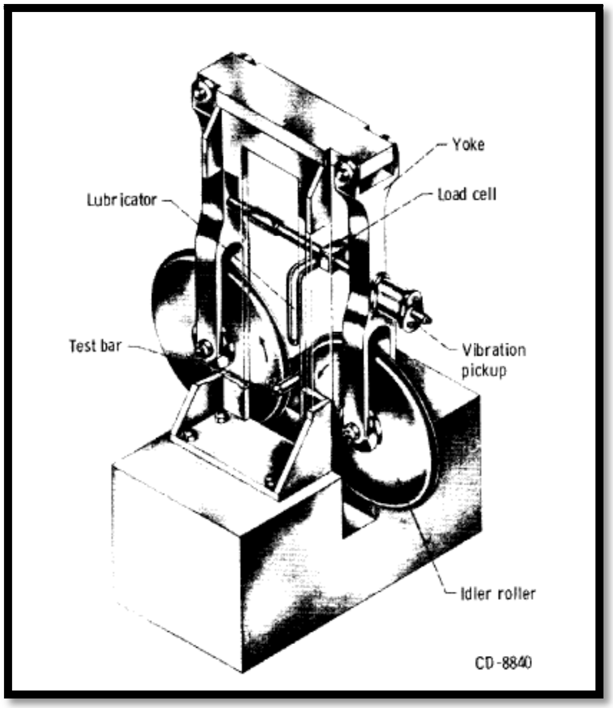


Figure 38. Rotational fatigue tester (Townsend, Zaretsky and Anderson, Comparison of Modified Vasco X-2 with AISI 9310 - Preliminary Report 1977)

The experiment determined that there was an 84% confidence between the two materials. This says that 84 times out of 100 AISI 9310 will last longer than Modified Vasco X-2. According to Johnson (Johnson, The Statistical Treatment of Fatigue Experiments 1964) this number states that there is no statistical difference between the two materials. There must be a confidence of 90 or greater to be determined statistically different. A summary of these results is in table 21.

Table 21. Fatigue life results from Comparison of Modified Vasco X-2 with AISI 9310

[Speed, 25 000 stress cycles per min.; maximum hertz stress, 4823×10^6 N/m^2 (700 000 psi); lubricant, MIL-L-7808, temp., ambient.]

Material	Life, millions of stress cycles		Weibull slope	Failure index (a)	Confidence number at 10-percent life level (b)
	10-Percent life	50-Percent life			
Modified Vasco X-2	63	148	2.2	20 out of 20	84
AISI 9310	140	570	1.4	6 out of 20	----

[a]Number of fatigue failures out of number of specimens tested.
[b]Percentage of time that 10-percent life obtained with AISI 9310 will have the same relation to the 10-percent life obtained with modified Vasco X-2.

The results of the gear tests of Townsend and Zaretsky (Townsend, Zaretsky and Anderson, Comparison of Modified Vasco X-2 with AISI 9310 - Preliminary Report 1977) were that the gears made from AISI 9310 survived on the magnitude of hours with millions of cycles. Failure occurred due to surface pitting or spalling. The modified Vasco X-2 gears, however, only survived in the 600,000 range for less than an hour and failure was due to tooth fracture.

A summary of the results is as follows:

1) Crack formation at the tips of the gear teeth during carburizing process of the modified Vasco X-2 resulted in fracture of the gear teeth after a period of less than one hour (600,000 revolutions) of operations under test conditions.
2) The lives of the AISI 9310 gears at a 90% probability of survival were 39.3, 19, and 7.1 hours at 222,000 psi, 248,000 psi, and 272,000 psi respectively.
3) Failure of the AISI 9310 gears was by surface pitting with no tooth fracture occurring.

4) The rolling element fatigue life of the AISI 9310 was approximately twice that of the modified Vasco X-2.
5) At temperatures of approximately 300 F there was no significant difference in hot hardness between the modified Vasco X-2 and AISI 9310 materials.

Summary of Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels

The second paper used for comparison was Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels by Dennis P. Townsend and Erwin V. Zaretsky, 1980 (Townsend and Zaretsky, Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels 1980). In this experiment, again, two different gear materials were tested to see which was superior. The AISI 9310 was compared to three different heat treatments of Modified Vasco X-2. The three different heat treatments came from three different vendors, Boeing Vertol, NASA, and Curtis-Wright. In this experiment, gears were tested as opposed to material rods as in the first experiment. The apparatus used is shown in figure 39. The results of the experiment are shown in table 22.

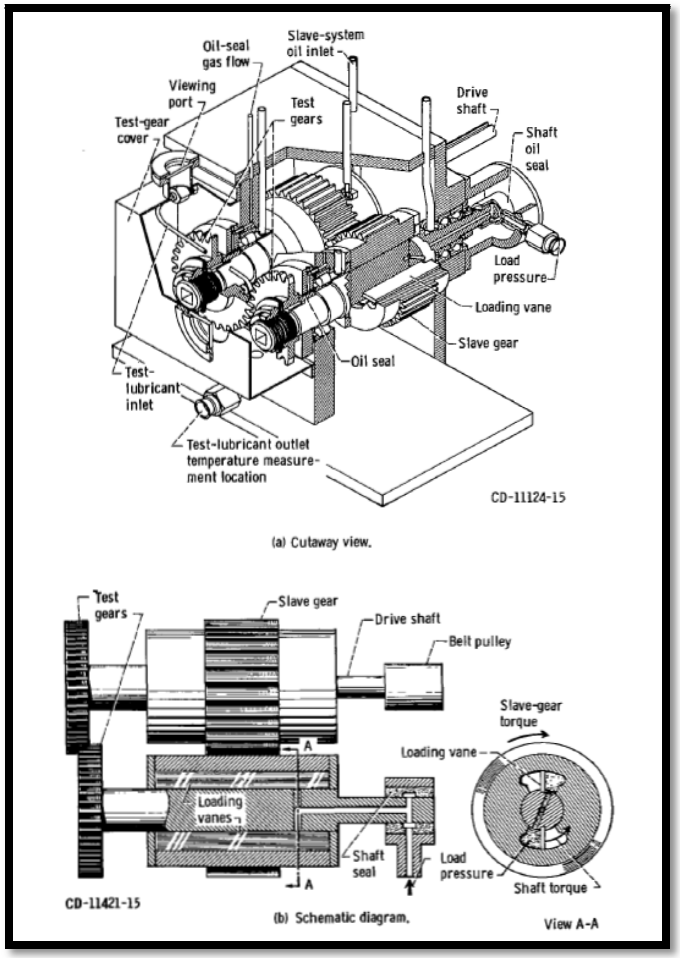


Figure 39. Gear tester used in original experiment (Townsend and Zaretsky, Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels 1980)

Table 22. Summary of gear fatigue life results from Comparison of Modified Vasco X-2 and AISI 9310 Gear Steels (Townsend and Zaretsky, Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels 1980)						
Material	Heat treat procedure	Gear system life, Revolutions		Weibull Slope	Failure Index[a]	Confidence number[b]
		10-Percent life	50-Percent life			
AISI 9310	---------	$23x10^6$	52×10^6	2.3	30 of 30	---
Modified Vasco X-2	Boeing Vertol	$38.4x10^6$	253×10^6	1.0	12 of 26	80
	NASA	0.8×10^6	27.6×10^6	0.53	18 of 21	99
	Curtis-Wright	3.3×10^6	$8 x10^6$	2.1	19 of 19	99

[a]Number of fatigue failures out of number of gears tested
[b]Percentage of time that 10-percent life obtained with AISI 9310 gears will have the same relations to the 10-percent life obtained with modified Vasco X-2 gears.

There was no statistical difference between AISI 9310 and the Boeing Vertol heat treatment of Modified Vasco X-2 with a confidence number of 80, however there was a statistical difference between AISI 9310 and the NASA and Curtis-Wright heat treated Modified Vasco X-2, both with confidence numbers of 99. It was reported that all 30 of the AISI 9310 gears failed, and all 19 of the Curtis-Wright gears failed, however, only 12 of the 26 of the Boeing Vertol gears failed, and only 18 of the 21 of the NASA gears failed. Graphs of the results are shown in figure 36.

Summary of Original Experimental Results

In the comparison of Modified Vasco X-2 with AISI 9310 rolling contact fatigue test the original experiment stated that the fatigue life of AISI 9310 was approximately twice that of Modified Vasco X-2, however, the two materials were not statistically different due to a confidence number of 84.

This was in agreement with the numbers calculated by methods 1-5. Seven of the ten confidence numbers were on the correct side of 90 to show no statistical difference, while the other three were 92, which is very close to the cutoff of 90. This leads to a conclusion by the methods presented that there is no statistical difference between the fatigue lives of AISI 9310 and Modified Vasco X-2.

In comparing AISI 9310 gears to the three different heat treatments of Modified Vasco X-2 gears, it was shown in the original experiment that Modified Vasco X-2 gears can be reasonably reliable if extreme quality control in the heat treating process is observed. The results did show that there was no statistical difference between the AISI 9310 gears and the Modified Vasco X-2 gears subjected to the heat treatment by Boeing Vertol with a confidence number of 80. The results generated by the Monte Carlo method and the L_{10} curve fit equations were in agreement with this number for all five methods. Also the original results showed there was a statistical difference between the AISI 9310 gears and the Modified Vasco X-2 gears subjected to the heat treatments by NASA, and Curtis-Wright, both with a confidence number of 99. This was also in agreement with the results generated by the Monte Carlo and L_{10} curve fit equations methods. All five methods gave a confidence number of 100.

Preventive Maintenance

Another reason for the need of fatigue studies is for warranty or preventive maintenance information. If a company studies the fatigue of one of their components, they will know how long to warranty that component. A company can determine a warranty based on the failure analysis of their part, this means that on average their part fails within that frame time. If someone happens to buy a part that falls below the mean, then they are covered and can get a new part. Likewise, if their part outlasts the warranty then they know they got their money's worth because the part is above average and may continue to work for a long time.

Similar to determining warranties, fatigue testing is used to determine preventive maintenance schedules. If you do not know how long a component lasts, then how do you know when to change it out? Fatigue testing allows for determining theses schedules with very good accuracy, depending on the application and what percentage of failure is allowed.

When fatigue tests are performed, typically only between ten and twenty samples are tested due to high cost and testing time. It was demonstrated in (Vlcek, Zaretsky and Hendricks, Test Population Selection From Weibull-Based, Monte Carlo Simulations of Fatigue Life 2008) that for a 30 percent variability in fatigue life data at least 30 to 35 samples must be tested. Any more samples will give better results, however, even with 200 samples, there is still a 15 percent variation from maximum life to minimum life (Vlcek, Zaretsky and Hendricks, Test Population Selection From Weibull-Based, Monte Carlo Simulations of Fatigue Life 2008). People have fallen victim to simple sample tests to predict what a population will do. It has been standard practice to perform these tests and acquire the data and then simply take an average of the data and then that will be the resultant number. This method of testing has been proven to not be sufficient. When only the average of a sample is taken in to account, certain aspects of the data are missed. Another misconception in reading data is to graph it and make assumptions visually. Graphs are made to be read and to put data into visual perspective. It is very easy to over look the fact that if you were to enlarge the graph, the data points may be lying directly on top of each other, which would indicate no statistical difference. These aspects of data collection are taken into account with Johnson's methodology. A test of ten bearings in a bin of 1,000 could potentially give unsatisfactory data. The strongest ten in a box could be picked and then it could be assumed that the rest are just as strong, when in reality, if they are put into production components, they could have disastrous effects.

Summary of Results

There were five models developed to analyze out-life suspensions in fatigue data. It was shown that methods 1 through 4 were in relatively close agreement with the experimental results from the literature that were determined graphically. Method 5 was not in agreement. This is believed to be because the number of samples used in the calculations was very small compared to the range of lives given, and there was too much variability in the randomness to notice any trends.

Chapter 5

Conclusion

Fatigue

The goal of a mechanical engineer is to design components that last and are safe to use. To determine whether or not a specific component or material will last, it must be tested. One such test is a fatigue test. This will give the engineer an idea of how long a component will last under certain circumstances. The problem with fatigue data is that it is probabilistic and not deterministic. There is no way to determine exactly when something will fail, however, there exist methods for predicting when a component will fail.

The first was developed by Waloddi Weibull in the 1950’s. He developed a method for statistically predicting failure. Following Weibull’s method, Leonard Johnson in the 1960’s developed a method to determine whether or not one material was statistically different from another. This method was widely used by engineers; however, it was difficult to calculate due to the limited graphs he published. This method was taken a step further by Zaretsky, Hendricks, and Vlcek. They took Johnsons charts and graphs and developed equations to fit them. This allowed for ease of use to calculate a confidence number. From there, a Monte Carlo simulation was written using the equations to expand on the experimental data.

Vlcek, Zaretsky, and Hendricks demonstrated the concept of a “bin” model which was the basis of the Monte Carlo simulations. In this “bin” model it is assumed that there is a population of failed samples, the exact lives of these samples is not known, however, it is known in what order they failed. This order can be ranked from 0 to 1 and, using the parameters of the Weibull equation, it is possible to calculate a life for each sample.

The experimental data (Weibull slope, and characteristic life) would be used as inputs into the program and the program would run thousands of simulations of the original experiment. This data would then either agree or disagree with the original experiment. This allows engineers to have valuable data yet not spend a lot of money and time on experimental testing.

These simulations, however, were limited. Not all fatigue tests are run to failure. Some tests get stopped for numerous reasons. They may get stopped because the engineer has a specific cut off point that the specimen does not need to exceed, or the power in the building could go out. Regardless of the reason for the test to stop, it is still valuable data.

This new simulation takes these suspensions into account and gives a statistically determined confidence number.

Method

The purpose of this research was to develop a model that would allow the statistical validation of a confidence number for fatigue tests comparing two materials in which out-life suspensions are present. The method used was a Monte Carlo simulation in which random numbers are generated, ranked, and then converted to lives using the Weibull equation (equation 1). This method of analyzing fatigue results has been used in the past, but not to incorporate suspensions. The goal of the simulations is to cut down on cost and time of fatigue tests while still having statistically accurate data.

Five methods were developed and then compared to known data sets for validation. The first method was designed to treat the data as if the failure index was that all samples of both populations failed. This did not directly take into account suspensions but was used for comparison purposes. This provided some insight into what possibly could have happened had all samples failed.

The second method was designed to incorporate only failed samples in the calculations. The failure index of method two was the same as the failure index in the original experiment. This again did not directly take suspensions into account in the simulation, but did represent what should be calculated as if done graphically. When calculating a confidence number graphically, all that is available is the failed samples, and this is what method 2 represented.

In method 3, suspensions were actually generated. The number of attempted samples was the number of generated lives; however, the number of samples failed was the number of generated lives used in the calculations. There was a cutoff sample size incorporated into the simulation. In generating all the lives it was assumed that there were out-lives not being used in the calculations. According to the graphs of the original data, it appeared that none of the samples passed a certain life and this lead to the assumption of out-lives.

Method 4 also made use of generating suspended lives and then used a cutoff to calculate confidence numbers. This time, however, instead of sample size being the cutoff, a particular life was the cutoff. Again, the number of lives generated was equal to the number of samples attempted, yet this time the number of samples used in calculating confidence numbers varied because the number of samples was determined by how many fell within the range of the cutoff life.

Method 5 was designed to be a hybrid of both methods 3 and 4. The samples generated to be used in the calculations were forced to a certain number of samples as well as fall within a certain range of lives.

Results

The results of this model of comparing two fatigue data sets containing suspensions are repeated in tables 23 and 24.

Table 23. Results of comparing Modified Vasco X-2 and AISI 9310 rolling contact fatigue data

Curve fit confidence numbers vs. Monte Carlo counted confidence numbers based on L10 lives

Material		Method					Experimental
		1	2	3	4	5	
Modified Vasco X-2 vs. AISI 9310	Curve fit	83	79	79	79	74	84
	MC	92	78	92	92	52	

Table 24. Results of comparing Modified Vasco X-2 and AISI 9310 gear fatigue data

Monte Carlo Counted Confidence Numbers

Material	Heat Treatment	Method					Experimental
		1	2	3	4	5	
AISI 9310	----	----	----	----	----	----	----
Modified Vasco X-2	Boeing Vertol	77	68	77	76	52	80
	NASA	100	100	100	100	100	99
	Curtis-Wright	100	----	----	----	----	99

The confidence numbers generated by the simulation, both the Monte Carlo numbers and the curve fit numbers were in agreement with the original graphical confidence numbers showing whether two materials were statistically different or not.

Recommendations for Further Study

The next step for incorporating suspensions into a Monte Carlo simulation would be to reproduce suspensions that are contained within the data. That is suspensions that are not out-lives. A simulation could be written to use the same bin model and counting method to determine a confidence number. The user could input how many suspensions there are and then let the program randomly pick which samples to suspend. This could be repeated to show trends.

This current model could also be adjusted to count the L_{50} and mean lives. The standard is to use the L_{10} life but this could still show trends. If a populations L_{50} or mean life is significantly greater this could lead to a conclusion to repeat the experiment if possible to see if trends repeat.

Another possible future study would be to perform an experiment with fatigue samples and specify the life at which to use as a cut-off out-life. This way it would be known exactly why the suspensions occurred and it would be a more accurate way of validating the simulation.

References

Askeland, Donald R., and Pradeep P. Phule. *The Science and Engineering of Materials.* Toronto: Thomson, 2006.

ASTM. "Standard Practive for Statistical Analysis of Linear or Linearized Stress-Life (S-N) and Strain-Life (ε-N) fatigue Data." *ASTM E739-91*, 1998.

Enguo, Dong, Zhang Lei, and Xing Yanyum. "Robustness analysis for planetary gear based on Monte Carlo method." *2010 WASE International Conference on Information Engineering.* IEEE, 2010. 212-215.

Haldar, Achintya, and Sankaran Mahadevan. *Probability, Reliability, and Statistical Methods in Engineering Design.* New York: John Wiley & Sons, 2000.

Holland, Frederic A., and Erwin V. Zaretsky. *Investigation of Weibull Statistics in Fracture Analysis of Cast Aluminum.* Technical Memorandum, Cleveland: NASA, 1989.

Johnson, Leonard G. *The Statistical Treatment of Fatigue Experiments.* New York: Elsevier Publishing Company, 1964.

—. *Theory and Technique of Variation Research.* New York: Elsevier Publishing Company, 1964.

Kalpakjian, Serope, and Steven R. Schmid. *Manufacturing Engineering and Technology.* Upper Saddle River: Pearson Prentice Hall, 2006.

McBride, Jacob. *Ranking of Fatigue Data Based Upon Monte Carlo Simulated Confidence Number Figures.* Thesis, Statesboro: Georgia Southern University, 2011.

Naylor, T. J. *Computer Simulation Techniques.* New York: Wiley, 1966.

Norton, Robert L. *Machine Design An Integrated Approach.* Upper Saddle River: Pearson Prentice Hall, 2006.

Oswald, Fred B., Timothy R. Jett, Roamer E. Predmore, and Erwin V. Zaretsky. "Probabilistic Analysis of Space Shuttle Body Flap Actuator Ball Bearings." *Tribology Transactions*, 2008: 193-203.

Rankine, W. "On the Causes of Unexpected Breakage of Journals of railway Axles." *Minutes of the Proceedings*, 1843: 105-107.

Restrepo, Carolina, and John E. Hurtado. "Pattern Recognition for a Flight Dynamics Monte Carlo Simulation." *American Institute of Aeronautics and Astronautics.*

Rubinstein, Reuven Y. *Simulation and the Monte Carlo Method.* New York: John Wiley & Sons, 1981.

Sutherland, Herbert J., and Paul S. Veers. "The Development of Confidence Limits for Fatigue Strength Data." *Wind Energy 2000, ASME/AIAA.* 2000.

Tevaarwerk, J. L. "Applied Tribology - Reliability, Weibull Analysis, Reliability Testing, and Reliability Prediction." October 3, 2002.

Townsend, Dennis P., and Erwin V. Zaretsky. *Comparisons of Modified Vasco X-2 and AISI 9310 Gear Steels.* Technical Paper, Cleveland: NASA, 1980.

Townsend, Dennis P., Erwin V. Zaretsky, and Neil E. Anderson. *Comparison of Modified Vasco X-2 with AISI 9310 - Preliminary Report.* Technical Memorandum, Cleveland: NASA, 1977.

Vlcek, Brian L., Erwin V. Zaretsky, and Robert C. Hendricks. "Test Population Selection From Weibull-Based, Monte Carlo Simulations of Fatigue Life." *American Institute of Aeronautics and Astronautics*, 2008.

Vlcek, Brian L., Robert C. Hendricks, and Erwin V. Zaretsky. *Determination of Rolling-Element Fatigue Life From Computer Generated Bearing Tests.* TM, Cleveland: NASA, 2003.

Vlcek, Brian L., Robert C. Hendricks, and Erwin V. Zaretsky. *Predictive Failure of Cylindrical Coatings Using Weibull Analysis.* TM, Cleveland: NASA, 2002.

—. "Probabilistic Analysis for Comparing Fatigue Data Based on Johnson-Weibull Parameters." *ASME International Design Engineering Technical Conference.* Las Vegas: ASME, 2007.

—. "Relative Ranking of Fatigue Lives of Rotating Aluminum Shafts Using L10 Weibull-Johnson Confidence Numbers." *The 12th International Symposium on Transport Phenomena and Dynamics of Rotation Machinery.* Honalulu, 2008.

Vlcek, Brian L., Robert C. Hendricks, Erwin V. Zaretsky, and Noel S. Murray. "Monte Carlo Validation of Weibull-Johnson Confidence Numbers for Relative Ranking of Fatigue Data Sets with Experimental Bearing Validations."

Wasserman, Gary S. *Reliability Verification, Testing, and Analysis in Engineering Design.* New York: Marcel Dekker, Inc., 2003.

Weibull, Waloddi. "A Statistical Distribution Function of Wide Applicability." *Journal of Applied Mechanics*, 1951: 293-297.

Wireman, Terry. *Preventative Maintenance.* New York: Industrial Press Inc., 2008.

www.directindustry.com. http://www.directindustry.com/prod/instron/servohydraulic-fatigue-testing-machines-18463-428354.html (accessed 2012).

www.pci-pcmcia-express.com. http://www.pci-pcmcia-express.com/P75240-ROTATING-BEAM-FATIGUE-TESTING-MAC (accessed 2012).

Zaretsky, Erwin V. "Design for life, plan for death." *Machine Design*, August 8, 1994: 57-59.
—. *STLE Life Factors for Rolling Bearings.* Park Ridge, IL: STLE, 1992.

Appendix

Appendix A: Computer Simulation for Method 1

```
Sub Macro1()
'
' Macro1 Macro
'
' Keyboard Shortcut: Ctrl+e
'

Worksheets(1).Select
Application.ScreenUpdating = False 'to keep screen from constanly updating and slowing down
simulation

trialsB = Cells(13, 2) 'number of times to run simulation, can be from 1 up

taz = 87 'this is for the index to copy the summary page and index the whole thing down 87 cells
shuttleb = 3

For loop37 = 1 To trialsB 'loop that runs entire program
Worksheets(1).Select

Column = 2 'indexs the summary page for all values, moves to right after every loop
countL10 = 0 'these counts are used to count which is bigger for comparison on summary page
countL10B = 0

countmean = 0
countmeanB = 0


'This is used in the code for archiving the data
   shuttle = 5


trials = Cells(12, 2) 'number of times one run is repeated, this value is typically 100 to get a
confidence number out of 100

For loop10 = 1 To trials  'within this loop all the numbers for one run get calculated and copied
to the summary page to be compared after 100 runs

alpha = Cells(9, 2) 'size of bin A
alpha2 = Cells(10, 2)  'size of bin B


If alpha > alpha2 Then  'this is just to number the samples from 1 to which ever bin is bigger

   x = 7
```

```
   For loop3 = 1 To alpha
   Cells(x, 3) = x - 6
   x = x + 1
   Next loop3
   beta = alpha          'beta is used in the loop that archives the numbers, and for clearing contents
on sheet1
      Else
      x = 7
      For loop3 = 1 To alpha2
      Cells(x, 3) = x - 6
      x = x + 1
      Next loop3

   beta = alpha2
   End If




'Bin A
'generating random number
alpha = Cells(9, 2)
r = 7
num1 = 1
For loop1 = 1 To alpha
   Cells(r, 4).Select
here2:
   ActiveCell.FormulaR1C1 = "=randbetween(1,1000)"

'Checking for duplicate
s = 6
For randcheck = 1 To num1
   If Cells(r, 4) = Cells(s, 4) Then GoTo here2
   s = s + 1
Next randcheck
   r = r + 1
   num1 = num1 + 1
Next loop1

'sorting column
      ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
   ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Cells(7, 4), _
      SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
   With ActiveWorkbook.Worksheets("Sheet1").Sort
      .SetRange Range(Cells(7, 4), Cells(alpha + 6, 4))
```

```
        .Header = xlNo
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With

'Bin B
'random number
alpha2 = Cells(10, 2)
r = 7
num12 = 1
For loop2 = 1 To alpha2
    Cells(r, 12).Select
here4:
    ActiveCell.FormulaR1C1 = "=RANDBETWEEN(1,1000)"

'Checking for duplicate
s = 6
For randcheck2 = 1 To num12
    If Cells(r, 12) = Cells(s, 12) Then GoTo here4
    s = s + 1
Next randcheck2
    r = r + 1
    num12 = num12 + 1
Next loop2

' Sorting column
    ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Cells(7, 12), _
        SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
    With ActiveWorkbook.Worksheets("Sheet1").Sort
        .SetRange Range(Cells(7, 12), Cells(alpha2 + 6, 12))
        .Header = xlNo
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With




    r1 = 7
```

```
For loop5 = 1 To alpha
'Rank A
   Cells(r1, 5) = (Cells(r1, 4) - 0.3) / (1000 + 0.4) 'ok to keep 1000 because bin is from 1 to 1000
'S of A
   Cells(r1, 6) = 1 - Cells(r1, 5)
   SA = Cells(r1, 6)
'Life of A
      A1 = Application.WorksheetFunction.Ln(1 / SA)
      B1 = Application.WorksheetFunction.Ln(A1)
      CA = Exp(B1 / Cells(3, 2)) * Cells(6, 2)
      Cells(r1, 7) = CA
   r1 = r1 + 1
Next loop5

   r1 = 7


For loop6 = 1 To alpha2
'Rank B
   Cells(r1, 13) = (Cells(r1, 12) - 0.3) / (1000 + 0.4)
'S of B
   Cells(r1, 14) = 1 - Cells(r1, 13)
   SB = Cells(r1, 14)
'Life of B
      a2 = Application.WorksheetFunction.Ln(1 / SB)
      B2 = Application.WorksheetFunction.Ln(a2)
      CB = Exp(B2 / Cells(4, 2)) * Cells(7, 2)
      Cells(r1, 15) = CB
   r1 = r1 + 1
Next loop6




r11 = 7

For loop7a = 1 To alpha
'rank for 1 to sample size for bin A
      Z1 = (Cells(r11, 3) - 0.3) / (alpha + 0.4)
      Cells(r11, 8) = Z1

'S for sample size for bin A
      Z2 = 1 - Cells(r11, 8)
      Cells(r11, 9) = Z2
```

```
'lnln(1/S) for sample size for bin A
        Z5 = 1 / Z2
        Z3 = Application.WorksheetFunction.Ln(Z5)
        Z4 = Application.WorksheetFunction.Ln(Z3)
        Cells(r11, 10) = Z4
    r11 = r11 + 1
Next loop7a

r111 = 7

For loop11a = 1 To alpha2
'rank for 1 to sample size for bin B
        Z1b = (Cells(r111, 3) - 0.3) / (alpha2 + 0.4)
        Cells(r111, 16) = Z1b

'S for sample size for bin B
        Z2b = 1 - Cells(r111, 16)
        Cells(r111, 17) = Z2b

'lnln(1/S) for sample size for bin B
        Z5b = 1 / Z2b
        Z3b = Application.WorksheetFunction.Ln(Z5b)
        Z4b = Application.WorksheetFunction.Ln(Z3b)
        Cells(r111, 18) = Z4b
    r111 = r111 + 1
Next loop11a




'lnA
r12 = 7
For loop8 = 1 To alpha
Cells(r12, 11) = Application.WorksheetFunction.Ln(Cells(r12, 7))
r12 = r12 + 1
Next loop8

'lnB
r13 = 7
For loop9 = 1 To alpha2
Cells(r13, 19) = Application.WorksheetFunction.Ln(Cells(r13, 15))
r13 = r13 + 1
Next loop9
```

```
'Organized answers on left of sheet 1, cells B16 to B48
'slope A
MA = Application.WorksheetFunction.LinEst(Range(Cells(7, 10), Cells(alpha + 6, 10)),
Range(Cells(7, 11), Cells(alpha + 6, 11)), True, True)
Cells(16, 2) = MA
MAa = Cells(16, 2)
Slope1A = MAa
'slope B
MB = Application.WorksheetFunction.LinEst(Range(Cells(7, 18), Cells(alpha2 + 6, 18)),
Range(Cells(7, 19), Cells(alpha2 + 6, 19)), True, True)
Cells(24, 2) = MB
MBb = Cells(24, 2)
Slope1B = MBb


'intercepts A and B
Ba = Application.WorksheetFunction.Intercept(Range(Cells(7, 10), Cells(alpha + 6, 10)),
Range(Cells(7, 11), Cells(alpha + 6, 11)))
Bb = Application.WorksheetFunction.Intercept(Range(Cells(7, 18), Cells(alpha2 + 6, 18)),
Range(Cells(7, 19), Cells(alpha2 + 6, 19)))

'Lbeta A calculations
V2 = (Ba / MAa)
VV = -1 * V2
LBa = Exp(VV)

'Lbeta B calculations
V3 = (Bb / MBb)
MO = -1 * V3
LBb = Exp(MO)


'plotting Lbetas
Cells(17, 2) = LBa
Cells(25, 2) = LBb

sassyA1 = LBa
sassyB1 = LBb

'L10 A
L10a = Exp(-2.25037 / MAa) * LBa
Cells(18, 2) = L10a
L10A1 = L10a
'L10 B
```

```
L10b = Exp(-2.25037 / MBb) * LBb
Cells(26, 2) = L10b
L10B1 = L10b

'L50 A
L50a = Exp(-0.36651 / MAa) * LBa
Cells(19, 2) = L50a
L50A1 = L50a
'L50 B
L50b = Exp(-0.36651 / MBb) * LBb
Cells(27, 2) = L50b
L50B1 = L50b

'Mean@ A
meanata = 62.1 * (MAa ^ -0.172)
Cells(20, 2) = meanata
chevyA1 = meanata
'Mean@ B
meanatb = 62.1 * (MBb ^ -0.172)
Cells(28, 2) = meanatb
chevyB1 = meanatb




    'Mean A Gamma function method
    musb = (MAa + 1) / MAa
    a1a = Application.WorksheetFunction.GammaLn(musb)
    LmeanA = LBa * Exp(a1a)
    Cells(21, 2) = LmeanA
    cheeseA1 = LmeanA
    'Mean A
    'D1a = 1 / (1 - (meanata / 100))
    'D2a = Application.WorksheetFunction.Ln(D1a)
    'D3a = Application.WorksheetFunction.Ln(D2a)
    'meanA = (Exp(D3a / MAa)) * LBa
    'Cells(21, 2) = meanA
    'cheeseA1 = meanA

    'Mean B Gamma function method
    musbB = (MBb + 1) / MBb
    a1aB = Application.WorksheetFunction.GammaLn(musbB)
    LMeanAb = LBb * Exp(a1aB)
    Cells(29, 2) = LMeanAb
    cheeseB1 = LMeanAb
```

```
        'Mean B
        'D1b = 1 / (1 - (meanatb / 100))
        'D2b = Application.WorksheetFunction.Ln(D1b)
        'D3b = Application.WorksheetFunction.Ln(D2b)
        'meanB = (Exp(D3b / MBb)) * LBb
        'Cells(29, 2) = meanB
        'cheeseB1 = meanB





'DOF A-B
Cells(40, 2) = (Cells(9, 2) - 1) * (Cells(10, 2) - 1)
dofab = Cells(40, 2)

'CONFIDENCE INTERAL FOR A


    'Ao
    Anot = (-0.0844 / Cells(16, 2)) - 0.05584
    'Bo
    Bnot = (1.2796 / Cells(16, 2)) + 0.6729
    'lnDOF
    lnDOF = Application.WorksheetFunction.Ln(Cells(40, 2))
    'MLR at 99
    MLR99 = (Anot * lnDOF + Bnot) ^ 2 + 1
    Cells(41, 2) = MLR99
    'D
    Dvegas = 3.912 / (MLR99 - 1)
    'MLRexp
    If Cells(21, 2) > Cells(29, 2) Then
    MLRexp = Cells(21, 2) / Cells(29, 2)
    Else
    MLRexp = Cells(29, 2) / Cells(21, 2)
    End If
    'C
    Cvegas = 1 - 0.5 * Exp(-Dvegas * (MLRexp - 1))
    Cells(42, 2) = Cvegas

'CONFIDENCE INTERVAL FOR B

    'AoB
    AnotB = (-0.0844 / Cells(24, 2)) - 0.05584
    'BoB
```

```
   BnotB = (1.2796 / Cells(24, 2)) + 0.6729
   'lnDOF
   lnDOF = Application.WorksheetFunction.Ln(Cells(40, 2))
   'MLR at 99
   MLR99B = (AnotB * lnDOF + BnotB) ^ 2 + 1
   Cells(43, 2) = MLR99B
   'D
   DvegasB = 3.912 / (MLR99B - 1)
   'MLRexp
   If Cells(21, 2) > Cells(29, 2) Then
   MLRexpB = Cells(21, 2) / Cells(29, 2)
   Else
   MLRexpB = Cells(29, 2) / Cells(21, 2)
   End If
   'C
   CvegasB = 1 - 0.5 * Exp(-DvegasB * (MLRexpB - 1))
   Cells(44, 2) = CvegasB

'C average
Cvegasavg = (Cells(42, 2) + Cells(44, 2)) / 2
Cells(45, 2) = Cvegasavg

'L10 dependent confidence numbers

'A
aaa = Exp((4.5286 / Cells(16, 2)) + 0.3152)
'ln(m)
lnmA = Application.WorksheetFunction.Ln(Cells(16, 2))
'B
BBB = 0.29574 * lnmA + (-0.45228)
'L10LR
L10LRA = aaa * Cells(40, 2) ^ BBB
'ao
litanot = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRA = Application.WorksheetFunction.Ln(L10LRA)
'a
lita = litanot / lnL10LRA
   'L10exp
   If Cells(18, 2) > Cells(26, 2) Then
   xnot = Cells(18, 2) / Cells(26, 2)
   Else
   xnot = Cells(26, 2) / Cells(18, 2)
   End If
'ln(xo)
lnxo = Application.WorksheetFunction.Ln(xnot)
```

```
'CL10
CL10 = 1 - Exp(((lita * lnxo) ^ 2 + 2896.3) / -3595.9)
Cells(46, 2) = CL10


'A
AAAb = Exp((4.5286 / Cells(24, 2)) + 0.3152)
'ln(m)
lnmAb = Application.WorksheetFunction.Ln(Cells(24, 2))
'B
BBBb = 0.29574 * lnmAb + (-0.45228)
'L10LR
L10LRAb = AAAb * Cells(40, 2) ^ BBBb
'ao
litanotb = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRAb = Application.WorksheetFunction.Ln(L10LRAb)
'a
litab = litanotb / lnL10LRAb
   'L10exp
   If Cells(18, 2) > Cells(26, 2) Then
   xnotb = Cells(18, 2) / Cells(26, 2)
   Else
   xnotb = Cells(26, 2) / Cells(18, 2)
   End If
'ln(xo)
lnxob = Application.WorksheetFunction.Ln(xnotb)
'CL10
CL10b = 1 - Exp(((litab * lnxob) ^ 2 + 2896.3) / -3595.9)
Cells(47, 2) = CL10b

'C L10 average
CL10avg = (CL10 + CL10b) / 2
Cells(48, 2) = CL10avg




Range("A1").Select


Worksheets("Summary").Select

Cells(2, 1) = loop37 'this was put in to count how many trials got ran incase the program quit
suddenly
```

```
'Slopes
Cells(4, Column) = Slope1A
Cells(12, Column) = Slope1B

'L10
Cells(6, Column) = L10A1
Cells(14, Column) = L10B1

'L50
Cells(7, Column) = L50A1
Cells(15, Column) = L50B1

'Mean
Cells(9, Column) = cheeseA1
Cells(17, Column) = cheeseB1

'Mean@
Cells(8, Column) = chevyA1
Cells(16, Column) = chevyB1

'Lb
Cells(5, Column) = sassyA1
Cells(13, Column) = sassyB1

'DOF
Cells(28, Column) = dofab

'MLR99
Cells(29, Column) = MLR99
Cells(31, Column) = MLR99B

'Cvegas
Cells(30, Column) = Cvegas
Cells(32, Column) = CvegasB

'C average
Cells(33, Column) = Cvegasavg

'CL10
Cells(34, Column) = CL10
Cells(35, Column) = CL10b
Cells(36, Column) = CL10avg

'L10A / L10B
Cells(60, Column) = Cells(6, Column) / Cells(14, Column)
```

```
'MeanA / MeanB
Cells(64, Column) = Cells(9, Column) / Cells(17, Column)

'Counting which is bigger, L10A or L10B
If Cells(60, Column) > 1 Then
countL10 = countL10 + 1
Else
countL10B = countL10B + 1
End If
Cells(68, 2) = countL10
Cells(69, 2) = countL10B

'Counting which is bigger meanA or meanB
If Cells(64, Column) > 1 Then
countmean = countmean + 1
Else
countmeanB = countmeanB + 1
End If
Cells(75, 2) = countmean
Cells(76, 2) = countmeanB




Cells(2, Column) = Column - 1 'just for numbering

Worksheets(1).Select

Cells(5, 3) = loop10 'again just to keep track of how many get run to ensure all get run

'This is for archiving all data onto sheet3
   Range(Cells(5, 3), Cells(beta + 6, 27)).Select
   Selection.Copy
   Worksheets(3).Select
   Cells(shuttle, shuttleb).Select
   ActiveSheet.Paste
   Range("A1").Select
   Worksheets(1).Select
   Range("A1").Select


'Just used this to clear contents before new run and also keep last loop data on sheet 1
If loop10 < trials Then
  Worksheets(1).Select
```

```
    Range(Cells(7, 3), Cells(beta + 6, 27)).Select
    Selection.ClearContents
    Range(Cells(16, 2), Cells(48, 2)).Select
    Selection.ClearContents

Else
End If

Cells(1, 1).Select

Column = Column + 1
shuttle = shuttle + beta + 3

Next loop10


Worksheets("Summary").Select

If Cells(68, 2) > Cells(69, 2) Then 'This was just to display which was greater
Cells(68, 3) = "A>B"
Else
Cells(68, 3) = "B>A"
End If



If Cells(75, 2) > Cells(76, 2) Then
Cells(75, 3) = "A>B"
Else
Cells(75, 3) = "B>A"
End If



Cells(83, 3) = "=average(B36:CW36)" 'averages L10 confidence numbers
Cells(83, 6) = "=average(B33:CW33)" 'averages mean confidence numbers



    Range(Cells(2, 1), Cells(85, 101)).Select 'copies data just generated on summary page down
85 cells to make room for next trials numbers
    Selection.Copy
    Cells(taz, 1).Select
    ActiveSheet.Paste
```

```
    Range("B2:CW85").Select
    Selection.ClearContents

    taz = taz + 85
    shuttleb = shuttleb + 18
Next loop37


Worksheets("Sheet1").Select



trials = Cells(13, 2)


Worksheets("Summary").Select

a = 168
ab = 169
ac = 170
d = 6
b = 3

Cells(2, 3) = Cells(168, 3) 'this starts the curve fit confidence number averaging for the whole
summary sheet


Cells(2, 6) = Cells(168, 6)


For cl10avgavgavgloop = 1 To trials 'averaging the curve fit equations on the summary page
Cells(2, 3) = Cells(2, 3) + Cells(a + 85, 3)
Cells(2, 6) = Cells(2, 6) + Cells(a + 85, 6)
a = a + 85



Next cl10avgavgavgloop
Cells(2, 3) = Cells(2, 3) / trials


Cells(2, 6) = Cells(2, 6) / trials



az = Cells(2, 3)
```

```
ax = Cells(2, 6)

Cells(1, 1).Select

Worksheets("SummaryB").Select

Cells(1, 2) = az 'average CL10 on final summary page

Cells(5, 2) = ax 'average mean confidence number on final summary page

Worksheets("Summary").Select

amcavg = 153
cmcavg = 154
dmcavg = 155
emcavg = 156
fmcavg = 157
gmcavg = 158

For loopmcavg = 1 To trials 'averaging the Monte Carlo numbers

Cells(1, 10) = Cells(amcavg, 2) + Cells(1, 10)
Cells(1, 11) = Cells(cmcavg, 2) + Cells(1, 11)

amcavg = amcavg + 85
cmcavg = cmcavg + 85

Next loopmcavg

bmcavg = Cells(1, 10) / trials
hmcavg = Cells(1, 11) / trials
```

```
Worksheets("SummaryB").Select
Cells(10, 2) = bmcavg 'average of L10a>L10b
Cells(11, 2) = hmcavg 'average of L10b>L10a





Cells(1, 1).Select

'this is where it takes the averages of slope, L10, L50

Worksheets(1).Select
t1 = Cells(12, 2)
trials = Cells(13, 2)


Worksheets(2).Select


aa = 89
cc = 91
dd = 92
ee = 97
ff = 99
gg = 100


For loop2 = 1 To trials

a = Cells(aa, 2)
b = 3
c = Cells(cc, 2)
d = Cells(dd, 2)
e = Cells(ee, 2)
f = Cells(ff, 2)
g = Cells(gg, 2)
   For loop1 = 1 To t1
   a = a + Cells(aa, b)
   c = c + Cells(cc, b)
   d = d + Cells(dd, b)
   e = e + Cells(ee, b)
   f = f + Cells(ff, b)
   g = g + Cells(gg, b)
   b = b + 1
```

```
 Next loop1
   Cells(aa, 104) = a / t1
   Cells(cc, 104) = c / t1
   Cells(dd, 104) = d / t1
   Cells(ee, 104) = e / t1
   Cells(ff, 104) = f / t1
   Cells(gg, 104) = g / t1



aa = aa + 85
cc = cc + 85
dd = dd + 85
ee = ee + 85
ff = ff + 85
gg = gg + 85

Next loop2


aaa = 174
ccc = 176
ddd = 177
eee = 182
fff = 184
ggg = 185



a2 = Cells(89, 104)
c2 = Cells(91, 104)
d2 = Cells(92, 104)
e2 = Cells(97, 104)
f2 = Cells(99, 104)
g2 = Cells(100, 104)


For loop3 = 1 To trials

a2 = a2 + Cells(aaa, 104)
c2 = c2 + Cells(ccc, 104)
d2 = d2 + Cells(ddd, 104)
e2 = e2 + Cells(eee, 104)
f2 = f2 + Cells(fff, 104)
g2 = g2 + Cells(ggg, 104)
```

```
aaa = aaa + 85
ccc = ccc + 85
ddd = ddd + 85
eee = eee + 85
fff = fff + 85
ggg = ggg + 85

Next loop3



 Worksheets(4).Select 'puts the averages of slope, L10, and L50 on final summary page

 Cells(16, 2) = a2 / trials
 Cells(16, 3) = e2 / trials
 Cells(17, 2) = c2 / trials
 Cells(17, 3) = f2 / trials
 Cells(18, 2) = d2 / trials
 Cells(18, 3) = g2 / trials


 Cells(1, 1).Select


End Sub
```

Appendix B: Computer Simulation for Method 2

```
Sub Macro1()
'
' Macro1 Macro
'
' Keyboard Shortcut: Ctrl+e
'

Worksheets(1).Select
Application.ScreenUpdating = False 'to keep screen from constanly updating and slowing down simulation

trialsB = Cells(13, 2) 'number of times to run simulation, can be from 1 up

taz = 87 'this is for the index to copy the summary page and index the whole thing down 87 cells
shuttleb = 3

For loop37 = 1 To trialsB 'loop that runs entire program
Worksheets(1).Select

Column = 2 'indexs the summary page for all values, moves to right after every loop
countL10 = 0 'these counts are used to count which is bigger for comparison on summary page
countL10B = 0

countmean = 0
countmeanB = 0


'This is used in the code for archiving the data
   shuttle = 5


trials = Cells(12, 2) 'number of times one run is repeated, this value is typically 100 to get a confidence number out of 100

For loop10 = 1 To trials  'within this loop all the numbers for one run get calculated and copied to the summary page to be compared after 100 runs

alpha = Cells(9, 2) 'size of bin A
alpha2 = Cells(10, 2)  'size of bin B


If alpha > alpha2 Then  'this is just to number the samples from 1 to which ever bin is bigger

   x = 7
```

```
   For loop3 = 1 To alpha
   Cells(x, 3) = x - 6
   x = x + 1
   Next loop3
   beta = alpha          'beta is used in the loop that archives the numbers, and for clearing contents
on sheet1
      Else
      x = 7
      For loop3 = 1 To alpha2
      Cells(x, 3) = x - 6
      x = x + 1
      Next loop3

   beta = alpha2
   End If




'Bin A
'generating random number
alpha = Cells(9, 2)
r = 7
num1 = 1
For loop1 = 1 To alpha
   Cells(r, 4).Select
here2:
   ActiveCell.FormulaR1C1 = "=randbetween(1,1000)"

'Checking for duplicate
s = 6
For randcheck = 1 To num1
   If Cells(r, 4) = Cells(s, 4) Then GoTo here2
   s = s + 1
Next randcheck
   r = r + 1
   num1 = num1 + 1
Next loop1

'sorting column
      ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
   ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Cells(7, 4), _
      SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
   With ActiveWorkbook.Worksheets("Sheet1").Sort
      .SetRange Range(Cells(7, 4), Cells(alpha + 6, 4))
```

```
        .Header = xlNo
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With

'Bin B
'random number
alpha2 = Cells(10, 2)
r = 7
num12 = 1
For loop2 = 1 To alpha2
    Cells(r, 12).Select
here4:
    ActiveCell.FormulaR1C1 = "=RANDBETWEEN(1,1000)"

'Checking for duplicate
s = 6
For randcheck2 = 1 To num12
    If Cells(r, 12) = Cells(s, 12) Then GoTo here4
    s = s + 1
Next randcheck2
    r = r + 1
    num12 = num12 + 1
Next loop2

' Sorting column
    ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Cells(7, 12), _
        SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
    With ActiveWorkbook.Worksheets("Sheet1").Sort
        .SetRange Range(Cells(7, 12), Cells(alpha2 + 6, 12))
        .Header = xlNo
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With




    r1 = 7
```

```
For loop5 = 1 To alpha
'Rank A
   Cells(r1, 5) = (Cells(r1, 4) - 0.3) / (1000 + 0.4) 'ok to keep 1000 because bin is from 1 to 1000
'S of A
   Cells(r1, 6) = 1 - Cells(r1, 5)
   SA = Cells(r1, 6)
'Life of A
      A1 = Application.WorksheetFunction.Ln(1 / SA)
      B1 = Application.WorksheetFunction.Ln(A1)
      CA = Exp(B1 / Cells(3, 2)) * Cells(6, 2)
      Cells(r1, 7) = CA
   r1 = r1 + 1
Next loop5

   r1 = 7


For loop6 = 1 To alpha2
'Rank B
   Cells(r1, 13) = (Cells(r1, 12) - 0.3) / (1000 + 0.4)
'S of B
   Cells(r1, 14) = 1 - Cells(r1, 13)
   SB = Cells(r1, 14)
'Life of B
      a2 = Application.WorksheetFunction.Ln(1 / SB)
      B2 = Application.WorksheetFunction.Ln(a2)
      CB = Exp(B2 / Cells(4, 2)) * Cells(7, 2)
      Cells(r1, 15) = CB
   r1 = r1 + 1
Next loop6




r11 = 7

For loop7a = 1 To alpha
'rank for 1 to sample size for bin A
      Z1 = (Cells(r11, 3) - 0.3) / (alpha + 0.4)
      Cells(r11, 8) = Z1

'S for sample size for bin A
      Z2 = 1 - Cells(r11, 8)
      Cells(r11, 9) = Z2
```

```
'lnln(1/S) for sample size for bin A
        Z5 = 1 / Z2
        Z3 = Application.WorksheetFunction.Ln(Z5)
        Z4 = Application.WorksheetFunction.Ln(Z3)
        Cells(r11, 10) = Z4
    r11 = r11 + 1
Next loop7a

r111 = 7

For loop11a = 1 To alpha2
'rank for 1 to sample size for bin B
        Z1b = (Cells(r111, 3) - 0.3) / (alpha2 + 0.4)
        Cells(r111, 16) = Z1b

'S for sample size for bin B
        Z2b = 1 - Cells(r111, 16)
        Cells(r111, 17) = Z2b

'lnln(1/S) for sample size for bin B
        Z5b = 1 / Z2b
        Z3b = Application.WorksheetFunction.Ln(Z5b)
        Z4b = Application.WorksheetFunction.Ln(Z3b)
        Cells(r111, 18) = Z4b
    r111 = r111 + 1
Next loop11a




'lnA
r12 = 7
For loop8 = 1 To alpha
Cells(r12, 11) = Application.WorksheetFunction.Ln(Cells(r12, 7))
r12 = r12 + 1
Next loop8

'lnB
r13 = 7
For loop9 = 1 To alpha2
Cells(r13, 19) = Application.WorksheetFunction.Ln(Cells(r13, 15))
r13 = r13 + 1
Next loop9
```

```
'Organized answers on left of sheet 1, cells B16 to B48
'slope A
MA = Application.WorksheetFunction.LinEst(Range(Cells(7, 10), Cells(alpha + 6, 10)),
Range(Cells(7, 11), Cells(alpha + 6, 11)), True, True)
Cells(16, 2) = MA
MAa = Cells(16, 2)
Slope1A = MAa
'slope B
MB = Application.WorksheetFunction.LinEst(Range(Cells(7, 18), Cells(alpha2 + 6, 18)),
Range(Cells(7, 19), Cells(alpha2 + 6, 19)), True, True)
Cells(24, 2) = MB
MBb = Cells(24, 2)
Slope1B = MBb


'intercepts A and B
Ba = Application.WorksheetFunction.Intercept(Range(Cells(7, 10), Cells(alpha + 6, 10)),
Range(Cells(7, 11), Cells(alpha + 6, 11)))
Bb = Application.WorksheetFunction.Intercept(Range(Cells(7, 18), Cells(alpha2 + 6, 18)),
Range(Cells(7, 19), Cells(alpha2 + 6, 19)))

'Lbeta A calculations
V2 = (Ba / MAa)
VV = -1 * V2
LBa = Exp(VV)

'Lbeta B calculations
V3 = (Bb / MBb)
MO = -1 * V3
LBb = Exp(MO)


'plotting Lbetas
Cells(17, 2) = LBa
Cells(25, 2) = LBb

sassyA1 = LBa
sassyB1 = LBb

'L10 A
L10a = Exp(-2.25037 / MAa) * LBa
Cells(18, 2) = L10a
L10A1 = L10a
'L10 B
```

```
L10b = Exp(-2.25037 / MBb) * LBb
Cells(26, 2) = L10b
L10B1 = L10b

'L50 A
L50a = Exp(-0.36651 / MAa) * LBa
Cells(19, 2) = L50a
L50A1 = L50a
'L50 B
L50b = Exp(-0.36651 / MBb) * LBb
Cells(27, 2) = L50b
L50B1 = L50b

'Mean@ A
meanata = 62.1 * (MAa ^ -0.172)
Cells(20, 2) = meanata
chevyA1 = meanata
'Mean@ B
meanatb = 62.1 * (MBb ^ -0.172)
Cells(28, 2) = meanatb
chevyB1 = meanatb




    'Mean A Gamma function method
    musb = (MAa + 1) / MAa
    a1a = Application.WorksheetFunction.GammaLn(musb)
    LmeanA = LBa * Exp(a1a)
    Cells(21, 2) = LmeanA
    cheeseA1 = LmeanA
    'Mean A
    'D1a = 1 / (1 - (meanata / 100))
    'D2a = Application.WorksheetFunction.Ln(D1a)
    'D3a = Application.WorksheetFunction.Ln(D2a)
    'meanA = (Exp(D3a / MAa)) * LBa
    'Cells(21, 2) = meanA
    'cheeseA1 = meanA

    'Mean B Gamma function method
    musbB = (MBb + 1) / MBb
    a1aB = Application.WorksheetFunction.GammaLn(musbB)
    LMeanAb = LBb * Exp(a1aB)
    Cells(29, 2) = LMeanAb
    cheeseB1 = LMeanAb
```

```
    'Mean B
    'D1b = 1 / (1 - (meanatb / 100))
    'D2b = Application.WorksheetFunction.Ln(D1b)
    'D3b = Application.WorksheetFunction.Ln(D2b)
    'meanB = (Exp(D3b / MBb)) * LBb
    'Cells(29, 2) = meanB
    'cheeseB1 = meanB





'DOF A-B
Cells(40, 2) = (Cells(9, 2) - 1) * (Cells(10, 2) - 1)
dofab = Cells(40, 2)

'CONFIDENCE INTERAL FOR A


  'Ao
  Anot = (-0.0844 / Cells(16, 2)) - 0.05584
  'Bo
  Bnot = (1.2796 / Cells(16, 2)) + 0.6729
  'lnDOF
  lnDOF = Application.WorksheetFunction.Ln(Cells(40, 2))
  'MLR at 99
  MLR99 = (Anot * lnDOF + Bnot) ^ 2 + 1
  Cells(41, 2) = MLR99
  'D
  Dvegas = 3.912 / (MLR99 - 1)
  'MLRexp
  If Cells(21, 2) > Cells(29, 2) Then
  MLRexp = Cells(21, 2) / Cells(29, 2)
  Else
  MLRexp = Cells(29, 2) / Cells(21, 2)
  End If
  'C
  Cvegas = 1 - 0.5 * Exp(-Dvegas * (MLRexp - 1))
  Cells(42, 2) = Cvegas

'CONFIDENCE INTERVAL FOR B

  'AoB
  AnotB = (-0.0844 / Cells(24, 2)) - 0.05584
  'BoB
```

```
    BnotB = (1.2796 / Cells(24, 2)) + 0.6729
    'lnDOF
    lnDOF = Application.WorksheetFunction.Ln(Cells(40, 2))
    'MLR at 99
    MLR99B = (AnotB * lnDOF + BnotB) ^ 2 + 1
    Cells(43, 2) = MLR99B
    'D
    DvegasB = 3.912 / (MLR99B - 1)
    'MLRexp
    If Cells(21, 2) > Cells(29, 2) Then
    MLRexpB = Cells(21, 2) / Cells(29, 2)
    Else
    MLRexpB = Cells(29, 2) / Cells(21, 2)
    End If
    'C
    CvegasB = 1 - 0.5 * Exp(-DvegasB * (MLRexpB - 1))
    Cells(44, 2) = CvegasB

'C average
Cvegasavg = (Cells(42, 2) + Cells(44, 2)) / 2
Cells(45, 2) = Cvegasavg

'L10 dependent confidence numbers

'A
aaa = Exp((4.5286 / Cells(16, 2)) + 0.3152)
'ln(m)
lnmA = Application.WorksheetFunction.Ln(Cells(16, 2))
'B
BBB = 0.29574 * lnmA + (-0.45228)
'L10LR
L10LRA = aaa * Cells(40, 2) ^ BBB
'ao
litanot = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRA = Application.WorksheetFunction.Ln(L10LRA)
'a
lita = litanot / lnL10LRA
    'L10exp
    If Cells(18, 2) > Cells(26, 2) Then
    xnot = Cells(18, 2) / Cells(26, 2)
    Else
    xnot = Cells(26, 2) / Cells(18, 2)
    End If
'ln(xo)
lnxo = Application.WorksheetFunction.Ln(xnot)
```

```
'CL10
CL10 = 1 - Exp(((lita * lnxo) ^ 2 + 2896.3) / -3595.9)
Cells(46, 2) = CL10


'A
AAAb = Exp((4.5286 / Cells(24, 2)) + 0.3152)
'ln(m)
lnmAb = Application.WorksheetFunction.Ln(Cells(24, 2))
'B
BBBb = 0.29574 * lnmAb + (-0.45228)
'L10LR
L10LRAb = AAAb * Cells(40, 2) ^ BBBb
'ao
litanotb = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRAb = Application.WorksheetFunction.Ln(L10LRAb)
'a
litab = litanotb / lnL10LRAb
    'L10exp
    If Cells(18, 2) > Cells(26, 2) Then
    xnotb = Cells(18, 2) / Cells(26, 2)
    Else
    xnotb = Cells(26, 2) / Cells(18, 2)
    End If
'ln(xo)
lnxob = Application.WorksheetFunction.Ln(xnotb)
'CL10
CL10b = 1 - Exp(((litab * lnxob) ^ 2 + 2896.3) / -3595.9)
Cells(47, 2) = CL10b

'C L10 average
CL10avg = (CL10 + CL10b) / 2
Cells(48, 2) = CL10avg




Range("A1").Select


Worksheets("Summary").Select

Cells(2, 1) = loop37 'this was put in to count how many trials got ran incase the program quit
suddenly
```

```
'Slopes
Cells(4, Column) = Slope1A
Cells(12, Column) = Slope1B

'L10
Cells(6, Column) = L10A1
Cells(14, Column) = L10B1

'L50
Cells(7, Column) = L50A1
Cells(15, Column) = L50B1

'Mean
Cells(9, Column) = cheeseA1
Cells(17, Column) = cheeseB1

'Mean@
Cells(8, Column) = chevyA1
Cells(16, Column) = chevyB1

'Lb
Cells(5, Column) = sassyA1
Cells(13, Column) = sassyB1

'DOF
Cells(28, Column) = dofab

'MLR99
Cells(29, Column) = MLR99
Cells(31, Column) = MLR99B

'Cvegas
Cells(30, Column) = Cvegas
Cells(32, Column) = CvegasB

'C average
Cells(33, Column) = Cvegasavg

'CL10
Cells(34, Column) = CL10
Cells(35, Column) = CL10b
Cells(36, Column) = CL10avg

'L10A / L10B
Cells(60, Column) = Cells(6, Column) / Cells(14, Column)
```

```
'MeanA / MeanB
Cells(64, Column) = Cells(9, Column) / Cells(17, Column)

'Counting which is bigger, L10A or L10B
If Cells(60, Column) > 1 Then
countL10 = countL10 + 1
Else
countL10B = countL10B + 1
End If
Cells(68, 2) = countL10
Cells(69, 2) = countL10B

'Counting which is bigger meanA or meanB
If Cells(64, Column) > 1 Then
countmean = countmean + 1
Else
countmeanB = countmeanB + 1
End If
Cells(75, 2) = countmean
Cells(76, 2) = countmeanB




Cells(2, Column) = Column - 1 'just for numbering

Worksheets(1).Select

Cells(5, 3) = loop10 'again just to keep track of how many get run to ensure all get run

'This is for archiving all data onto sheet3
    Range(Cells(5, 3), Cells(beta + 6, 27)).Select
    Selection.Copy
    Worksheets(3).Select
    Cells(shuttle, shuttleb).Select
    ActiveSheet.Paste
    Range("A1").Select
    Worksheets(1).Select
    Range("A1").Select


'Just used this to clear contents before new run and also keep last loop data on sheet 1
If loop10 < trials Then
  Worksheets(1).Select
```

```
    Range(Cells(7, 3), Cells(beta + 6, 27)).Select
    Selection.ClearContents
    Range(Cells(16, 2), Cells(48, 2)).Select
    Selection.ClearContents

Else
End If

Cells(1, 1).Select

Column = Column + 1
shuttle = shuttle + beta + 3

Next loop10


Worksheets("Summary").Select

If Cells(68, 2) > Cells(69, 2) Then 'This was just to display which was greater
Cells(68, 3) = "A>B"
Else
Cells(68, 3) = "B>A"
End If



If Cells(75, 2) > Cells(76, 2) Then
Cells(75, 3) = "A>B"
Else
Cells(75, 3) = "B>A"
End If



Cells(83, 3) = "=average(B36:CW36)" 'averages L10 confidence numbers
Cells(83, 6) = "=average(B33:CW33)" 'averages mean confidence numbers




    Range(Cells(2, 1), Cells(85, 101)).Select 'copies data just generated on summary page down
85 cells to make room for next trials numbers
    Selection.Copy
    Cells(taz, 1).Select
    ActiveSheet.Paste
```

```
    Range("B2:CW85").Select
    Selection.ClearContents

    taz = taz + 85
    shuttleb = shuttleb + 18
Next loop37


Worksheets("Sheet1").Select



trials = Cells(13, 2)


Worksheets("Summary").Select

a = 168
ab = 169
ac = 170
d = 6
b = 3

Cells(2, 3) = Cells(168, 3) 'this starts the curve fit confidence number averaging for the whole
summary sheet


Cells(2, 6) = Cells(168, 6)


For cl10avgavgavgloop = 1 To trials 'averaging the curve fit equations on the summary page
Cells(2, 3) = Cells(2, 3) + Cells(a + 85, 3)
Cells(2, 6) = Cells(2, 6) + Cells(a + 85, 6)
a = a + 85



Next cl10avgavgavgloop
Cells(2, 3) = Cells(2, 3) / trials


Cells(2, 6) = Cells(2, 6) / trials



az = Cells(2, 3)
```

```
ax = Cells(2, 6)

Cells(1, 1).Select

Worksheets("SummaryB").Select

Cells(1, 2) = az 'average CL10 on final summary page

Cells(5, 2) = ax 'average mean confidence number on final summary page

Worksheets("Summary").Select

amcavg = 153
cmcavg = 154
dmcavg = 155
emcavg = 156
fmcavg = 157
gmcavg = 158

For loopmcavg = 1 To trials 'averaging the Monte Carlo numbers

Cells(1, 10) = Cells(amcavg, 2) + Cells(1, 10)
Cells(1, 11) = Cells(cmcavg, 2) + Cells(1, 11)

amcavg = amcavg + 85
cmcavg = cmcavg + 85

Next loopmcavg

bmcavg = Cells(1, 10) / trials
hmcavg = Cells(1, 11) / trials
```

```
Worksheets("SummaryB").Select
Cells(10, 2) = bmcavg 'average of L10a>L10b
Cells(11, 2) = hmcavg 'average of L10b>L10a





Cells(1, 1).Select

'this is where it takes the averages of slope, L10, L50

Worksheets(1).Select
t1 = Cells(12, 2)
trials = Cells(13, 2)


Worksheets(2).Select


aa = 89
cc = 91
dd = 92
ee = 97
ff = 99
gg = 100


For loop2 = 1 To trials

a = Cells(aa, 2)
b = 3
c = Cells(cc, 2)
d = Cells(dd, 2)
e = Cells(ee, 2)
f = Cells(ff, 2)
g = Cells(gg, 2)
   For loop1 = 1 To t1
   a = a + Cells(aa, b)
   c = c + Cells(cc, b)
   d = d + Cells(dd, b)
   e = e + Cells(ee, b)
   f = f + Cells(ff, b)
   g = g + Cells(gg, b)
   b = b + 1
```

```
 Next loop1
   Cells(aa, 104) = a / t1
   Cells(cc, 104) = c / t1
   Cells(dd, 104) = d / t1
   Cells(ee, 104) = e / t1
   Cells(ff, 104) = f / t1
   Cells(gg, 104) = g / t1



aa = aa + 85
cc = cc + 85
dd = dd + 85
ee = ee + 85
ff = ff + 85
gg = gg + 85

Next loop2


aaa = 174
ccc = 176
ddd = 177
eee = 182
fff = 184
ggg = 185



a2 = Cells(89, 104)
c2 = Cells(91, 104)
d2 = Cells(92, 104)
e2 = Cells(97, 104)
f2 = Cells(99, 104)
g2 = Cells(100, 104)


For loop3 = 1 To trials

a2 = a2 + Cells(aaa, 104)
c2 = c2 + Cells(ccc, 104)
d2 = d2 + Cells(ddd, 104)
e2 = e2 + Cells(eee, 104)
f2 = f2 + Cells(fff, 104)
g2 = g2 + Cells(ggg, 104)
```

```
aaa = aaa + 85
ccc = ccc + 85
ddd = ddd + 85
eee = eee + 85
fff = fff + 85
ggg = ggg + 85

Next loop3



 Worksheets(4).Select 'puts the averages of slope, L10, and L50 on final summary page

 Cells(16, 2) = a2 / trials
 Cells(16, 3) = e2 / trials
 Cells(17, 2) = c2 / trials
 Cells(17, 3) = f2 / trials
 Cells(18, 2) = d2 / trials
 Cells(18, 3) = g2 / trials


 Cells(1, 1).Select


End Sub
```

Appendix C: Computer Simulation for Method 3

```
Sub Macro1()
'
' Macro1 Macro
'
' Keyboard Shortcut: Ctrl+e
'


Worksheets(1).Select
Application.ScreenUpdating = False 'to keep screen from constantly updating and slowing down simulation

trialsB = Cells(13, 2) 'amount of times 100 trials are run

taz = 87 'index to copy the summary page and index the whole thing down 87 cells
shuttleb = 3

For loop37 = 1 To trialsB 'runs trials for however many times its specified in trialsB cell
Worksheets(1).Select

Column = 2 'indexs the summary page for all values, moves to right after every loop
countL10 = 0 'used to count which is bigger for comparison on summary page
countL10B = 0

countmean = 0
countmeanB = 0

'this was used in the code for arching the data, as of now not used
   shuttle = 5


trials = Cells(12, 2) 'usually 100 to represent how many times out of 100 something will occur

For loop10 = 1 To trials 'main loop

alpha = Cells(9, 2) 'this is how many samples are in bin A
alpha2 = Cells(10, 2) 'this is how many samples are in bin B


If alpha > alpha2 Then 'this just puts the bold numbers in column C for organizational purposes

   x = 7
   For loop3 = 1 To alpha
   Cells(x, 3) = x - 6
   x = x + 1
```

```
    Next loop3
    beta = alpha            'beta is used in the loop that archives the numbers, and clears sheet1
        Else
        x = 7
        For loop3 = 1 To alpha2
        Cells(x, 3) = x - 6
        x = x + 1
        Next loop3

    beta = alpha2
    End If



'Bin A
'generating random number
alpha = Cells(9, 2)
r = 7
num1 = 1
For loop1 = 1 To alpha
    Cells(r, 4).Select
here2:
    ActiveCell.FormulaR1C1 = "=randbetween(1,1000)"

'Checking for duplicate
s = 6
For randcheck = 1 To num1
    If Cells(r, 4) = Cells(s, 4) Then GoTo here2
    s = s + 1
Next randcheck
    r = r + 1
    num1 = num1 + 1
Next loop1

'sorting column
        ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Cells(7, 4), _
        SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
    With ActiveWorkbook.Worksheets("Sheet1").Sort
        .SetRange Range(Cells(7, 4), Cells(alpha + 6, 4))
        .Header = xlNo
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With
```

```
'Bin B
'random number
alpha2 = Cells(10, 2)
r = 7
num12 = 1
For loop2 = 1 To alpha2
   Cells(r, 12).Select
here4:
   ActiveCell.FormulaR1C1 = "=RANDBETWEEN(1,1000)"

'Checking for duplicate
s = 6
For randcheck2 = 1 To num12
   If Cells(r, 12) = Cells(s, 12) Then GoTo here4
   s = s + 1
Next randcheck2
   r = r + 1
   num12 = num12 + 1
Next loop2

' Sorting column
   ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
   ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Cells(7, 12), _
      SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
   With ActiveWorkbook.Worksheets("Sheet1").Sort
      .SetRange Range(Cells(7, 12), Cells(alpha2 + 6, 12))
      .Header = xlNo
      .MatchCase = False
      .Orientation = xlTopToBottom
      .SortMethod = xlPinYin
      .Apply
   End With


r1 = 7
For loop5 = 1 To alpha
'Rank A
   Cells(r1, 5) = (Cells(r1, 4) - 0.3) / (1000 + 0.4)
'S of A
   Cells(r1, 6) = 1 - Cells(r1, 5)
   SA = Cells(r1, 6)
'Life of A
      A1 = Application.WorksheetFunction.Ln(1 / SA)
      B1 = Application.WorksheetFunction.Ln(A1)
      CA = Exp(B1 / Cells(3, 2)) * Cells(6, 2)
```

```
        Cells(r1, 7) = CA
    r1 = r1 + 1
Next loop5

r1 = 7
For loop6 = 1 To alpha2
'Rank B
    Cells(r1, 13) = (Cells(r1, 12) - 0.3) / (1000 + 0.4)
'S of B
    Cells(r1, 14) = 1 - Cells(r1, 13)
    SB = Cells(r1, 14)
'Life of B
        a2 = Application.WorksheetFunction.Ln(1 / SB)
        B2 = Application.WorksheetFunction.Ln(a2)
        CB = Exp(B2 / Cells(4, 2)) * Cells(7, 2)
        Cells(r1, 15) = CB
    r1 = r1 + 1
Next loop6


r11 = 7
For loop7a = 1 To alpha
'rank for 1 to sample size
        Z1 = (Cells(r11, 3) - 0.3) / (alpha + 0.4)
        Cells(r11, 8) = Z1

'S for sample size
        Z2 = 1 - Cells(r11, 8)
        Cells(r11, 9) = Z2

'lnln(1/S) for sample size
        Z5 = 1 / Z2
        Z3 = Application.WorksheetFunction.Ln(Z5)
        Z4 = Application.WorksheetFunction.Ln(Z3)
        Cells(r11, 10) = Z4
    r11 = r11 + 1
Next loop7a

r111 = 7

For loop11a = 1 To alpha2
'rank for 1 to sample size
        Z1b = (Cells(r111, 3) - 0.3) / (alpha2 + 0.4)
        Cells(r111, 16) = Z1b

'S for sample size
```

```
        Z2b = 1 - Cells(r111, 16)
        Cells(r111, 17) = Z2b

'lnln(1/S) for sample size
        Z5b = 1 / Z2b
        Z3b = Application.WorksheetFunction.Ln(Z5b)
        Z4b = Application.WorksheetFunction.Ln(Z3b)
        Cells(r111, 18) = Z4b
    r111 = r111 + 1
Next loop11a

'lnA
r12 = 7
For loop8 = 1 To alpha
Cells(r12, 11) = Application.WorksheetFunction.Ln(Cells(r12, 7))
r12 = r12 + 1
Next loop8

'lnB
r13 = 7
For loop9 = 1 To alpha2
Cells(r13, 19) = Application.WorksheetFunction.Ln(Cells(r13, 15))
r13 = r13 + 1
Next loop9

'FROM HERE UP NOTHING CHANGES




'ATTENTION, as it sits the code assumes no suspensions or cut offs in bin A, only in B




cutoffnum = Cells(11, 2) 'the number of samples to KEEP
```

```
'Organized answers on left of sheet1, cells B16 to B48
'slope A
MA = Application.WorksheetFunction.LinEst(Range(Cells(7, 10), Cells(alpha + 6, 10)),
Range(Cells(7, 11), Cells(alpha + 6, 11)), True, True) 'original code
'MA = Application.WorksheetFunction.LinEst(Range(Cells(7, 10), Cells(cutoffnum + 6, 10)),
Range(Cells(7, 11), Cells(cutoffnum + 6, 11)), True, True) 'cut off code
Cells(16, 2) = MA
MAa = Cells(16, 2)
Slope1A = MAa
'slope B
'MB = Application.WorksheetFunction.LinEst(Range(Cells(7, 18), Cells(alpha2 + 6, 18)),
Range(Cells(7, 19), Cells(alpha2 + 6, 19)), True, True) original code
MB = Application.WorksheetFunction.LinEst(Range(Cells(7, 18), Cells(cutoffnum + 6, 18)),
Range(Cells(7, 19), Cells(cutoffnum + 6, 19)), True, True) 'cut off code
Cells(24, 2) = MB
MBb = Cells(24, 2)
Slope1B = MBb


'intercepts A and B
Ba = Application.WorksheetFunction.Intercept(Range(Cells(7, 10), Cells(alpha + 6, 10)),
Range(Cells(7, 11), Cells(alpha + 6, 11))) 'original code
'Ba = Application.WorksheetFunction.Intercept(Range(Cells(7, 10), Cells(cutoffnum + 6, 10)),
Range(Cells(7, 11), Cells(cutoffnum + 6, 11))) 'cut off code
'Bb = Application.WorksheetFunction.Intercept(Range(Cells(7, 18), Cells(alpha2 + 6, 18)),
Range(Cells(7, 19), Cells(alpha2 + 6, 19))) original code
Bb = Application.WorksheetFunction.Intercept(Range(Cells(7, 18), Cells(cutoffnum + 6, 18)),
Range(Cells(7, 19), Cells(cutoffnum + 6, 19))) 'cut off code

'Lbeta A calculations
V2 = (Ba / MAa)
VV = -1 * V2
LBa = Exp(VV)

'Lbeta B calculations
V3 = (Bb / MBb)
MO = -1 * V3
LBb = Exp(MO)


'plotting Lbetas
Cells(17, 2) = LBa
Cells(25, 2) = LBb

sassyA1 = LBa
sassyB1 = LBb
```

```
'L10 A
L10a = Exp(-2.25037 / MAa) * LBa
Cells(18, 2) = L10a
L10A1 = L10a
'L10 B
L10b = Exp(-2.25037 / MBb) * LBb
Cells(26, 2) = L10b
L10B1 = L10b

'L50 A
L50a = Exp(-0.36651 / MAa) * LBa
Cells(19, 2) = L50a
L50A1 = L50a
'L50 B
L50b = Exp(-0.36651 / MBb) * LBb
Cells(27, 2) = L50b
L50B1 = L50b

'Mean@ A
meanata = 62.1 * (MAa ^ -0.172)
Cells(20, 2) = meanata
chevyA1 = meanata
'Mean@ B
meanatb = 62.1 * (MBb ^ -0.172)
Cells(28, 2) = meanatb
chevyB1 = meanatb

    'Mean A Gamma function method
    musb = (MAa + 1) / MAa
    a1a = Application.WorksheetFunction.GammaLn(musb)
    LmeanA = LBa * Exp(a1a)
    Cells(21, 2) = LmeanA
    cheeseA1 = LmeanA
    'Mean A
    'D1a = 1 / (1 - (meanata / 100))
    'D2a = Application.WorksheetFunction.Ln(D1a)
    'D3a = Application.WorksheetFunction.Ln(D2a)
    'meanA = (Exp(D3a / MAa)) * LBa
    'Cells(21, 2) = meanA
    'cheeseA1 = meanA

    'Mean B Gamma function method
    musbB = (MBb + 1) / MBb
    a1aB = Application.WorksheetFunction.GammaLn(musbB)
    LMeanAb = LBb * Exp(a1aB)
```

```
    Cells(29, 2) = LMeanAb
    cheeseB1 = LMeanAb
    'Mean B
    'D1b = 1 / (1 - (meanatb / 100))
    'D2b = Application.WorksheetFunction.Ln(D1b)
    'D3b = Application.WorksheetFunction.Ln(D2b)
    'meanB = (Exp(D3b / MBb)) * LBb
    'Cells(29, 2) = meanB
    'cheeseB1 = meanB
```

```
'This is where curve fit equations begin to compute. the equations use generated Weibull slopes, not inputed slopes.
'This incorperates suspension method into curve fit equations.



'DOF A-B
Cells(40, 2) = (Cells(9, 2) - 1) * (Cells(11, 2) - 1)
dofab = Cells(40, 2)

'CONFIDENCE INTERAL FOR A


  'Ao
  Anot = (-0.0844 / Cells(16, 2)) - 0.05584
  'Bo
  Bnot = (1.2796 / Cells(16, 2)) + 0.6729
  'lnDOF
  lnDOF = Application.WorksheetFunction.Ln(Cells(40, 2))
  'MLR at 99
  MLR99 = (Anot * lnDOF + Bnot) ^ 2 + 1
  Cells(41, 2) = MLR99
  'D
  Dvegas = 3.912 / (MLR99 - 1)
  'MLRexp
  If Cells(21, 2) > Cells(29, 2) Then
  MLRexp = Cells(21, 2) / Cells(29, 2)
  Else
  MLRexp = Cells(29, 2) / Cells(21, 2)
  End If
```

```
    'C
    Cvegas = 1 - 0.5 * Exp(-Dvegas * (MLRexp - 1))
    Cells(42, 2) = Cvegas

'CONFIDENCE INTERVAL FOR B

    'AoB
    AnotB = (-0.0844 / Cells(24, 2)) - 0.05584
    'BoB
    BnotB = (1.2796 / Cells(24, 2)) + 0.6729
    'lnDOF
    lnDOF = Application.WorksheetFunction.Ln(Cells(40, 2))
    'MLR at 99
    MLR99B = (AnotB * lnDOF + BnotB) ^ 2 + 1
    Cells(43, 2) = MLR99B
    'D
    DvegasB = 3.912 / (MLR99B - 1)
    'MLRexp
    If Cells(21, 2) > Cells(29, 2) Then
    MLRexpB = Cells(21, 2) / Cells(29, 2)
    Else
    MLRexpB = Cells(29, 2) / Cells(21, 2)
    End If
    'C
    CvegasB = 1 - 0.5 * Exp(-DvegasB * (MLRexpB - 1))
    Cells(44, 2) = CvegasB

'C average
Cvegasavg = (Cells(42, 2) + Cells(44, 2)) / 2
Cells(45, 2) = Cvegasavg

'L10 dependent confidence numbers

'A
aaa = Exp((4.5286 / Cells(16, 2)) + 0.3152)
'ln(m)
lnmA = Application.WorksheetFunction.Ln(Cells(16, 2))
'B
BBB = 0.29574 * lnmA + (-0.45228)
'L10LR
L10LRA = aaa * Cells(40, 2) ^ BBB
'ao
litanot = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRA = Application.WorksheetFunction.Ln(L10LRA)
'a
```

```
lita = litanot / lnL10LRA
    'L10exp
    If Cells(18, 2) > Cells(26, 2) Then
    xnot = Cells(18, 2) / Cells(26, 2)
    Else
    xnot = Cells(26, 2) / Cells(18, 2)
    End If
'ln(xo)
lnxo = Application.WorksheetFunction.Ln(xnot)
'CL10
CL10 = 1 - Exp(((lita * lnxo) ^ 2 + 2896.3) / -3595.9)
Cells(46, 2) = CL10


'A
AAAb = Exp((4.5286 / Cells(24, 2)) + 0.3152)
'ln(m)
lnmAb = Application.WorksheetFunction.Ln(Cells(24, 2))
'B
BBBb = 0.29574 * lnmAb + (-0.45228)
'L10LR
L10LRAb = AAAb * Cells(40, 2) ^ BBBb
'ao
litanotb = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRAb = Application.WorksheetFunction.Ln(L10LRAb)
'a
litab = litanotb / lnL10LRAb
    'L10exp
    If Cells(18, 2) > Cells(26, 2) Then
    xnotb = Cells(18, 2) / Cells(26, 2)
    Else
    xnotb = Cells(26, 2) / Cells(18, 2)
    End If
'ln(xo)
lnxob = Application.WorksheetFunction.Ln(xnotb)
'CL10
CL10b = 1 - Exp(((litab * lnxob) ^ 2 + 2896.3) / -3595.9)
Cells(47, 2) = CL10b

'C L10 average
CL10avg = (CL10 + CL10b) / 2
Cells(48, 2) = CL10avg

Range("A1").Select
```

```
Worksheets("Summary").Select

Cells(2, 1) = loop37 'this tells me how many trialsB were run, its used to make sure all
simulations were run


'Slopes
Cells(4, Column) = Slope1A
Cells(12, Column) = Slope1B

'L10
Cells(6, Column) = L10A1
Cells(14, Column) = L10B1

'L50
Cells(7, Column) = L50A1
Cells(15, Column) = L50B1

'Mean
Cells(9, Column) = cheeseA1
Cells(17, Column) = cheeseB1

'Mean@
Cells(8, Column) = chevyA1
Cells(16, Column) = chevyB1

'Lb
Cells(5, Column) = sassyA1
Cells(13, Column) = sassyB1

'DOF
Cells(28, Column) = dofab

'MLR99
Cells(29, Column) = MLR99
Cells(31, Column) = MLR99B

'Cvegas
Cells(30, Column) = Cvegas
Cells(32, Column) = CvegasB

'C average
Cells(33, Column) = Cvegasavg

'CL10
```

```
Cells(34, Column) = CL10
Cells(35, Column) = CL10b
Cells(36, Column) = CL10avg



'L10A / L10B
Cells(60, Column) = Cells(6, Column) / Cells(14, Column)


'MeanA / MeanB
Cells(64, Column) = Cells(9, Column) / Cells(17, Column)


'Counting which is bigger, L10A or L10B
If Cells(60, Column) > 1 Then
countL10 = countL10 + 1
Else
countL10B = countL10B + 1
End If
Cells(68, 2) = countL10
Cells(69, 2) = countL10B


'Counting which is bigger meanA or meanB
If Cells(64, Column) > 1 Then
countmean = countmean + 1
Else
countmeanB = countmeanB + 1
End If
Cells(75, 2) = countmean
Cells(76, 2) = countmeanB


Cells(2, Column) = Column - 1

Worksheets(1).Select

Cells(5, 3) = loop10


'This is used for archiving the data
    'Range(Cells(5, 3), Cells(beta + 6, 27)).Select
    'Selection.Copy
    'Worksheets(3).Select
    'Cells(shuttle, shuttleb).Select
```

```
    'ActiveSheet.Paste
    'Range("A1").Select
    'Worksheets(1).Select
    'Range("A1").Select


'Just used this to keep last loop data on sheet 1
If loop10 < trials Then
  Worksheets(1).Select
    Range(Cells(7, 3), Cells(beta + 6, 27)).Select
    Selection.ClearContents
    Range(Cells(16, 2), Cells(70, 2)).Select
    Selection.ClearContents

Else
End If
Cells(1, 1).Select

Column = Column + 1
shuttle = shuttle + beta + 3

Next loop10


Worksheets("Summary").Select

If Cells(68, 2) > Cells(69, 2) Then  'This was just to display which was greater
Cells(68, 3) = "A>B"
Else
Cells(68, 3) = "B>A"
End If



If Cells(75, 2) > Cells(76, 2) Then
Cells(75, 3) = "A>B"
Else
Cells(75, 3) = "B>A"
End If



Cells(83, 3) = "=average(B36:CW36)"  'averages L10 confidence numbers


Cells(83, 6) = "=average(B33:CW33)"  'averages mean confidence numbers
```

```
    Range(Cells(2, 1), Cells(85, 101)).Select   'copies data just generated on summary page down
85 cells to make room for next trials numbers
    Selection.Copy
    Cells(taz, 1).Select
    ActiveSheet.Paste
    Range("B2:CW85").Select
    Selection.ClearContents

    taz = taz + 85
    shuttleb = shuttleb + 18

Next loop37


Worksheets("Sheet1").Select

trials = Cells(13, 2)

Worksheets("Summary").Select

a = 168
ab = 169
ac = 170
d = 6
b = 3

Cells(2, 3) = Cells(168, 3)  'this starts the curve fit confidence number averaging for the whole
summary sheet

Cells(2, 6) = Cells(168, 6)


For cl10avgavgavgloop = 1 To trials 'averaging the curve fit equations on the summary page
Cells(2, 3) = Cells(2, 3) + Cells(a + 85, 3)
Cells(2, 6) = Cells(2, 6) + Cells(a + 85, 6)
a = a + 85



Next cl10avgavgavgloop
Cells(2, 3) = Cells(2, 3) / trials


Cells(2, 6) = Cells(2, 6) / trials
```

```
az = Cells(2, 3)
ax = Cells(2, 6)
Cells(1, 1).Select


Worksheets("SummaryB").Select

Cells(1, 2) = az  'average CL10 on final summary page

Cells(5, 2) = ax  'average mean confidence number on final summary page



Worksheets("Summary").Select

amcavg = 153
cmcavg = 154
dmcavg = 155
emcavg = 156
fmcavg = 157
gmcavg = 158


For loopmcavg = 1 To trials  'averaging the Monte Carlo counting numbers


Cells(1, 10) = Cells(amcavg, 2) + Cells(1, 10)
Cells(1, 11) = Cells(cmcavg, 2) + Cells(1, 11)

amcavg = amcavg + 85
cmcavg = cmcavg + 85

Next loopmcavg


bmcavg = Cells(1, 10) / trials
hmcavg = Cells(1, 11) / trials


Worksheets("SummaryB").Select
Cells(10, 2) = bmcavg  'average of L10a>L10b
Cells(11, 2) = hmcavg  'average of L10b>L10a


Cells(1, 1).Select
```

```
'this is where it takes the averages of slope, L10, L50

Worksheets(1).Select
t1 = Cells(12, 2)
trials = Cells(13, 2)


Worksheets(2).Select


aa = 89
cc = 91
dd = 92
ee = 97
ff = 99
gg = 100


For loop2 = 1 To trials

a = Cells(aa, 2)
b = 3
c = Cells(cc, 2)
d = Cells(dd, 2)
e = Cells(ee, 2)
f = Cells(ff, 2)
g = Cells(gg, 2)
    For loop1 = 1 To t1
    a = a + Cells(aa, b)
    c = c + Cells(cc, b)
    d = d + Cells(dd, b)
    e = e + Cells(ee, b)
    f = f + Cells(ff, b)
    g = g + Cells(gg, b)
    b = b + 1
     Next loop1
    Cells(aa, 104) = a / t1
    Cells(cc, 104) = c / t1
    Cells(dd, 104) = d / t1
    Cells(ee, 104) = e / t1
    Cells(ff, 104) = f / t1
    Cells(gg, 104) = g / t1
```

```
aa = aa + 85
cc = cc + 85
dd = dd + 85
ee = ee + 85
ff = ff + 85
gg = gg + 85

Next loop2


aaa = 174
ccc = 176
ddd = 177
eee = 182
fff = 184
ggg = 185



a2 = Cells(89, 104)
c2 = Cells(91, 104)
d2 = Cells(92, 104)
e2 = Cells(97, 104)
f2 = Cells(99, 104)
g2 = Cells(100, 104)


For loop3 = 1 To trials

a2 = a2 + Cells(aaa, 104)
c2 = c2 + Cells(ccc, 104)
d2 = d2 + Cells(ddd, 104)
e2 = e2 + Cells(eee, 104)
f2 = f2 + Cells(fff, 104)
g2 = g2 + Cells(ggg, 104)

aaa = aaa + 85
ccc = ccc + 85
ddd = ddd + 85
eee = eee + 85
fff = fff + 85
ggg = ggg + 85

Next loop3
```

```
Worksheets(4).Select  'puts the averages of slope, L10, and L50 on final summary page

Cells(16, 2) = a2 / trials
Cells(16, 3) = e2 / trials
Cells(17, 2) = c2 / trials
Cells(17, 3) = f2 / trials
Cells(18, 2) = d2 / trials
Cells(18, 3) = g2 / trials


Cells(1, 1).Select


End Sub
```

Appendix D: Computer Simulation for Method 4

```
Sub Macro1()
'
' Macro1 Macro
'
' Keyboard Shortcut: Ctrl+r
'

Worksheets(1).Select
Application.ScreenUpdating = False  'to keep screen from constantly updating an slowing down the simulation

trialsB = Cells(13, 2)  'number of times to run simulation, can be from 1 up

taz = 87 'this is for the index to copy the summary page and index all the data down 87 cells
rowspace = 5 'used to index the copied data from sheet1 to sheet3 down for next run
For loop37 = 1 To trialsB 'loop that runs entire program
Worksheets(1).Select

Column = 2 'indexes the summary page for all values, moves to right after every loop
countL10 = 0 'these counts are used to count which is bigger for comparison on summary page
countL10B = 0

countmean = 0
countmeanB = 0


trials = Cells(12, 2) 'number of times one run is repeated, typically 100 to get a confidence number out of 100
columnspace = 3 'used to index the copied data from sheet1 to sheet3 to the right everytime
For loop10 = 1 To trials 'loop for generating numbers for confidence numbers

alpha = Cells(9, 2) 'size of bin A
alpha2 = Cells(10, 2) 'size of bin B


If alpha > alpha2 Then 'to number the samples from 1 to which ever bin is bigger

    x = 7
    For loop3 = 1 To alpha
    Cells(x, 3) = x - 6
    x = x + 1
    Next loop3
    beta = alpha 'beta used for archinving
        Else
```

```
        x = 7
        For loop3 = 1 To alpha2
        Cells(x, 3) = x - 6
        x = x + 1
        Next loop3

    beta = alpha2
    End If




'Bin A
'generating random number
alpha = Cells(9, 2)
r = 7
num1 = 1
For loop1 = 1 To alpha
    Cells(r, 4).Select
here2:
    ActiveCell.FormulaR1C1 = "=randbetween(1,1000)"

'Checking for duplicate
s = 6
For randcheck = 1 To num1
    If Cells(r, 4) = Cells(s, 4) Then GoTo here2
    s = s + 1
Next randcheck
    r = r + 1
    num1 = num1 + 1
Next loop1

'sorting column
        ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Cells(7, 4), _
        SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
    With ActiveWorkbook.Worksheets("Sheet1").Sort
        .SetRange Range(Cells(7, 4), Cells(alpha + 6, 4))
        .Header = xlNo
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With
```

```
here199:



'Bin B
'random number
alpha2 = Cells(10, 2)
r = 7
num12 = 1
For loop2 = 1 To alpha2
   Cells(r, 12).Select
here4:
   ActiveCell.FormulaR1C1 = "=RANDBETWEEN(1,1000)"

'Checking for duplicate
s = 6
For randcheck2 = 1 To num12
   If Cells(r, 12) = Cells(s, 12) Then GoTo here4
   s = s + 1
Next randcheck2
   r = r + 1
   num12 = num12 + 1
Next loop2

' Sorting column
   ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
   ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Cells(7, 12), _
      SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
   With ActiveWorkbook.Worksheets("Sheet1").Sort
      .SetRange Range(Cells(7, 12), Cells(alpha2 + 6, 12))
      .Header = xlNo
      .MatchCase = False
      .Orientation = xlTopToBottom
      .SortMethod = xlPinYin
      .Apply
   End With



   r1 = 7

For loop5 = 1 To alpha
'Rank A
```

```
   Cells(r1, 5) = (Cells(r1, 4) - 0.3) / (1000 + 0.4)
'S of A
   Cells(r1, 6) = 1 - Cells(r1, 5)
   SA = Cells(r1, 6)
'Life of A
      A1 = Application.WorksheetFunction.Ln(1 / SA)
      B1 = Application.WorksheetFunction.Ln(A1)
      CA = Exp(B1 / Cells(3, 2)) * Cells(6, 2)
      Cells(r1, 7) = CA
   r1 = r1 + 1
Next loop5



   r1 = 7

For loop6 = 1 To alpha2
'Rank B
   Cells(r1, 13) = (Cells(r1, 12) - 0.3) / (1000 + 0.4)
'S of B
   Cells(r1, 14) = 1 - Cells(r1, 13)
   SB = Cells(r1, 14)
'Life of B
      a2 = Application.WorksheetFunction.Ln(1 / SB)
      B2 = Application.WorksheetFunction.Ln(a2)
      CB = Exp(B2 / Cells(4, 2)) * Cells(7, 2)
      Cells(r1, 15) = CB

     If CB > Cells(5, 2) Then GoTo here5 'this is where the program terminates when the desired
life is reached

   r1 = r1 + 1
Next loop6



here5:
If Cells(7, 15) > Cells(5, 2) - 1 Then GoTo here199 'this makes sure more then one life is
generated so the program doesnt crash when computing slopes and intercepts

soccer = loop6
If loop6 = Cells(10, 2) + 1 Then soccer = Cells(10, 2)  'incase there are more lives generated than
needed,
                                          'needed because goto here5 causes loop6 to go up one more
```

```
r11 = 7

For loop7a = 1 To alpha
'rank for 1 to sample size
        Z1 = (Cells(r11, 3) - 0.3) / (alpha + 0.4)
        Cells(r11, 8) = Z1

'S for sample size
        Z2 = 1 - Cells(r11, 8)
        Cells(r11, 9) = Z2

'lnln(1/S) for sample size
        Z5 = 1 / Z2
        Z3 = Application.WorksheetFunction.Ln(Z5)
        Z4 = Application.WorksheetFunction.Ln(Z3)
        Cells(r11, 10) = Z4
    r11 = r11 + 1
Next loop7a

r111 = 7

For loop11a = 1 To soccer 'this indexing is needed to cut off loop at right time otherwise could
add 1
'rank for 1 to sample size
        Z1b = (Cells(r111, 3) - 0.3) / (alpha2 + 0.4) 'does this go to alpha2 or what the program
stoped at?
        Cells(r111, 16) = Z1b

'S for sample size
        Z2b = 1 - Cells(r111, 16)
        Cells(r111, 17) = Z2b

'lnln(1/S) for sample size
        Z5b = 1 / Z2b
        Z3b = Application.WorksheetFunction.Ln(Z5b)
        Z4b = Application.WorksheetFunction.Ln(Z3b)
        Cells(r111, 18) = Z4b
    r111 = r111 + 1
Next loop11a




'lnA
r12 = 7
```

```
For loop8 = 1 To alpha
Cells(r12, 11) = Application.WorksheetFunction.Ln(Cells(r12, 7))
r12 = r12 + 1
Next loop8

'lnB
r13 = 7
For loop9 = 1 To soccer
Cells(r13, 19) = Application.WorksheetFunction.Ln(Cells(r13, 15))
r13 = r13 + 1
Next loop9

Cells(2, 4) = soccer - 1 'to chop off the last life that is over the limit
soccer2 = Cells(2, 4) + 6 'this selects the right lives in the sheet

If soccer2 = 7 Then GoTo here199 'this again to make sure more that one gets generated to be
able to compute slopes and intercepts

    Range(Cells(7, 12), Cells(soccer2, 19)).Select 'copies the data to be used to the right in same
sheet
    Selection.Copy
    Cells(7, 22).Select
    ActiveSheet.Paste
    Range("A1").Select

'Cells(5, 3) = loop10   'copies data from sheet1 to sheet3
'Range(Cells(5, 3), Cells(beta + 6, 29)).Select
'Selection.Copy
'Worksheets("Sheet3").Select
'Cells(rowspace, columnspace).Select
'ActiveSheet.Paste


Worksheets("Sheet1").Select


'here down should change




'Organized answers on left of spreadsheet
'slope A
MA = Application.WorksheetFunction.LinEst(Range(Cells(7, 10), Cells(alpha + 6, 10)),
Range(Cells(7, 11), Cells(alpha + 6, 11)), True, True)
Cells(16, 2) = MA
```

```
MAa = Cells(16, 2)
Slope1A = MAa
'slope B
'MB = Application.WorksheetFunction.LinEst(Range(Cells(7, 18), Cells(alpha2 + 6, 18)),
Range(Cells(7, 19), Cells(alpha2 + 6, 19)), True, True) 'original code
MB = Application.WorksheetFunction.LinEst(Range(Cells(7, 28), Cells(soccer2, 28)),
Range(Cells(7, 29), Cells(soccer2, 29)), True, True) 'cutoff code
Cells(24, 2) = MB
MBb = Cells(24, 2)
Slope1B = MBb


'intercepts A and B
Ba = Application.WorksheetFunction.Intercept(Range(Cells(7, 10), Cells(alpha + 6, 10)),
Range(Cells(7, 11), Cells(alpha + 6, 11)))

'Bb = Application.WorksheetFunction.Intercept(Range(Cells(7, 18), Cells(alpha2 + 6, 18)),
Range(Cells(7, 19), Cells(alpha2 + 6, 19))) 'original code
Bb = Application.WorksheetFunction.Intercept(Range(Cells(7, 28), Cells(soccer2, 28)),
Range(Cells(7, 29), Cells(soccer2, 29))) 'cutoff code

'Lbeta A calculations
V2 = (Ba / MAa)
VV = -1 * V2
LBa = Exp(VV)

'Lbeta B calculations
V3 = (Bb / MBb)
MO = -1 * V3
LBb = Exp(MO)


'plotting Lbetas
Cells(17, 2) = LBa
Cells(25, 2) = LBb

sassyA1 = LBa
sassyB1 = LBb

'L10 A
L10a = Exp(-2.25037 / MAa) * LBa
Cells(18, 2) = L10a
L10A1 = L10a
'L10 B
L10b = Exp(-2.25037 / MBb) * LBb
Cells(26, 2) = L10b
```

```
L10B1 = L10b

'L50 A
L50a = Exp(-0.36651 / MAa) * LBa
Cells(19, 2) = L50a
L50A1 = L50a
'L50 B
L50b = Exp(-0.36651 / MBb) * LBb
Cells(27, 2) = L50b
L50B1 = L50b

'Mean@ A
meanata = 62.1 * (MAa ^ -0.172)
Cells(20, 2) = meanata
chevyA1 = meanata
'Mean@ B
meanatb = 62.1 * (MBb ^ -0.172)
Cells(28, 2) = meanatb
chevyB1 = meanatb




    'Mean A Gamma function method
    musb = (MAa + 1) / MAa
    a1a = Application.WorksheetFunction.GammaLn(musb)
    LmeanA = LBa * Exp(a1a)
    Cells(21, 2) = LmeanA
    cheeseA1 = LmeanA
    'Mean A
    'D1a = 1 / (1 - (meanata / 100))
    'D2a = Application.WorksheetFunction.Ln(D1a)
    'D3a = Application.WorksheetFunction.Ln(D2a)
    'meanA = (Exp(D3a / MAa)) * LBa
    'Cells(21, 2) = meanA
    'cheeseA1 = meanA

    'Mean B Gamma function method
    musbB = (MBb + 1) / MBb
    a1aB = Application.WorksheetFunction.GammaLn(musbB)
    LMeanAb = LBb * Exp(a1aB)
    Cells(29, 2) = LMeanAb
    cheeseB1 = LMeanAb
    'Mean B
    'D1b = 1 / (1 - (meanatb / 100))
```

```
    'D2b = Application.WorksheetFunction.Ln(D1b)
    'D3b = Application.WorksheetFunction.Ln(D2b)
    'meanB = (Exp(D3b / MBb)) * LBb
    'Cells(29, 2) = meanB
    'cheeseB1 = meanB
```

```
'DOF A-B
Cells(40, 2) = (Cells(9, 2) - 1) * (soccer2 - 6 - 1) 'degrees of freedom for bin A times cutoff
number
dofab = Cells(40, 2)

'CONFIDENCE INTERAL FOR A


  'Ao
  Anot = (-0.0844 / Cells(16, 2)) - 0.05584
  'Bo
  Bnot = (1.2796 / Cells(16, 2)) + 0.6729
  'lnDOF
  lnDOF = Application.WorksheetFunction.Ln(Cells(40, 2))
  'MLR at 99
  MLR99 = (Anot * lnDOF + Bnot) ^ 2 + 1
  Cells(41, 2) = MLR99
  'D
  Dvegas = 3.912 / (MLR99 - 1)
  'MLRexp
  If Cells(21, 2) > Cells(29, 2) Then
  MLRexp = Cells(21, 2) / Cells(29, 2)
  Else
  MLRexp = Cells(29, 2) / Cells(21, 2)
  End If
  'C
  Cvegas = 1 - 0.5 * Exp(-Dvegas * (MLRexp - 1))
  Cells(42, 2) = Cvegas

'CONFIDENCE INTERVAL FOR B

  'AoB
  AnotB = (-0.0844 / Cells(24, 2)) - 0.05584
  'BoB
  BnotB = (1.2796 / Cells(24, 2)) + 0.6729
```

```
   'lnDOF
   lnDOF = Application.WorksheetFunction.Ln(Cells(40, 2))
   'MLR at 99
   MLR99B = (AnotB * lnDOF + BnotB) ^ 2 + 1
   Cells(43, 2) = MLR99B
   'D
   DvegasB = 3.912 / (MLR99B - 1)
   'MLRexp
   If Cells(21, 2) > Cells(29, 2) Then
   MLRexpB = Cells(21, 2) / Cells(29, 2)
   Else
   MLRexpB = Cells(29, 2) / Cells(21, 2)
   End If
   'C
   CvegasB = 1 - 0.5 * Exp(-DvegasB * (MLRexpB - 1))
   Cells(44, 2) = CvegasB

'C average
Cvegasavg = (Cells(42, 2) + Cells(44, 2)) / 2
Cells(45, 2) = Cvegasavg

'L10 dependent confidence numbers

'A
aaa = Exp((4.5286 / Cells(16, 2)) + 0.3152)
'ln(m)
lnmA = Application.WorksheetFunction.Ln(Cells(16, 2))
'B
bbb = 0.29574 * lnmA + (-0.45228)
'L10LR
L10LRA = aaa * Cells(40, 2) ^ bbb
'ao
litanot = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRA = Application.WorksheetFunction.Ln(L10LRA)
'a
lita = litanot / lnL10LRA
   'L10exp
   If Cells(18, 2) > Cells(26, 2) Then
   xnot = Cells(18, 2) / Cells(26, 2)
   Else
   xnot = Cells(26, 2) / Cells(18, 2)
   End If
'ln(xo)
lnxo = Application.WorksheetFunction.Ln(xnot)
'CL10
```

```
CL10 = 1 - Exp(((lita * lnxo) ^ 2 + 2896.3) / -3595.9)
Cells(46, 2) = CL10


'A
AAAb = Exp((4.5286 / Cells(24, 2)) + 0.3152)
'ln(m)
lnmAb = Application.WorksheetFunction.Ln(Cells(24, 2))
'B
BBBb = 0.29574 * lnmAb + (-0.45228)
'L10LR
L10LRAb = AAAb * Cells(40, 2) ^ BBBb
'ao
litanotb = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRAb = Application.WorksheetFunction.Ln(L10LRAb)
'a
litab = litanotb / lnL10LRAb
    'L10exp
    If Cells(18, 2) > Cells(26, 2) Then
    xnotb = Cells(18, 2) / Cells(26, 2)
    Else
    xnotb = Cells(26, 2) / Cells(18, 2)
    End If
'ln(xo)
lnxob = Application.WorksheetFunction.Ln(xnotb)
'CL10
CL10b = 1 - Exp(((litab * lnxob) ^ 2 + 2896.3) / -3595.9)
Cells(47, 2) = CL10b

'C L10 average
CL10avg = (CL10 + CL10b) / 2
Cells(48, 2) = CL10avg


Cells(38, 2) = soccer2 - 6

Range(Cells(7, 3), Cells(beta + 6, 29)).Select
Selection.ClearContents



Range("A1").Select
```

```
Worksheets("Summary").Select

Cells(2, 1) = loop37


'Slopes
Cells(4, Column) = Slope1A
Cells(12, Column) = Slope1B

'L10
Cells(6, Column) = L10A1
Cells(14, Column) = L10B1

'L50
Cells(7, Column) = L50A1
Cells(15, Column) = L50B1

'Mean
Cells(9, Column) = cheeseA1
Cells(17, Column) = cheeseB1

'Mean@
Cells(8, Column) = chevyA1
Cells(16, Column) = chevyB1

'Lb
Cells(5, Column) = sassyA1
Cells(13, Column) = sassyB1

'DOF
Cells(28, Column) = dofab

Cells(26, Column) = soccer2 - 6


'MLR99
Cells(29, Column) = MLR99
Cells(31, Column) = MLR99B


'Cvegas
Cells(30, Column) = Cvegas
Cells(32, Column) = CvegasB


'C average
```

```
Cells(33, Column) = Cvegasavg


'CL10
Cells(34, Column) = CL10
Cells(35, Column) = CL10b
Cells(36, Column) = CL10avg



'L10A / L10B
Cells(60, Column) = Cells(6, Column) / Cells(14, Column)


'MeanA / MeanB
Cells(64, Column) = Cells(9, Column) / Cells(17, Column)


'Counting which is bigger, L10A or L10B
If Cells(60, Column) > 1 Then
countL10 = countL10 + 1
Else
countL10B = countL10B + 1
End If
Cells(68, 2) = countL10
Cells(69, 2) = countL10B



'Counting which is bigger meanA or meanB
If Cells(64, Column) > 1 Then
countmean = countmean + 1
Else
countmeanB = countmeanB + 1
End If
Cells(75, 2) = countmean
Cells(76, 2) = countmeanB


Cells(2, Column) = Column - 1

Worksheets(1).Select

Cells(1, 1).Select

Column = Column + 1
```

```
columnspace = columnspace + 28

Next loop10


Worksheets("Summary").Select

If Cells(68, 2) > Cells(69, 2) Then  'This was just to display which was greater
Cells(68, 3) = "A>B"
Else
Cells(68, 3) = "B>A"
End If



If Cells(75, 2) > Cells(76, 2) Then
Cells(75, 3) = "A>B"
Else
Cells(75, 3) = "B>A"
End If



Cells(83, 3) = "=average(B36:CW36)"   'averages L10 confidence numbers
Cells(83, 6) = "=average(B33:CW33)"   'averages mean confidence numbers




   Range(Cells(2, 1), Cells(85, 101)).Select 'copies data just generated on summary page down
85 cells to make room for next trials numbers
   Selection.Copy
   Cells(taz, 1).Select
   ActiveSheet.Paste
   Range("B2:CW85").Select
   Selection.ClearContents

   taz = taz + 85
   rowspace = rowspace + beta + 3
Next loop37


Worksheets("Sheet1").Select
```

```
trials = Cells(13, 2)


Worksheets("Summary").Select

a = 168
ab = 169
ac = 170
d = 6
b = 3

Cells(2, 3) = Cells(168, 3)  'this starts the curve fit confidence number averaging for the whole
summary sheet


Cells(2, 6) = Cells(168, 6)


For cl10avgavgavgloop = 1 To trials           'averaging the curve fit equations on the summary
page
Cells(2, 3) = Cells(2, 3) + Cells(a + 85, 3)
Cells(2, 6) = Cells(2, 6) + Cells(a + 85, 6)
a = a + 85



Next cl10avgavgavgloop
Cells(2, 3) = Cells(2, 3) / trials


Cells(2, 6) = Cells(2, 6) / trials



az = Cells(2, 3)


ax = Cells(2, 6)


Cells(1, 1).Select



Worksheets("SummaryB").Select
```

```
Cells(1, 2) = az    'average CL10 on final summary page

Cells(5, 2) = ax    'average mean confidence number on final summary page



Worksheets("Summary").Select

amcavg = 153
cmcavg = 154
dmcavg = 155
emcavg = 156
fmcavg = 157
gmcavg = 158


For loopmcavg = 1 To trials    'averaging the Monte Carlo numbers


Cells(1, 10) = Cells(amcavg, 2) + Cells(1, 10)
Cells(1, 11) = Cells(cmcavg, 2) + Cells(1, 11)


amcavg = amcavg + 85
cmcavg = cmcavg + 85

Next loopmcavg



bmcavg = Cells(1, 10) / trials
hmcavg = Cells(1, 11) / trials





Worksheets("SummaryB").Select
Cells(10, 2) = bmcavg 'average of l10a>L10b
Cells(11, 2) = hmcavg 'average of L10b>L10a
```

```
Cells(1, 1).Select


'this is where it takes the averages of slope, L10, L50

Worksheets(1).Select
t1 = Cells(12, 2)
trials = Cells(13, 2)


Worksheets(2).Select


aa = 89
cc = 91
dd = 92
ee = 97
ff = 99
gg = 100


For loop2 = 1 To trials

a = Cells(aa, 2)
b = 3
c = Cells(cc, 2)
d = Cells(dd, 2)
e = Cells(ee, 2)
f = Cells(ff, 2)
g = Cells(gg, 2)
    For loop1 = 1 To t1
    a = a + Cells(aa, b)
    c = c + Cells(cc, b)
    d = d + Cells(dd, b)
    e = e + Cells(ee, b)
    f = f + Cells(ff, b)
    g = g + Cells(gg, b)
    b = b + 1
     Next loop1
    Cells(aa, 104) = a / t1
    Cells(cc, 104) = c / t1
    Cells(dd, 104) = d / t1
    Cells(ee, 104) = e / t1
    Cells(ff, 104) = f / t1
    Cells(gg, 104) = g / t1
```

```
aa = aa + 85
cc = cc + 85
dd = dd + 85
ee = ee + 85
ff = ff + 85
gg = gg + 85

Next loop2


aaa = 174
ccc = 176
ddd = 177
eee = 182
fff = 184
ggg = 185



a2 = Cells(89, 104)
c2 = Cells(91, 104)
d2 = Cells(92, 104)
e2 = Cells(97, 104)
f2 = Cells(99, 104)
g2 = Cells(100, 104)


For loop3 = 1 To trials

a2 = a2 + Cells(aaa, 104)
c2 = c2 + Cells(ccc, 104)
d2 = d2 + Cells(ddd, 104)
e2 = e2 + Cells(eee, 104)
f2 = f2 + Cells(fff, 104)
g2 = g2 + Cells(ggg, 104)

aaa = aaa + 85
ccc = ccc + 85
ddd = ddd + 85
eee = eee + 85
fff = fff + 85
ggg = ggg + 85

Next loop3
```

```
 Worksheets(4).Select   'puts the averages of slope, L10, and L50 on final summary page

 Cells(16, 2) = a2 / trials
 Cells(16, 3) = e2 / trials
 Cells(17, 2) = c2 / trials
 Cells(17, 3) = f2 / trials
 Cells(18, 2) = d2 / trials
 Cells(18, 3) = g2 / trials


 Cells(1, 1).Select


End Sub
```

Appendix E: Computer Simulation for Method 5

```
Sub Macro1()
'
' Macro1 Macro
'
' Keyboard Shortcut: Ctrl+e
'


Worksheets(1).Select
Application.ScreenUpdating = False

trialsB = Cells(13, 2)

taz = 87
'rowspace = 5
For loop37 = 1 To trialsB
Worksheets(1).Select

Column = 2
countL10 = 0
countL10B = 0
countmean = 0
countmeanB = 0



trials = Cells(12, 2)
'columnspace = 3
For loop10 = 1 To trials

alpha = Cells(9, 2)
alpha2 = Cells(10, 2)


If alpha > alpha2 Then

   x = 7
   For loop3 = 1 To alpha
   Cells(x, 3) = x - 6
   x = x + 1
   Next loop3
   beta = alpha
      Else
      x = 7
      For loop3 = 1 To alpha2
```

```
        Cells(x, 3) = x - 6
        x = x + 1
        Next loop3

    beta = alpha2
    End If




'Bin A
'generating random number
alpha = Cells(9, 2)
r = 7
num1 = 1
For loop1 = 1 To alpha
    Cells(r, 4).Select
here2:
    ActiveCell.FormulaR1C1 = "=randbetween(1,1000)"

'Checking for duplicate
s = 6
For randcheck = 1 To num1
    If Cells(r, 4) = Cells(s, 4) Then GoTo here2
    s = s + 1
Next randcheck
    r = r + 1
    num1 = num1 + 1
Next loop1

'sorting column
        ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Cells(7, 4), _
        SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
    With ActiveWorkbook.Worksheets("Sheet1").Sort
        .SetRange Range(Cells(7, 4), Cells(alpha + 6, 4))
        .Header = xlNo
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With
```

```
r1 = 7

For loop5 = 1 To alpha
'Rank A
   Cells(r1, 5) = (Cells(r1, 4) - 0.3) / (1000 + 0.4)
'S of A
   Cells(r1, 6) = 1 - Cells(r1, 5)
   SA = Cells(r1, 6)
'Life of A
      A1 = Application.WorksheetFunction.Ln(1 / SA)
      B1 = Application.WorksheetFunction.Ln(A1)
      CA = Exp(B1 / Cells(3, 2)) * Cells(6, 2)
      Cells(r1, 7) = CA
   r1 = r1 + 1
Next loop5

r11 = 7

For loop7a = 1 To alpha
'rank for 1 to sample size
      Z1 = (Cells(r11, 3) - 0.3) / (alpha + 0.4)
      Cells(r11, 8) = Z1

'S for sample size
      Z2 = 1 - Cells(r11, 8)
      Cells(r11, 9) = Z2

'lnln(1/S) for sample size
      Z5 = 1 / Z2
      Z3 = Application.WorksheetFunction.Ln(Z5)
      Z4 = Application.WorksheetFunction.Ln(Z3)
      Cells(r11, 10) = Z4
   r11 = r11 + 1
Next loop7a

'lnA
r12 = 7
For loop8 = 1 To alpha
Cells(r12, 11) = Application.WorksheetFunction.Ln(Cells(r12, 7))
r12 = r12 + 1
Next loop8
```

```
here99:



'Bin B
'random number
alpha2 = Cells(10, 2)
r = 7
num12 = 1
For loop2 = 1 To alpha2
   Cells(r, 12).Select
here4:
   ActiveCell.FormulaR1C1 = "=RANDBETWEEN(1,1000)"

'Checking for duplicate
s = 6
For randcheck2 = 1 To num12
   If Cells(r, 12) = Cells(s, 12) Then GoTo here4
   s = s + 1
Next randcheck2
   r = r + 1
   num12 = num12 + 1
Next loop2

' Sorting column
   ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
   ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Cells(7, 12), _
      SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
   With ActiveWorkbook.Worksheets("Sheet1").Sort
      .SetRange Range(Cells(7, 12), Cells(alpha2 + 6, 12))
      .Header = xlNo
      .MatchCase = False
      .Orientation = xlTopToBottom
      .SortMethod = xlPinYin
      .Apply
   End With


r1 = 7

For loop6 = 1 To alpha2
'Rank B
   Cells(r1, 13) = (Cells(r1, 12) - 0.3) / (1000 + 0.4)
'S of B
   Cells(r1, 14) = 1 - Cells(r1, 13)
   SB = Cells(r1, 14)
```

```
'Life of B
        a2 = Application.WorksheetFunction.Ln(1 / SB)
        B2 = Application.WorksheetFunction.Ln(a2)
        CB = Exp(B2 / Cells(4, 2)) * Cells(7, 2)
        Cells(r1, 15) = CB

       If CB > Cells(5, 2) Then GoTo here5

    r1 = r1 + 1
Next loop6


here5:
soccer = loop6
If soccer < Cells(10, 2) + 1 Then GoTo here99   'this is the part making sure the failure index is
fixed

If loop6 = Cells(10, 2) + 1 Then soccer = Cells(10, 2)




r111 = 7

For loop11a = 1 To soccer
'rank for 1 to sample size
        Z1b = (Cells(r111, 3) - 0.3) / (alpha2 + 0.4)
        Cells(r111, 16) = Z1b

'S for sample size
        Z2b = 1 - Cells(r111, 16)
        Cells(r111, 17) = Z2b

'lnln(1/S) for sample size
        Z5b = 1 / Z2b
        Z3b = Application.WorksheetFunction.Ln(Z5b)
        Z4b = Application.WorksheetFunction.Ln(Z3b)
        Cells(r111, 18) = Z4b
    r111 = r111 + 1
Next loop11a
```

```
'lnB
r13 = 7
For loop9 = 1 To soccer
Cells(r13, 19) = Application.WorksheetFunction.Ln(Cells(r13, 15))
r13 = r13 + 1
Next loop9


Cells(2, 4) = soccer
soccer2 = Cells(2, 4) + 6



Cells(5, 3) = loop10

'copying data from sheet1 to sheet3

   'Range(Cells(5, 3), Cells(beta + 6, 19)).Select
   'Selection.Copy
   'Worksheets("Sheet3").Select
   'Cells(rowspace, columnspace).Select
   'ActiveSheet.Paste
   'Worksheets("Sheet1").Select



'here down should change




'Organized answers on left of spreadsheet
'slope A
MA = Application.WorksheetFunction.LinEst(Range(Cells(7, 10), Cells(alpha + 6, 10)),
Range(Cells(7, 11), Cells(alpha + 6, 11)), True, True)
Cells(16, 2) = MA
MAa = Cells(16, 2)
Slope1A = MAa
'slope B
'MB = Application.WorksheetFunction.LinEst(Range(Cells(7, 18), Cells(alpha2 + 6, 18)),
Range(Cells(7, 19), Cells(alpha2 + 6, 19)), True, True)
MB = Application.WorksheetFunction.LinEst(Range(Cells(7, 18), Cells(soccer2, 18)),
Range(Cells(7, 19), Cells(soccer2, 19)), True, True)
Cells(24, 2) = MB
MBb = Cells(24, 2)
Slope1B = MBb
```

```
'intercepts A and B
Ba = Application.WorksheetFunction.Intercept(Range(Cells(7, 10), Cells(alpha + 6, 10)),
Range(Cells(7, 11), Cells(alpha + 6, 11)))
'Bb = Application.WorksheetFunction.Intercept(Range(Cells(7, 18), Cells(alpha2 + 6, 18)),
Range(Cells(7, 19), Cells(alpha2 + 6, 19)))
Bb = Application.WorksheetFunction.Intercept(Range(Cells(7, 18), Cells(soccer2, 18)),
Range(Cells(7, 19), Cells(soccer2, 19)))

'Lbeta A calculations
V2 = (Ba / MAa)
VV = -1 * V2
LBa = Exp(VV)

'Lbeta B calculations
V3 = (Bb / MBb)
MO = -1 * V3
LBb = Exp(MO)


'plotting Lbetas
Cells(17, 2) = LBa
Cells(25, 2) = LBb

sassyA1 = LBa
sassyB1 = LBb

'L10 A
L10a = Exp(-2.25037 / MAa) * LBa
Cells(18, 2) = L10a
L10A1 = L10a
'L10 B
L10b = Exp(-2.25037 / MBb) * LBb
Cells(26, 2) = L10b
L10B1 = L10b

'L50 A
L50a = Exp(-0.36651 / MAa) * LBa
Cells(19, 2) = L50a
L50A1 = L50a
'L50 B
L50b = Exp(-0.36651 / MBb) * LBb
Cells(27, 2) = L50b
L50B1 = L50b
```

```
'Mean@ A
meanata = 62.1 * (MAa ^ -0.172)
Cells(20, 2) = meanata
chevyA1 = meanata
'Mean@ B
meanatb = 62.1 * (MBb ^ -0.172)
Cells(28, 2) = meanatb
chevyB1 = meanatb
```

```
    'Mean A Gamma function method
    musb = (MAa + 1) / MAa
    a1a = Application.WorksheetFunction.GammaLn(musb)
    LmeanA = LBa * Exp(a1a)
    Cells(21, 2) = LmeanA
    cheeseA1 = LmeanA
    'Mean A
    'D1a = 1 / (1 - (meanata / 100))
    'D2a = Application.WorksheetFunction.Ln(D1a)
    'D3a = Application.WorksheetFunction.Ln(D2a)
    'meanA = (Exp(D3a / MAa)) * LBa
    'Cells(21, 2) = meanA
    'cheeseA1 = meanA

    'Mean B Gamma function method
    musbB = (MBb + 1) / MBb
    a1aB = Application.WorksheetFunction.GammaLn(musbB)
    LMeanAb = LBb * Exp(a1aB)
    Cells(29, 2) = LMeanAb
    cheeseB1 = LMeanAb
    'Mean B
    'D1b = 1 / (1 - (meanatb / 100))
    'D2b = Application.WorksheetFunction.Ln(D1b)
    'D3b = Application.WorksheetFunction.Ln(D2b)
    'meanB = (Exp(D3b / MBb)) * LBb
    'Cells(29, 2) = meanB
    'cheeseB1 = meanB
```

```
'DOF A-B
Cells(40, 2) = (Cells(9, 2) - 1) * (Cells(10, 2) - 1)
dofab = Cells(40, 2)

'CONFIDENCE INTERAL FOR A


    'Ao
    Anot = (-0.0844 / Cells(16, 2)) - 0.05584
    'Bo
    Bnot = (1.2796 / Cells(16, 2)) + 0.6729
    'lnDOF
    lnDOF = Application.WorksheetFunction.Ln(Cells(40, 2))
    'MLR at 99
    MLR99 = (Anot * lnDOF + Bnot) ^ 2 + 1
    Cells(41, 2) = MLR99
    'D
    Dvegas = 3.912 / (MLR99 - 1)
    'MLRexp
    If Cells(21, 2) > Cells(29, 2) Then
    MLRexp = Cells(21, 2) / Cells(29, 2)
    Else
    MLRexp = Cells(29, 2) / Cells(21, 2)
    End If
    'C
    Cvegas = 1 - 0.5 * Exp(-Dvegas * (MLRexp - 1))
    Cells(42, 2) = Cvegas

'CONFIDENCE INTERVAL FOR B

    'AoB
    AnotB = (-0.0844 / Cells(24, 2)) - 0.05584
    'BoB
    BnotB = (1.2796 / Cells(24, 2)) + 0.6729
    'lnDOF
    lnDOF = Application.WorksheetFunction.Ln(Cells(40, 2))
    'MLR at 99
    MLR99B = (AnotB * lnDOF + BnotB) ^ 2 + 1
    Cells(43, 2) = MLR99B
    'D
    DvegasB = 3.912 / (MLR99B - 1)
    'MLRexp
    If Cells(21, 2) > Cells(29, 2) Then
    MLRexpB = Cells(21, 2) / Cells(29, 2)
    Else
    MLRexpB = Cells(29, 2) / Cells(21, 2)
```

```
    End If
    'C
    CvegasB = 1 - 0.5 * Exp(-DvegasB * (MLRexpB - 1))
    Cells(44, 2) = CvegasB

'C average
Cvegasavg = (Cells(42, 2) + Cells(44, 2)) / 2
Cells(45, 2) = Cvegasavg

'L10 dependent confidence numbers

'A
aaa = Exp((4.5286 / Cells(16, 2)) + 0.3152)
'ln(m)
lnmA = Application.WorksheetFunction.Ln(Cells(16, 2))
'B
bbb = 0.29574 * lnmA + (-0.45228)
'L10LR
L10LRA = aaa * Cells(40, 2) ^ bbb
'ao
litanot = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRA = Application.WorksheetFunction.Ln(L10LRA)
'a
lita = litanot / lnL10LRA
    'L10exp
    If Cells(18, 2) > Cells(26, 2) Then
    xnot = Cells(18, 2) / Cells(26, 2)
    Else
    xnot = Cells(26, 2) / Cells(18, 2)
    End If
'ln(xo)
lnxo = Application.WorksheetFunction.Ln(xnot)
'CL10
CL10 = 1 - Exp(((lita * lnxo) ^ 2 + 2896.3) / -3595.9)
Cells(46, 2) = CL10


'A
AAAb = Exp((4.5286 / Cells(24, 2)) + 0.3152)
'ln(m)
lnmAb = Application.WorksheetFunction.Ln(Cells(24, 2))
'B
BBBb = 0.29574 * lnmAb + (-0.45228)
'L10LR
L10LRAb = AAAb * Cells(40, 2) ^ BBBb
```

```
'ao
litanotb = (-3595.9 * -4.60517 - 2896.3) ^ 0.5
'ln(L10LR)
lnL10LRAb = Application.WorksheetFunction.Ln(L10LRAb)
'a
litab = litanotb / lnL10LRAb
    'L10exp
    If Cells(18, 2) > Cells(26, 2) Then
    xnotb = Cells(18, 2) / Cells(26, 2)
    Else
    xnotb = Cells(26, 2) / Cells(18, 2)
    End If
'ln(xo)
lnxob = Application.WorksheetFunction.Ln(xnotb)
'CL10
CL10b = 1 - Exp(((litab * lnxob) ^ 2 + 2896.3) / -3595.9)
Cells(47, 2) = CL10b

'C L10 average
CL10avg = (CL10 + CL10b) / 2
Cells(48, 2) = CL10avg


Cells(38, 2) = soccer2 - 6

Range(Cells(7, 3), Cells(beta + 6, 19)).Select
Selection.ClearContents

Range("A1").Select

Worksheets("Summary").Select
```

```
Cells(2, 1) = loop37


'Slopes
Cells(4, Column) = Slope1A
Cells(12, Column) = Slope1B

'L10
Cells(6, Column) = L10A1
Cells(14, Column) = L10B1

'L50
Cells(7, Column) = L50A1
Cells(15, Column) = L50B1

'Mean
Cells(9, Column) = cheeseA1
Cells(17, Column) = cheeseB1

'Mean@
Cells(8, Column) = chevyA1
Cells(16, Column) = chevyB1

'Lb
Cells(5, Column) = sassyA1
Cells(13, Column) = sassyB1

'DOF
Cells(28, Column) = dofab


Cells(26, Column) = soccer2 - 6




'MLR99
Cells(29, Column) = MLR99
Cells(31, Column) = MLR99B




'Cvegas
Cells(30, Column) = Cvegas
Cells(32, Column) = CvegasB
```

```
'C average
Cells(33, Column) = Cvegasavg




'CL10
Cells(34, Column) = CL10
Cells(35, Column) = CL10b
Cells(36, Column) = CL10avg



'L10A / L10B
Cells(60, Column) = Cells(6, Column) / Cells(14, Column)


'MeanA / MeanB
Cells(64, Column) = Cells(9, Column) / Cells(17, Column)


'Counting which is bigger, L10A or L10B
If Cells(60, Column) > 1 Then
countL10 = countL10 + 1
Else
countL10B = countL10B + 1
End If
Cells(68, 2) = countL10
Cells(69, 2) = countL10B



'Counting which is bigger meanA or meanB
If Cells(64, Column) > 1 Then
countmean = countmean + 1
Else
countmeanB = countmeanB + 1
End If
Cells(75, 2) = countmean
Cells(76, 2) = countmeanB
```

```
Cells(2, Column) = Column - 1

Worksheets(1).Select



Cells(1, 1).Select

Column = Column + 1
'columnspace = columnspace + 19

Next loop10


Worksheets("Summary").Select

If Cells(68, 2) > Cells(69, 2) Then
Cells(68, 3) = "A>B"
Else
Cells(68, 3) = "B>A"
End If



If Cells(75, 2) > Cells(76, 2) Then
Cells(75, 3) = "A>B"
Else
Cells(75, 3) = "B>A"
End If



Cells(83, 3) = "=average(B36:CW36)"
```

```
Cells(83, 6) = "=average(B33:CW33)"




    Range(Cells(2, 1), Cells(85, 101)).Select
    Selection.Copy
    Cells(taz, 1).Select
    ActiveSheet.Paste
    Range("B2:CW85").Select
    Selection.ClearContents

    taz = taz + 85
    'rowspace = rowspace + beta + 3
Next loop37


Worksheets("Sheet1").Select



trials = Cells(13, 2)


Worksheets("Summary").Select

a = 168
ab = 169
ac = 170
d = 6
b = 3

Cells(2, 3) = Cells(168, 3)


Cells(2, 6) = Cells(168, 6)


For cl10avgavgavgloop = 1 To trials
Cells(2, 3) = Cells(2, 3) + Cells(a + 85, 3)
Cells(2, 6) = Cells(2, 6) + Cells(a + 85, 6)
a = a + 85
```

```
Next cl10avgavgavgloop
Cells(2, 3) = Cells(2, 3) / trials


Cells(2, 6) = Cells(2, 6) / trials



az = Cells(2, 3)


ax = Cells(2, 6)


Cells(1, 1).Select



Worksheets("SummaryB").Select

Cells(1, 2) = az

Cells(5, 2) = ax



Worksheets("Summary").Select

amcavg = 153
cmcavg = 154
dmcavg = 155
emcavg = 156
fmcavg = 157
gmcavg = 158


For loopmcavg = 1 To trials


Cells(1, 10) = Cells(amcavg, 2) + Cells(1, 10)
Cells(1, 11) = Cells(cmcavg, 2) + Cells(1, 11)


amcavg = amcavg + 85
cmcavg = cmcavg + 85
```

```
Next loopmcavg

bmcavg = Cells(1, 10) / trials
hmcavg = Cells(1, 11) / trials

Worksheets("SummaryB").Select
Cells(10, 2) = bmcavg
Cells(11, 2) = hmcavg

Cells(1, 1).Select

'this is where it takes the averages of slope, L10, L50

Worksheets(1).Select
t1 = Cells(12, 2)
trials = Cells(13, 2)

Worksheets(2).Select

aa = 89
cc = 91
dd = 92
ee = 97
ff = 99
gg = 100

For loop2 = 1 To trials

a = Cells(aa, 2)
b = 3
c = Cells(cc, 2)
```

```
d = Cells(dd, 2)
e = Cells(ee, 2)
f = Cells(ff, 2)
g = Cells(gg, 2)
    For loop1 = 1 To t1
    a = a + Cells(aa, b)
    c = c + Cells(cc, b)
    d = d + Cells(dd, b)
    e = e + Cells(ee, b)
    f = f + Cells(ff, b)
    g = g + Cells(gg, b)
    b = b + 1
     Next loop1
    Cells(aa, 104) = a / t1
    Cells(cc, 104) = c / t1
    Cells(dd, 104) = d / t1
    Cells(ee, 104) = e / t1
    Cells(ff, 104) = f / t1
    Cells(gg, 104) = g / t1



aa = aa + 85
cc = cc + 85
dd = dd + 85
ee = ee + 85
ff = ff + 85
gg = gg + 85

Next loop2


aaa = 174
ccc = 176
ddd = 177
eee = 182
fff = 184
ggg = 185



a2 = Cells(89, 104)
c2 = Cells(91, 104)
d2 = Cells(92, 104)
e2 = Cells(97, 104)
f2 = Cells(99, 104)
```

```
g2 = Cells(100, 104)


For loop3 = 1 To trials

a2 = a2 + Cells(aaa, 104)
c2 = c2 + Cells(ccc, 104)
d2 = d2 + Cells(ddd, 104)
e2 = e2 + Cells(eee, 104)
f2 = f2 + Cells(fff, 104)
g2 = g2 + Cells(ggg, 104)

aaa = aaa + 85
ccc = ccc + 85
ddd = ddd + 85
eee = eee + 85
fff = fff + 85
ggg = ggg + 85

Next loop3



 Worksheets(4).Select

 Cells(16, 2) = a2 / trials
 Cells(16, 3) = e2 / trials
 Cells(17, 2) = c2 / trials
 Cells(17, 3) = f2 / trials
 Cells(18, 2) = d2 / trials
 Cells(18, 3) = g2 / trials


 Cells(1, 1).Select


End Sub
```

Appendix F: Abstract and Poster for 2009 STLE Annual Meeting

PRESENTATION TYPE: Student Poster
TOPIC: Student Poster Competition
TITLE: Confidence Ranking of Monte Carlo Simulated Fatigue Data based on a Weibull-Johnson Methodology
AUTHORS (LAST NAME, FIRST NAME): Murray, Noel S.[1]
INSTITUTIONS (ALL): 1. Mechanical Engineering Technology, Georgia Southern University, Statesboro, GA, USA.
ABSTRACT BODY:
Body: Statistical and probabilistic models of fatigue lives were used to determine whether data sets were significantly different. Monte Carlo simulations based on Weibull-Johnson parameters were used to simulate fatigue lives. These lives can be for bearings, shafts, gears, or any component that fails as a result of fatigue. The Monte Carlo simulation is repeated one hundred times to determine a confidence number. Linear approximations of Leonard Johnson's Confidence Number curves were used to calculate separate confidence numbers, and were compared to those generated by the Monte Carlo simulations. This work contributes to the validation of the linear approximations used, which expand greatly on the limited cases published by Johnson. A known experimental data set was also used to validate the Monte Carlo simulations, validating the relative ranking of fatigue data sets with variations due to test conditions, material variations, differences between batches and heat treatments, and vendors.

Biography: I went to high school at Mount de Sales Academy in Macon Georgia and graduated in 2003. I then went straight to Georgia Southern University and graduated in May 2008 with a BS in Mechanical Engineering Technology. I started graduate school at Georgia Southern in August 2008 where I am currently getting a masters in Applied Engineering with a concentration in Engineering Management. While a student at Georgia Southern I have been involved with Formula SAE, SAE Mini Baja, Senior design project where we built a solar powered car, and competed in the History Channel's "City of the Future" competition.
KEYWORDS: Computer Use in Maintenance, Statistical Analysis, Maintenance.

Confidence Ranking of Monte Carlo Simulated Fatigue Data based on Weibull-Johnson Methodology

Noel Murray
Georgia Southern University, Department of Mechanical and Electrical Engineering Technology

Abstract

Probabilistic models of fatigue lives were used to determine whether data sets for 2 spur gears were significantly different. Monte Carlo simulations based on Weibull-Johnson parameters were used to simulate fatigue lives. The Monte Carlo simulation is repeated one hundred times to determine a confidence number. Linear approximations of Leonard Johnson's Confidence Number curves were used to calculate confidence numbers, and were compared to those generated by the Monte Carlo simulations. Comparison was also made to graphically determined Confidence numbers. This work contributes to the validation of the linear approximations used, which expand greatly on the limited cases published by Johnson. A known experimental data set was also used to validate the Monte Carlo simulations. It was shown that the L_{10} fatigue life of a spur gear made of Material A was not significantly different from that of material B.

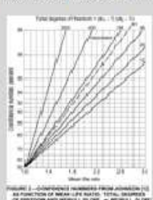
Introduction

Weibull Analysis of Fatigue Data
- Empirical Equation Discovered by W. Weibull in the 1920's
- Fit to fracture data
- Requires small data sets to predict fatigue failure at any level of survivability
- 2-Parameter Weibull Equation

$$Ln\,Ln\frac{1}{S} = m\,Ln\frac{L_{S}}{L_\beta}$$

- S – Probability of Survival
- m- slope of Weibull plot (indicates data scatter)
- L_S - Life at the indicated level of survival
- L_β – characteristic life (at which 63.2% have failed)

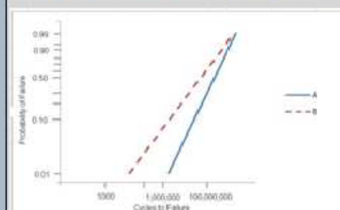
Establishing when fatigue data sets are different
- Leonard Johnson Confidence Numbers establish the number of times out of 100 the life of one data set is greater than another if the process is repeated
- Confidence Numbers are a function of Weibull Parameters (L_0 or mean life ratios and Weibull slope) and degree of freedom
- Johnson's method is limited to graphically determining the Confidence number.



- Forced to interpolate between curves and figures

Introduction (cont.)

Weibull Plot for Spur Gear A and B
Based on Data in the Literature (Townsend, Zaretsky, et. al)



Method of establishing confidence that two values are statistically different

- Need to establish with confidence that the L_{10} fatigue life of spur gear A is greater than spur gear B

Method 1: Graphically Determine Johnson Confidence Number

Method 2: Calculate Using Linear Approximations

Vlcek, Hendricks and Zaretsky Developed Linear Approximations to Fit Johnson's Curves

$$C_{L_{10}} = 1 - e^{-\frac{[(a \times \ln(x_r))^2 + E_1]}{E_2}}$$

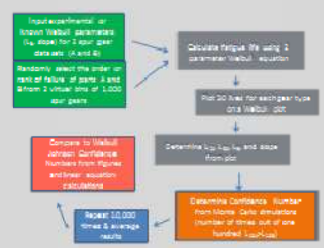
- a- constant using fitted L_{10} fitted ratio
- x_r- experimental L_{10} life ratio
- E_1- scaling constant
- E_2- scaling constant

Method

Method 3: Microsoft Excel Monte Carlo Simulation of Fatigue Lives Repeated 100 times to determine Confidence Number

- The Weibull parameters (slope and characteristic life) of the test populations being compared are known from the literature
- A virtual bin of 1,000 fatigue failed parts is assumed
- Corresponding to a comparative set of experimental data, the test population size (n) was fixed at 20.
- For each set of data, the ranking identification of 20 parts are randomly pulled from the virtual bin
- Using the 2-Parameter Weibull equation, the rankings are converted to fatigue lives
- Weibull plot are generated from which simulated L_{10} lives and Weibull slopes are determined
- The process is repeated for two or more data sets to be compared
- The relative ranking of the data sets' L_{10} Lives is determined
- The process is repeated 100 times to determine the percentage of times the L_{10} life of data set A is greater than that of data set B
- In turn, the process was repeated 10,000 times for statistical confidence
- As a test case, the confidence in the relative ranking of fatigue lives of two different spur gear materials was determined using these 3 methods and Weibull parameters reported in the literature (Townsend, Zaretsky et al).

Flow chart of Excel Macro of Weibull-based Monte Carlo Simulation



Results

Table 1: Johnson Confidence Numbers (based on 3 different methods) that the L10 Fatigue Life of Spur Gear Material B is Significantly Greater than Spur Gear Material A

	Confidence Number
Method 1: Graphical Determination from Johnson Figures as reported in Townsend, Zaretsky et al	0.84
Method 2: Calculated using linear Approximations of Vlcek, Hendricks, and Zaretsky and Monte Carlo Fatigue Lives	0.83
Method 3: Determined by Monte Carlo Simulation of 100 Fatigue Lives (the average of 10,000 simulations)	0.92

Discussion and Conclusion

- The statistical difference between two different L_{10} fatigue lives was determined using Johnson Confidence Numbers
- Johnson Confidence Numbers were determined using a graphical, linear approximation, and Monte Carlo Simulation
- The graphical and linear approximation were in excellent agreement (0.84 and 0.83 respectively)
- The Monte Carlo method trended slightly higher with an average Confidence Number 0.92 for 10,000 simulations
- The Monte Carlo Method is dependant upon the accuracy of the initial Weibull Inputs (these were based upon only 20 trials).
- Based on Method 1 and 2, there is no statistical difference between the L_{10} fatigue Lives of gear A and gear B.

References

Weibull, W. (1951), "A Statistical Distribution Function of Wide Applicability," J. Appl. Mech. Trans ASME, 18, 3, pp. 293–297.

Johnson, L. G.; The Statistical Treatment of Fatigue Experiments. Elsevier Publishing Co., Amsterdam, The Netherlands, 1964

Townsend, D. P., Zaretsky, E. V., Anderson, N. E.; Comparison of Modified Vasco X-2 with AISI 9310 – Preliminary Report, 1977 NASA TM-X-73649

Brian L Vlcek, Robert C Hendricks, and Erwin V. Zaretsky. Determination of Rolling-Element Fatigue Life from Computer Generated Bearing Tests. Tribology Transactions, 46 (4): p 479-493, Oct 2003.

Brian L Vlcek, Robert C. Hendricks, and Erwin V. Zaretsky. Relative Ranking of Fatigue Lives of Rotating Aluminum Shafts Using L_{10} Weibull-Johnson Confidence Numbers. Proceedings of the 12th International Conference on Rotating Machine. March 2008. Honolulu, HI.

Acknowledgements:

Brian L Vlcek, PhD
Professor and Program Coordinator

Appendix G: Long Abstract Submitted for Presentation at 2010 STLE Annual Meeting

2010 STLE Annual Meeting & Exhibition
May 16-20, 2010
Bally's Las Vegas Hotel and Casino
Las Vegas, Nevada, USA

GEARS I: Session

Modified Vasco X-2 and AISI 9310 Spur Gear Fatigue Failure Revisited with Weibull-Johnson Monte Carlo Simulations

Noel Murray

Brian L Vlcek, PhD

Georgia Southern University

Statesboro, GA 30460

BLVLCEK@GeorgiaSouthern.edu

912-478-5721

INTRODUCTION

Over three decades ago, Vasco X-2 was considered as a superior alternative for AISI 9310 as a gear material for aircraft and helicopter transmission systems. The Vasco X-2 material was effectively a through hardened steel (H-12 tool steel) with a reduced carbon content (0.11 to 0.16 percent). Since the finished product could be case hardened while maintaining a soft core, fracture toughness of gears manufactured from the material was expected to improve [1]. The hardness and fatigue life (rolling contact and gear tests) of Vasco X-2 for three different heat treatment methods, were determined experimentally by Townsend, Zaretsky, and Anderson [1-

2], and compared to results for AISI 9310. Since sufficient information existed in the literature [1-2] to determine inputs for a Monte Carlo simulation of fatigue life based upon the method of Vlcek, et. al [3], this spur gear material study was revisited.

Monte Carlo Simulation of Fatigue Lives

For the Monte Carlo model, it is assumed that 1000 virtual gears exist in a virtual bin. While the actual magnitude of the fatigue life of each of the 1000 virtual gears is not known, the relative ranking (from 1 to 1000) of the gears lives is assumed to be known, and is ordered from 1 to 1000. A subset of desired number of test specimens (test specimen population size) is randomly drawn from the bin by randomly determining order numbers. If the Weibull parameters (Weibull slope and the characteristic life at which 63.2% of test specimens having failed) for the material are known from some limited amount of previous experimentation or modeling, the above order numbers can be converted into gear lives using the two-Parameter Weibull Equation [4], where

$$\ln\ln\left(\frac{1}{S}\right)=m\ln\left(\frac{L_S}{L_\beta}\right) \quad \text{where } 0<L<\infty;\, 0<S<1 \tag{1}$$

From this subset of simulated lives, Weibull parameters (L_{10}, L_{50}, and slope) are determined using a least squares linear curve fit methodology. The process is then repeated a sufficient number of times (10,000) to establish statistical confidence in the observed trends. Vlcek et. al. [3] demonstrated that a Weibull-based Monte Carlo simulation could be used to predict the fatigue life of simple multi-component rolling element bearings, and reasonable agreement with experimental data was shown. Similarly, Weibull-based Monte Carlo simulations were used to model the fatigue life of single-component rotating aluminum shafts [5], and complex multi-component helicopter transmissions [6].

The fatigue lives of three different Vasco X-2 gear material lots, each with a different heat treatment, were reported in the literature [2]. The heat treatment lots were identified as Boeing-Vertol, NASA, and Curtis Wright. Corresponding heat treatment processes and resulting material properties can be found in references 1-2. The experimental fatigue lives were determined by Townsend et. al [1-2] using both the rolling-contact bench top test and the NASA gear test apparatus. Weibull slopes, L_{10}, and L_{50} from the experimental gear tests [2] are summarized in Table 1. The failure index (number of failures out of the number of tests attempted) is also reported.

Unlike the previous Monte Carlo fatigue simulations reported by Vlcek et. al. (3), this experimental data set included suspensions/censored data. A suspension is a test that is terminated before failure. The suspension may be random, due to factors such as mechanical or electrical breakdown of the tester. More likely suspensions are intentional, such as stopping a test for economical reasons after a predefined threshold is exceeded. While a suspension cannot be treated as a failure, it nevertheless represents useful information that should not be ignored. Leonard Johnson [7] discusses a methodology for accounting for suspensions in a Weibull analysis.

Suspensions within the Monte Carlo Simulation

In the case of the Townsend [2] study, gear fatigue failure lives exceeding a predetermined threshold were encountered, and testing was suspended resulting in a failure index of 12 out of 26 for the Modified Vasco X-2 gears heat treated according to the procedure of Boeing Vertol [2]. In the case of gears heat treated according to the NASA procedure, the failure index was 18 out of 21. To account for suspended tests, four different suspension models were developed and integrated into the Monte Carlo model of Vlcek et. al. [3]. The suspension models were:

(i) Use Weibull parameters (Characteristic Life at which 63.2 percent of the specimens will have failed and Weibull slope) determined for the experimental fatigue lives (accounting for suspensions according to Johnson [7] in the preliminary Weibull analysis) as model inputs, and draw simulated fatigue life subpopulations equal in size to the total number of experimental tests attempted/started,

(ii) Use Weibull parameters (Characteristic Life and Weibull slope) determined for the experimental fatigue lives as model inputs, and draw simulated fatigue life subpopulations equal in size to the failure index of the suspended experiments,

(iii) Use Weibull parameters (Characteristic Life and Weibull slope) determined for the experimental fatigue lives as model inputs, assemble a random set of fatigue lives equal to the number of tests attempted, order the random set from smallest to largest, and eliminated a sufficient number of large lives (working backwards from the largest) to match the failure index, and

(iv) Use Weibull parameters (Characteristic Life and Weibull slope) determined for the experimental fatigue lives as model inputs, assemble a random set of fatigue lives equal to the number of tests attempted, order the random set from smallest to largest, and eliminated all lives above a predefined threshold.

Confidence Numbers

Statistical significance between compared L_{10} lives of two materials was established by using Johnson Confidence Numbers [7]. A Confidence Number is a statement of how often the same probabilistic variable, such as the L_{10} fatigue life, of material A will be observed as being greater than that of Material B if the test and measurement were repeated one hundred times. Confidence Numbers greater than 90 percent (i.e. 90 out of 100 times measurement (A) will be greater than measurement (B) are considered statistically significant, different or independent.

Confidence Numbers were determined by (a) graphically interpolating the values from the published Figures of Leonard Johnson [*7*], and (b) by a method of comparing groups of 100 L_{10} lives determined for Monte Carlo simulated fatigue lives.

For technique (a) Confidence Numbers were graphically determined using curves and figures developed by Leonard Johnson [7]. Knowing the Weibull slope to select the correct figure, the correct curve on the figure is selected based upon total degrees of freedom. The Confidence number is read at the intersection between the ratio of two L_{10} lives being compared and the curve associated to the total degrees of freedom, the Confidence Number can be read. Values that fall between curves (and figures) require graphical interpolation.

For Confidence Number technique (b), one hundred L_{10} lives are determined using Monte Carlo simulated fatigue lives (as presented above) for each of two gear materials to be compared. The first L_{10} life of material A is compared to the first of material B to determine which is greater. The process is repeated for the second thru the one hundredth. In this manner the number of L_{10} lives of material A that were greater than the number of L_{10} lives of material B (out of one hundred) is established. The process is repeated a sufficient number of times (5,000-10,000) to establish significance in the trends in the results.

Experimental Basis For Monte Carlo Simulation

Townsend et al [2] also performed Rolling-Contact Fatigue tests for rolling elements made from AISI 9310 and Modified Vasco X-2 (NASA Heat Treatment). Their results are summarized in Table 2. Monte Carlo simulated fatigue lives of both AISI 9310 and Modified Vasco X-2 (NASA Heat Treatment) were determined as part of this study. Confidence Numbers were again used to establish with statistical significance whether or not the L_{10} lives of the two materials were different.

RESULTS AND DISCUSSION

Monte Carlo simulated fatigue lives for AISI 9310 and Modified Vasco X-2 (3 different heat treatments) were determined using a Johnson-Weibull Monte Carlo model developed and validated by Vlcek et. al. [3]. Experimentally determined Weibull slope and characteristic life served as model inputs.

Two of the test series reported in Townsend et. al. [2] had suspensions/censored data within the data sets—long lived test runs were terminated at a predetermined threshold. Modified Vasco X-2 spur gears with heat treatment procedure according to Boeing Vertol had a Failure Index of 12 of 26; in other words, 26 tests were attempted, 12 failed, and 14 long-lived tests were terminated at a threshold. Twenty-one modified Vasco X-2 spur gears (with a heat treatment procedure according to NASA) tests were attempted and eighteen failed with three suspended. The Monte Carlo fatigue failure model developed by Vlcek et. al. [3] had to be modified to account for suspensions (censored data sets).

The average simulated and experimental (from ref [2]) spur gear fatigue L_{10} life for both AISI 9310 and Vasco X-2 (three heat treatment procedures) are summarized in Table 3a. Additionally, the L_{50} life is reported in Table 3b, and the Weibull slopes are summarized in Table 3c. The Weibull slopes are a reflection of the scatter within the data; increasing slope indicates a decrease in scatter.

The simulated results, as reported in Table 3a-c, were generally in very good agreement with the experimental results reported in Townsend et. al [2]. The spur gears manufactured from Vasco X-2 according to the Boeing Vertol heat treatment procedure had a significantly longer average L_{10} fatigue life than those obtained for the other two heat treatment procedures of Vasco X-2. There is an order of magnitude difference between L_{10} life obtained with the Boeing Vertol heat treatment and the other two techniques. This was consistent with the trend observed in the experimental work of Townsend et. al. [2]. The average simulated L_{10} life of the spur gears manufactured from Vasco X-2 according to the Boeing Vertol heat treatment procedure for each of the three suspension models were within 3.5 percent of the experimentally determined [2] L_{10} life (Table 3a). There is also excellent agreement between simulated and experimental L_{50} lives (Table 3b) and simulated and experimental Weibull slopes (Table 3c).

The average simulated L_{10} life of spur gears manufactured from AISI 9310 was within 3.9 percent of the experimentally determined [2] L_{10} life (Table 3a). The average simulated L_{50} fatigue life was within one percent of the experimental value (Table 3b) and the percent difference in Weibull slopes was 3.5 percent (Table 3c).

While the simulated L_{10} life of spur gears manufactured from Vasco X-2 (Boeing Vertol heat treatment) was greater than the simulated L_{10} life of spur gears manufactured from AISI 9310 by a factor of 1.68, the statistical significance of this difference needs to be determined. To this end, Confidence Numbers were determined two different ways—(a) using the graphical techniques of Johnson [7], and (b) and establishing the comparative ranking between two sets of one hundred L_{10} fatigue lives. The results of both techniques are summarized in Table 4. The AISI 9310 served as the baseline to which the three heat treatments of Vasco X-2 were compared. In the case of the spur gears manufactured from Vasco X-2 and heat treated according to the Boeing Vertol method and those manufactured from AISI 9310, there was no statistical difference between L_{10} lives. The Confidence Numbers ranged from 67-77 percent for each of the four Suspension Models considered. Since the Confidence Number did not exceed ninety-percent, the differences in L_{10} life are not considered statistically significant. Additionally, the simulated Confidence Numbers are in good agreement with that determined graphically using the experimental results of Townsend et al [2].

With a simulated Confidence Number of 100 (Table 4), it can be stated that there is a significant difference between the L_{10} life of spur gears manufactured from Vasco X-2 (NASA heat treatment) and AISI 9310 (Table 4). The L_{10} life of the Modified Vasco X-2 (NASA heat treatment) is significantly less than that of the AISI 9310 (Table 3a). Similarly, with a simulated Confidence Number also of 100 (Table 4), the L_{10} life of the Modified Vasco X-2 (Curtis-Wright heat treatment) is significantly less than that of the AISI 9310 (Table 3a). These observations are consistent with those of Townsend et al. who graphically determined Confidence numbers of 99 percent for both of the above comparisons.

CONCLUSIONS

1. Monte Carlo simulations of spur gear fatigue life predicted the L_{10} life of spur gears manufactured from both AISI 9310 and Modified Vasco X-2 with very good agreement with values experimentally determined by Townsend et. al. [2].

2. Three of the four Suspension Models (i iii and iv) resulted in simulated Confidence Numbers that were reasonably close to those obtained graphically with experimental results from Townsend et.al. [2].

3. The simulated L_{10} fatigue life of spur gears manufactured from Modified Vasco X-2 was dependent upon heat treatment technique. This observation is consistent with that experimentally determined by Townsend et. al. {2].

4. While the simulated L_{10} life of spur gears manufactured from Modified Vasco X-2 (Boeing Vertol heat treatment) was 1.68 times greater than that of spur gears manufactured from AISI 9310, there was no statistical difference between the two L_{10} lives based upon simulated Confidence Numbers. This conclusion is consistent with that experimentally determined by Townsend et. al. [2].

REFERENCES

1. Townsend, D.P., Zaretsky, E.V., Anderson, N.E.: "Comparison of Modified Vasco X-2 with AISI 9310 – Preliminary Report." NASA Technical Memorandum. Cleveland, Ohio. April 1977.

2. Townsend, D.P., Zaretsky, E.V.,: "Comparison of Modified Vasco X-2 with AISI 9310 Gear Steels." NASA Technical Paper. Cleveland, Ohio. November 1980.

3. Vlcek, B. L., Hendricks, R. C., and Zaretsky, E. V. (2003), "Determination of Rolling-Element Fatigue Life from Computer Generated Bearing Tests," STLE Tribology Trans. 46, 4, pp. 479-493.

4. Weibull, W. (1951), "A Statistical Distribution Function of Wide Applicability," J. Appl. Mech. Trans. ASME, 18, 3, pp. 293–297.

5. Vlcek, B. L., Hendricks, R.C. and Zaretsky, E. V.: " Relative Ranking of Fatigue Lives of Rotating Aluminum Shafts Using L_{10} Weibull-Johnson Confidence Numbers." Proceedings of the 12th International Conference on Rotating Machine. March 2008. Honolulu, HI.

6. Zaretsky, E.V., Lewicki, D.G., Savage, M., and Vlcek, B.L.:"Determination of Turboprop Reduction Gearbox System Fatigue Life and Reliability." *STLE Tribology Transaction* Volume 50, Issue 4, pp 507-516, October 2007.

7. Johnson, L.G.: The statistical Treatment of Fatigue Experiments. Elsevier Publishing Co., Amsterdam, The Netherlands, 1964.

KEYWORDS: Gear Failure, Probabilistic Fatigue Analysis, Weibull Analysis, AISI 9310 Material, Vasco X-2 Material, Confidence Numbers

Table 1 Summary of Experimental Gear Fatigue Life Results from Townsend et al [2]						
Material	Heat Treatment Procedure (Ref 2)	L_{10} Gear System Life (revolutions)	L_{50} Gear System Life (revolutions)	Weibull Slope	Tests Attempted (subset size)	Failures of Tests Attempted
AISI 9310	-----	$23x10^6$	$52x10^6$	2.3	30	30
Modified Vasco X-2	Boeing Vertol	$38.4x10^6$	$253x10^6$	1.0	26	12
Modified Vasco X-2	NASA	$0.8x10^6$	$27.6x10^6$	0.53	21	18
Modified Vasco X-2	Curtis-Wright	$3.3x10^6$	$8x10^6$	2.1	19	19
Gear Test Parameters: Pitch diameter, 8.89 centimeters (3.5 inch); spur gears; maximum Hertz stress, $1.71x10^9$ N/m^2 (248 ksi); speed, 10,000 rpm; lubricant, synthetic paraffinic oil; gear temperature, 86^{o}C (170^{o}F).						

Table 2 Summary of Experimental Rolling Contact Fatigue Life Results from Townsend et al [2]						
Material	Heat Treatment Procedure (Ref 2)	L_{10} Life (stress cycles)	L_{50} Life (stress cycles)	Weibull Slope	Tests Attempted (subset size)	Failures of Tests Attempted
AISI 9310	-----	$4.18x10^6$	$9.43x10^6$	2.31	10	10
Modified Vasco X-2	NASA	$6.3x10^6$	$14.8x10^6$	2.2	20	20
Rolling Contact Test Parameters: Speed, 25,000 stress cycles per minute; maximum Hertz stress, $4823x10^6$ N/m^2 (700 000 psi); lubricant, MIL-L-7808; temperature, ambient.						

Table 3a: Average L_{10} Life (cycles) Determined with Experimental [2]and Monte Carlo Simulated Data (4 different methods of modeling suspensions) for AISI 9310 and Modified Vasco X-2 (three different heat treatments) Gear Material

Material	Heat Treatment Procedure	Suspension Model (i)	Suspension Model (ii)	Suspension Model (iii)	Suspension Model (iv)	Experimental [2]
AISI 9310	----	$22.12x10^6$	n/a*	n/a*	n/a*	$23x10^6$
Modified Vasco X-2	Boeing Vertol	$37.06x10^6$	$39.39x10^6$	$38.41x10^6$	$38.63x10^6$	$38.4x10^6$
Modified Vasco X-2	NASA	$0.97x10^6$	$1.02x10^6$	$1.01x10^6$	$1.02x10^6$	$0.8x10^6$
Modified Vasco X-2	Curtis-Wright	$3.15x10^6$	n/a*	n/a*	n/a*	$3.3x10^6$

n/a* -- Model not applicable because there were no suspensions in the test

Method (i) – test population size equals attempted number of runs

Method (ii)—test population size equals number of failures

Method (iii)—test population size equals attempted number of runs, top of test populaton subset truncated to match number of failures

Method (iv))—test population size equals attempted number of runs, top of test population subset truncated at a predetermined threshold ($300x10^6$)

Table 3b: Average L_{50} Life (cycles) Determined with Experimental [2]and Monte Carlo Simulated Data (4 different methods of modeling suspensions) for AISI 9310 and Modified Vasco X-2 (three different heat treatments) Gear Material

		Suspension Model (i)	Suspension Model (ii)	Suspension Model (iii)	Suspension Model (iv)	Experimental [2]
AISI 9310	----	52.34×10^6	n/a*	n/a*	n/a*	52×10^6
Modified Vasco X-2	Boeing Vertol	255.66×10^6	261.84×10^6	307.97×10^6	296.20×10^6	253×10^6
Modified Vasco X-2	NASA	30.43×10^6	30.69×10^6	31.79×10^6	31.15×10^6	27.6×10^6
Modified Vasco X-2	Curtis-Wright	8.1×10^6	n/a*	n/a*	n/a*	8×10^6

n/a* -- Model not applicable because there were no suspensions in the test

Method (i) – test population size equals attempted number of runs

Method (ii)—test population size equals number of failures

Method (iii)—test population size equals attempted number of runs, top of test populaton subset truncated to match number of failures

Method (iv))—test population size equals attempted number of runs, top of test population subset truncated at a predetermined threshold (300×10^6)

Table 3c: Average Weibull Slope Determined with Experimental [2]and Monte Carlo Simulated Data (4 different methods of modeling suspensions) for AISI 9310 and Modified Vasco X-2 (three different heat treatments) Gear Material

		Suspension Model (i)	Suspension Model (ii)	Suspension Model (iii)	Suspension Model (iv)	Experimental [2]
AISI 9310	----	2.22	n/a*	n/a*	n/a*	2.3
Modified Vasco X-2	Boeing Vertol	0.96	0.96	0.99	0.99	1.0
Modified Vasco X-2	NASA	0.51	0.51	0.51	0.51	0.53
Modified Vasco X-2	Curtis-Wright	2.02	n/a*	n/a*	n/a*	n/a*

n/a* -- Model not applicable because there were no suspensions in the test

Method (i) – test population size equals attempted number of runs

Method (ii)—test population size equals number of failures

Method (iii)—test population size equals attempted number of runs, top of test populaton subset truncated to match number of failures

Method (iv))—test population size equals attempted number of runs, top of test population subset truncated at a predetermined threshold ($300x10^6$)

Table 4: Establishing Statistical Significance between L10 Lives Using Confidence Numbers Determined By Two Different Techniques						
		Simulated Confidence Numbers+				Graphical Confidence Numbers++
Material	Heat Treatment	Suspension Model (i)	Suspension Model (ii)	Suspension Model (iii)	Suspension Model (iv)	Experimental [2]
AISI 9310	----	----	----	----	----	----
Modified Vasco X-2	Boeing Vertol	77	67	76	76	80
Modified Vasco X-2	NASA	100	100	100	100	99
Modified Vasco X-2	Curtis-Wright	100	100	100	100	99

Simulated Confidence Numbers+ -- Determining the number (out of 100) of L_{10} lives (determined for simulated fatigue lives) in group A that are greater than those in group B

Graphical Confidence Numbers++ --Graphical Interpolation of Leonard Johnson's Figures [7] using experimental results from Townsend et al [2].

Method (i) – test population size equals attempted number of runs

Method (ii)—test population size equals number of failures

Method (iii)—test population size equals attempted number of runs, top of test populaton subset truncated to match number of failures

Method (iv))—test population size equals attempted number of runs, top of test population subset truncated at a predetermined threshold ($300x10^6$)

Appendix H: Poster from GSU COGS Poster Competition 2012