Reliability-based Analysis and Design of 2D Trusses

by

Alexis Joseph Ludena

Bachelor of Science in Civil Engineering
Massachusetts Institute of Technology, 2014

Submitted to the Department of Civil and Environmental Engineering
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING IN CIVIL AND ENVIRONMENTAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Signature of Author: Signature redacted
Department of Civil and Environmental Engineering
May 9, 2014

Certified by: Signature redacted
Jerome J. Connor
Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by: Signature redacted
Heidi M. Nepf
Chair, Departmental Committee for Graduate Students

Reliability-based Analysis and Design of 2D Trusses

by

Alexis Joseph Ludena

Submitted to the Department of Civil and Environmental Engineering
on May 9,2014, in Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING IN CIVIL AND ENVIRONMENTAL ENGINEERING

Abstract

Current safety factors used in structural design do not accurately account for uncertainties in material properties and required loads. These factors usually lead to overly designed structures but can also lead to under-designed structures because they are poor estimates of uncertainty. To correctly quantify the uncertainty in a structure we use reliability-based methods to analyze a 2D truss.

This study first explores various types of methods used to calculate the reliability of an element to develop an automated analysis program. After finding the best methods needed for an accurate calculation of reliability, we define a set of random variables which affect the reliability of a structure. By developing a computationally automated framework to calculate the reliability of a 2D truss and its bar elements, we can gauge the efficiency and effectiveness of current design factors used. Additionally, we can also quantify the sensitivity of our analysis to its parameters to better understand the impact a single random variable can have in the overall calculation of reliability. Lastly, this reliability analysis framework can be used to conduct the reliability-based design of a steel bar member and a 2D truss system to optimize their probability of failure for various failure criteria.

Thesis Supervisor: Jerome J. Connor

Title: Professor of Civil and Environmental Engineering

Reliability-based Analysis and Design of 2D Trusses

by

Alexis Joseph Ludena

Submitted to the Department of Civil and Environmental Engineering
on May 9,2014, in Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING IN CIVIL AND ENVIRONMENTAL ENGINEERING

Abstract

Current safety factors used in structural design do not accurately account for uncertainties in material properties and required loads. These factors usually lead to overly designed structures but can also lead to under-designed structures because they are poor estimates of uncertainty. To correctly quantify the uncertainty in a structure we use reliability-based methods to analyze a 2D truss.

This study first explores various types of methods used to calculate the reliability of an element to develop an automated analysis program. After finding the best methods needed for an accurate calculation of reliability, we define a set of random variables which affect the reliability of a structure. By developing a computationally automated framework to calculate the reliability of a 2D truss and its bar elements, we can gauge the efficiency and effectiveness of current design factors used. Additionally, we can also quantify the sensitivity of our analysis to its parameters to better understand the impact a single random variable can have in the overall calculation of reliability. Lastly, this reliability analysis framework can then be used to conduct the reliability-based design of a bar element and 2D truss to optimize for the probability of failure for various failure criteria.

Thesis Supervisor: Jerome J. Connor

Title: Professor of Civil and Environmental Engineering

Acknowledgements

First of all, I would like to deeply thank the Massachusetts Institute of Technology for providing me with the best financial and educational resources I could ever imagine as undergrad. The institute has allowed me to experience life-changing moments which I am forever grateful. As for my development as a student, I would like to thank the Department of Civil and Environmental Engineering for introducing me to a close-knit family of faculty and students. Within the department, I'm especially grateful for Kris Kipp's constant help when navigating through my undergrad and graduate curriculum and always being there to answer any of my questions. As for faculty, I'm thankful for Prof. Jerome Connor's guidance since I was freshman all the way through the M.Eng program and Dr.Pierre Ghisbain, for spending countless hours advising the Steel Bridge team, introducing me to various interesting topics in structural engineering, including reliability, and for always being readily available to answer any of my questions. Outside of the department, I am extremely grateful to the Office of the Dean for Graduate Education, especially Patty Glidden, Dean Blanche Staton, and Dean Christine Ortiz for providing me with the necessary funding to complete my graduate degree.

Outside of the classroom, I'm fortunate to have developed lasting friendships with Ana Plascencia, who has helped me grow and mature as person while supporting me throughout most of my undergraduate career, my fraternity, Theta Delta Chi, which has provided me with the craziest, most-loving, creative experiences and people I could've asked for, and recently, Marwan Sarieddine, for introducing me to his radical theologies and style of living yet logical and weirdly rewarding experiences.

Lastly, all of my educational achievements, including this thesis, would not be possible if it wasn't for the utmost love and support I received throughout my lifetime from my parents Nancy and Virgilio, and my sisters Lisette and Tessi.

Table of Contents

1. Introduction 13

1.1. Motivation 13

1.2. Scope 13

2. Definition of Failure 15

2.1. Performance functions 15

2.1.1. Based on limit states 15

2.1.2. Performance functions based on serviceability limit state 15

2.2. Probability of Failure 16

2.2.1. General concept of probability of failure 16

2.2.2. Concept of Probability of Failure 17

2.3. Calculating the probability of failure 18

2.4. Reliability Index 19

3. Methods of Structural Reliability 24

3.1. Normally Distributed Margin 24

3.2. First Order Second Method (FOSM) 27

3.3. Hasofer-Lind Reliability Index 28

4. Reliability Analysis of a bar element 33

4.1. Applying FOSM and Hasofer-Lind 35

5. Reliability Based Design of a Bar Element 37

5.1. Buckling Limit State 37

5.2. Yielding Limit State 40

5.3. Relationships between Probability of Failures 41

6. Introduction to System Reliability 47

6.1. Failure Path 47

6.2. β-unzipping method 48

6.2.3. Level 0 49

6.2.4. Level 1 49

7. Reliability analysis of a 2D Truss 51

7.1. Analysis at Level 0 54

7.2. Analysis at Level 1 55

8. Reliability-based design of a 2D Truss **57**
9. Conclusion **59**
10. References **61**
11. Appendix **63**
11.1. FOSM Calculations 63
11.2. Hasofer-Lind Method Application 66
11.3. Hasofer-Lind Method for Buckling Limit State MATLAB Code 67
11.4. Hasofer-Lind Method for Yielding Limit State MATLAB code 69
11.5. Hasofer-Lind Method for Deflection Limit State MATLAB code 71
11.6. Calculation of Probability of Failure Surfaces using MATLAB 73
11.7. Plotting of Probability of Failure Surfaces using MATLAB 76
11.8. Supplementary code from *MATLAB Codes for Finite Element Analysis* 79
11.9. System reliability calculation of a 2D Truss using MATLAB 81

List of Figures

Figure 2.1-Probability density function 16
Figure 2.2-Overlapping resistance and stress distributions 19
Figure 2.3-CDF of Gaussian Distribution 20
Figure 2.4-Graphical representation of transformation 20
Figure 3.1-Limit State Equation Distribution 25
Figure 3.2-Taylor approximation of limit state equation 28
Figure 3.3-Bivariate distribution of resistance and stress variables 29
Figure 3.4-Standardized bivariate distribution 30
Figure 4.1-Steel bar subjected to axial load 33
Figure 5.1-Buckling Probability of Failure, *P=100 kips* 38
Figure 5.2-Cross Section Reference of Figure 5.1 39
Figure 5.3-Desired bar dimensions for a P_f=95% and P=100 kips 39
Figure 5.4-Yielding Probability of Failure 41
Figure 5.5-Yielding Probability of Failure 42
Figure 5.6-Intersection between limit state surfaces 43
Figure 5.7-Intersection between limit state surfaces 43
Figure 5.8-Boundaries between Buckling and Yielding limit states 44
Figure 5.9-Extruded boundaries between Buckling and Yielding limit states 45
Figure 6.1-Failure Tree 48
Figure 6.2-Series System 49
Figure 7.1-2D Pratt Truss 51
Figure 7.2-Local probabilistic axial loads on 2D truss 52
Figure 7.3-Deformation of 2D truss 53

List of Tables

Table 4.1-Reliabilities for different limit state equations using FOSM 35
Table 4.2-Reliabilities for different limit state equations using Hasofer-Lind Method 36
Table 4.3-Difference in reliabilities using different calculation methods 36
Table 7.1-Reliabillities of 2D truss' bar members 54
Table 7.2-Filtered bar members' reliabilities 55
Table 8.1 57

1. Introduction

1.1. Motivation

In the field of structural design, various safety factors are used to safely design structural systems. Methods include LRFD Factors and partial safety factors. Though these methods account for the probabilistic nature of material and load characteristics to develop safety factors, these factors are only used in deterministic analyses which don't take into account uncertainty. To develop a better understanding of the probabilistic nature of structural design we develop a framework to analyze the reliability of a 2D truss.

1.2. Scope

This thesis aims to computationally apply structural reliability methods on a 2D truss system in order to show its probability of failure for various cases at a component and system-wide level. This analysis can then be used to study the efficiency of current safety factors used in the design of trusses and also to see the sensitivity of the reliability of a truss to uncertainties in its geometry, material properties, and expected loads. With further development, the analysis can then be used to conduct a reliability-based design methodology for an element or truss.

2. Definition of Failure

To define the reliability of an element, we must first define failure. However, there are many definitions of failure because there are many ways to assess failure for an element. Each failure mode can have its own limit state function or performance based function which will have its own probability of failure. Overall, the failure of an element is subjective to the limit state to which it is evaluated against. Next we will talk about the most common types of limit state functions applicable to structural elements and introduce the mathematical meaning of failure.

2.1. Performance functions

2.1.1. Based on limit states

A performance function based on limit states, in the context of a structural member, can be seen as the strength or the resistance of the elements evaluated. It is basically an assessment of the capacity of the element to resist the loads acting on it. Performance functions for an element typically include the tensile strength when in tension or the buckling capacity when in compression.

2.1.2. Performance functions based on serviceability limit state

On the other hand, performance functions can be based on a serviceability limit state for a structure. This type of limit state function is basically dependent on the usage of the structure. As an example, the deflection of a certain element may not exceed a certain value (usually a coefficient multiplied by the span length). This a very difficult limit to impose since its formulation is very subjective to the users of the structures and not necessarily to the health of the structure.

2.2. Probability of Failure

2.2.1. General concept of probability of failure

To mathematically define failure, we first look at a random variable X_i following a probability density function (PDF) noted as $f_X(x)$ as shown in Figure 21.

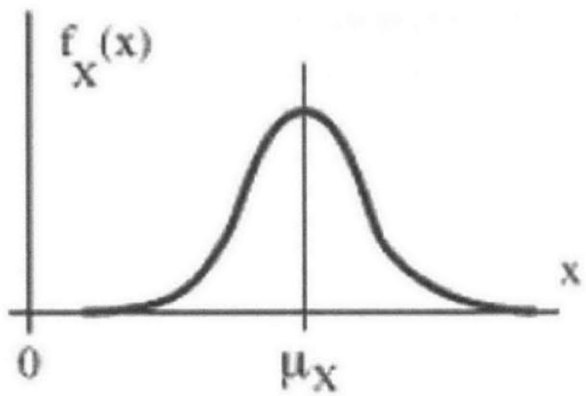

Figure 2.1-Probability density function

The mean or expected value of X can be calculated as follows:

$$\mu_X = \int x f_X(x)dx \tag{2.1}$$

Another important property of the PDF is its variance, which is a measure of the variability of X. It can be calculated as:

$$\sigma_X^2 = \int (x - \mu_X)^2 f_X(x)dx \tag{2.2}$$

where σ_X is the standard deviation.

Of interest is also finding the probability when $X < x$, calculated as:

$$P(X < x) = F_x(x) = \int_{-\infty}^{x} f_x(x)dx \tag{2.3}$$

Where $F_x(x)$ is called the cumulative density function (CDF).

2.2.2. Concept of Probability of Failure

We can now apply the definition of a random variable to calculate the reliability of a structural element. Focusing on a single structural element, we define its resistance R, or corresponding capacity, as a function of random variables:

$$R(X_1, X_2, X_3, \dots, X_n)$$

On the other hand, its stress or solicitation S, is defined as:

$$S(X_1, X_2, X_3, \dots, X_n)$$

Now, the limit state function G is formulated as:

$$G(X_1, X_2, X_3, \dots, X_n) = R(X_1, X_2, X_3, \dots, X_n) - S(X_1, X_2, X_3, \dots, X_n) \qquad (2.4)$$

since both R and S are in terms of random variables and consequently G is also in terms of those random variables.

We can then define the limit state equation which separates the acceptable region from the failure region as:

$$G(X_1, X_2, X_3, \dots, X_n) = 0$$

And failure is defined as:

$$G(X_1, X_2, X_3, \dots, X_n) < 0$$

or when

$$R(X_1, X_2, X_3, \dots, X_n) > S(X_1, X_2, X_3, \dots, X_n)$$

Failure can be seen as when the limit state equation is negative; meaning the value of the stress S is greater than the resistance R of the element. Therefore the probability of failure of the element can be calculated as:

$$P_f = P(G < 0) \qquad (2.5)$$

Methods to calculate P_f for various types of limit state functions will be discussed in Section 2.3.

2.3. Calculating the probability of failure

We can assume that both the resistance *R* and stress *S* follow arbitrary probability distributions such as the ones discussed in Section 2.2.1. We can define a particular value of the stress *S* at a given coordinate x_i as:

$$s_i = S(x_i)$$

Therefore, failure occurs when $s_i > R$ and the probability of failure can be calculated as the sum of the probabilities when $S = S_i$ and $R < s_i$ and can be noted as:

$$P_f = \sum_i P(S = s_i \cap R < s_i) \tag{2.6}$$

According to Bayes' Theorem

$$P(S = s_i \cap R < s_i) = P(R < S|S = s_i) \times P(S = s_i)$$

Therefore

$$P_f = \sum_i P(R < S|S = s_i) \times P(S = s_i)$$

Or expressed as an integral

$$P_f = \int_{-\infty}^{\infty} F_R(x) \times f_s(x)dx \tag{2.7}$$

Equation (2.7) is known as the convolution integral and can only be solved in a closed form for certain rudimentary cases. Numerical integration methods can be applied to solve the integral for various types of distributions. P_f can also be seen graphically in Figure 2.2.

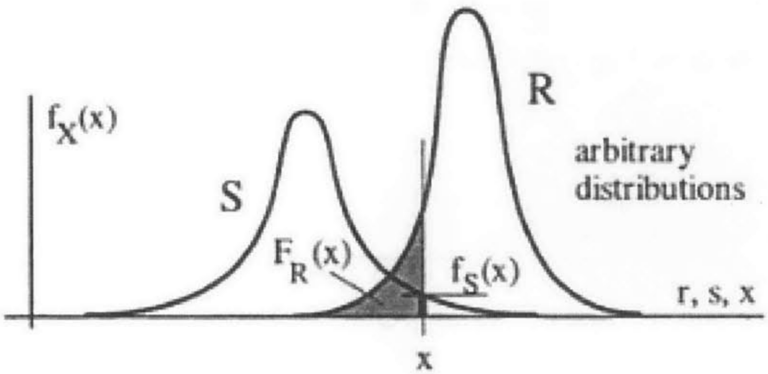

Figure 2.2-Overlapping resistance and stress distributions

2.4.Reliability Index

In the previous section we showed how to calculate the probability of failure, however, this value can be better shown by calculating the corresponding reliability index instead. The reliability index is a better way to quantify the safety of an element and can then be used to calculate the probability. We will now go through a basic example to show the use of the reliability index.

We first assume that the resistance of an element is just composed of the random variable *R*, where

$$R = N(\mu_R, \sigma_R)$$

Where $N(\mu_R, \sigma_R)$ means *R* is normally distributed with a mean resistance value of μ_R and standard deviation σ_R.

We then assume the stress *S* is deterministic and has a constant value of s_i and no standard deviation. Therefore the probability of failure for the element can be denoted as:

$$P_f = P\big((G = R - s_i) < 0\big) = P(R < s_i) = \Phi_R(s_i) \tag{2.8}$$

Where $\Phi_X(x)$ is the cumulative distribution function (CDF) of the Gaussian Distribution $F_X(x)$ and can be seen graphically on Figure 2.3 below.

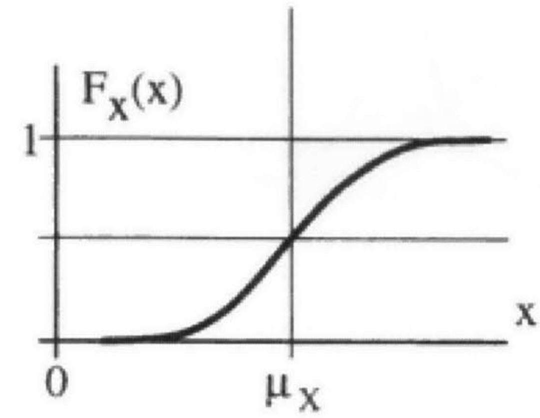

Figure 2.3-CDF of Gaussian Distribution

However, we can re-express the probability of failure of a random variable by transforming the random variable into the standard space. In the standard space, the random variable will have a mean of 0 and a standard deviation of 1. To achieve this, we apply the following transformation on a random variable *X* which can be seen in Figure 2.4:

$$X' = \frac{x - \mu_X}{\sigma_X} \tag{2.9}$$

Where $X' = N(0,1)$

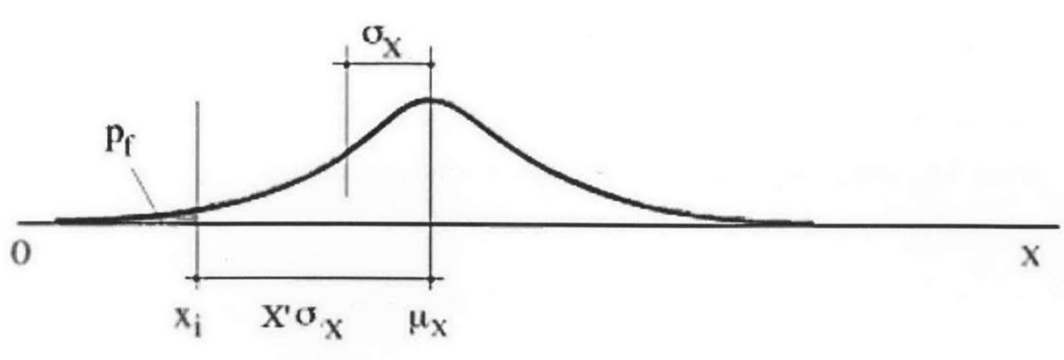

Figure 2.4-Graphical representation of transformation

Applying Equation 2.9 for a realization x_i of X , we obtain

$$X_i{}' = \frac{x_i - \mu_X}{\sigma_X}$$

And define the probability of failure as

$$P_f = P(X' < X_i') = \Phi_{X'}\left(\frac{x_i - \mu_X}{\sigma_X}\right)$$

From Figure 2.4 we can see that the distance between the realization and the mean of X can be expressed as a factor of the standard deviation as such:

$$\mu_X - x_i = \beta\sigma_X$$

Or

$$\beta = \frac{\mu_X - x_i}{\sigma_X} \qquad (2.10)$$

Where β is the reliability index and $\beta = -X_i{}'$. Therefore the probability of failure can now be expressed as:

$$P_f = \Phi_{X'}(-\beta) \qquad (2.11)$$

Applying Equations 2.10 and 2.11 to the element yields

$$\beta = \frac{\mu_R - s_i}{\sigma_R}$$

$$P_f = \Phi_{R'}(-\beta) = \Phi_{R'}\left(\frac{\mu_R - s_i}{\sigma_R}\right)$$

In this context, we can see the reliability index β as the factored distance between the mean resistance and mean stress. We have to be aware that this is just a very simplified way to calculate the reliability index and will dramatically vary based on the limit state functions used and the type of distributions of the random variables used. In the next sections we will cover methods for evaluating the reliability index for more complicated limit state functions.

3. Methods of Structural Reliability

3.1. Normally Distributed Margin

In the previous section we calculated the reliability index for a single variable. We will now calculate the reliability index for an element which has a limit state function composed of two random variables. We first assume that the resistance of the element is just composed of the random variable *R*, where

$$R = N(\mu_R, \sigma_R)$$

On the other hand the stress on the element is also defined as the random variable *S*, where:

$$S = N(\mu_S, \sigma_S)$$

We can now define the limit function as the safety margin *M*, where:

$$M = G = R - S \tag{3.1}$$

Where

$$\mu_M = \mu_R - \mu_S \tag{3.2}$$

$$\sigma_M = \sqrt{\sigma_R^2 + \sigma_S^2} \tag{3.3}$$

Assuming *R* and *S* are mutually independent.

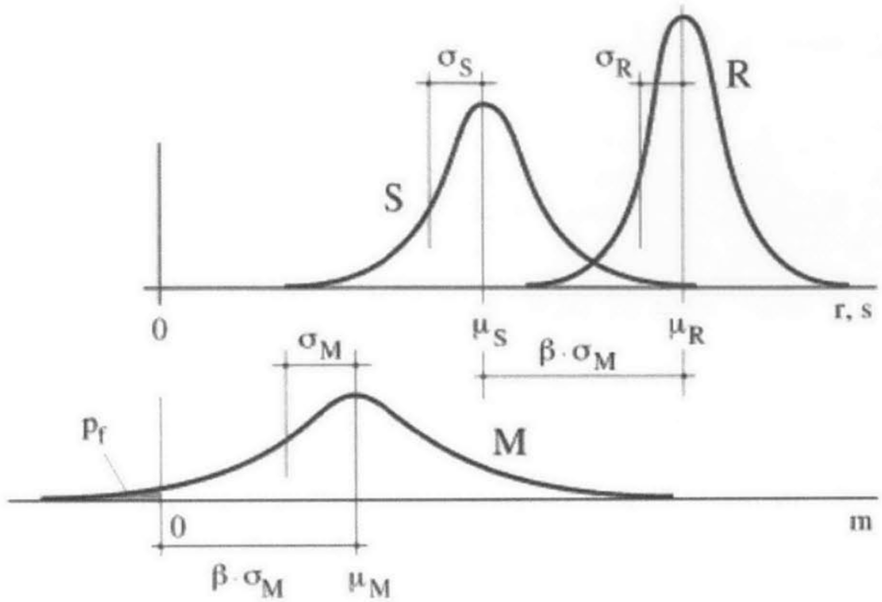

Figure 3.1-Limit State Equation Distribution

We can now calculate the index of reliability of the margin as

$$\beta = \frac{\mu_M}{\sigma_M} = \frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}} \tag{3.4}$$

$$P_f = \Phi(-\beta) \tag{3.5}$$

Of special interest are the weighting factors α_i which represent how much weight the standard deviation of each random variable has in the value of the probability of failure. These factors will have a geometric importance in later sections as we discuss the calculation of the reliability index. For *R* and *S*, the factors are as follows

$$\alpha_R = \frac{\sigma_R}{\sqrt{\sigma_R^2 + \sigma_S^2}} \qquad \alpha_S = \frac{\sigma_S}{\sqrt{\sigma_R^2 + \sigma_S^2}} \tag{3.6}$$

Where

$$\alpha_R^2 + \alpha_S^2 = 1$$

We can now use these weighting factors to develop a design condition for the element with the requirement $\beta \geq \beta_0$, where β_0 can be seen as the safety level. From Equation 3.4, we obtain:

$$\mu_R - \mu_S \geq \beta_0 \sigma_M$$

$$\mu_R - \mu_S \geq \beta_0 \frac{\sigma_R}{\sqrt{\sigma_R^2 + \sigma_S^2}} \sigma_R + \beta_0 \frac{\sigma_S}{\sqrt{\sigma_R^2 + \sigma_S^2}} \sigma_S$$

$$\mu_R - \mu_S \geq \beta_0 \alpha_R \sigma_R + \beta_0 \alpha_S \sigma_S$$

Reordering by similar terms

$$\mu_R - \beta_0 \alpha_R \sigma_R \geq \mu_S + \beta_0 \alpha_S \sigma_S \tag{3.7}$$

We can abbreviate this inequality as

$$r^* \geq s^* \tag{3.8}$$

Where r^* and s^* are the design values and are the coordinates of the design point which satisfy the safety requirement of β_0.

3.2. First Order Second Method (FOSM)

We now look to calculate the reliability index for a linear limit state function composed of n number of random variables and is expressed below:

$$G(X_1, X_2, X_3, \dots, X_n) = a_0 + a_1 X_1 + a_2 X_2 + a_3 X_3 + \dots + a_n X_n$$

The reliability index can now be calculated as

$$\beta = \frac{a_0 + \sum_{i=1}^{n} a_i \mu_i}{\sqrt{\sum_{i=1}^{n} (a_i \sigma_i)^2}} \tag{3.9}$$

assuming all of the random variables are uncorrelated.

However, this method of calculating the reliability index only works for limit state functions which are linear which is not always the case. To get around this, we can look to linearize a non-linear limit state function by

$$G(X_1, X_2, X_3, \dots, X_n) = G(x_i^*) + \sum_{i=1}^{n} (X_i - x_i^*) \frac{\partial G}{\partial X_i} |_* \tag{3.10}$$

Where x_i^* are the design points at which the Taylor Series is developed around and $\frac{\partial G}{\partial X_i}|_*$ corresponds to the derivative evaluated at the design point. This approximation can be graphically seen in Figure 3.2.

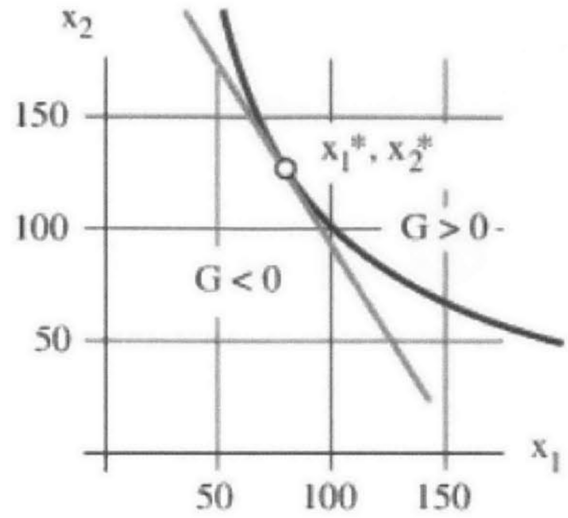

Figure 3.2-Taylor approximation of limit state equation

Once G is linearized, it will have the same form as Equation 3.10 and the reliability index can then be easily calculated using the mean values of X as design points.

In general, the FOSM method is a robust method for calculating the reliability index and will give inaccurate results for nonlinear margin functions since higher orders of the Taylor series are ignored. The major problem with this method is also its reliance on the form of the limit state function. For example, using the FOSM method yields different reliability indexes if the limit state function $R - E < 0$ is expressed as $R/E < 1$ instead. This problem of invariance in addressed and fixed in methods discussed in Section 3.3.

3.3.Hasofer-Lind Reliability Index

To solve the problem of invariance, we must first look at the reliability index from a geometrical standpoint. In Figure 3.3, we represent R and S as a two-dimensional joint probability density with a volume of 1. We then plot the linear limit state equation $G = R - S = 0$. We can see that the equation separates the density into a safe region, where $R > S$, and an unsafe region, where $R \leq S$ and failure occurs. Since failure begins at any point on the straight line, the design points $r^*$and s^* is where failure is most likely to occur because that's where the volume of the failure region is the greatest.

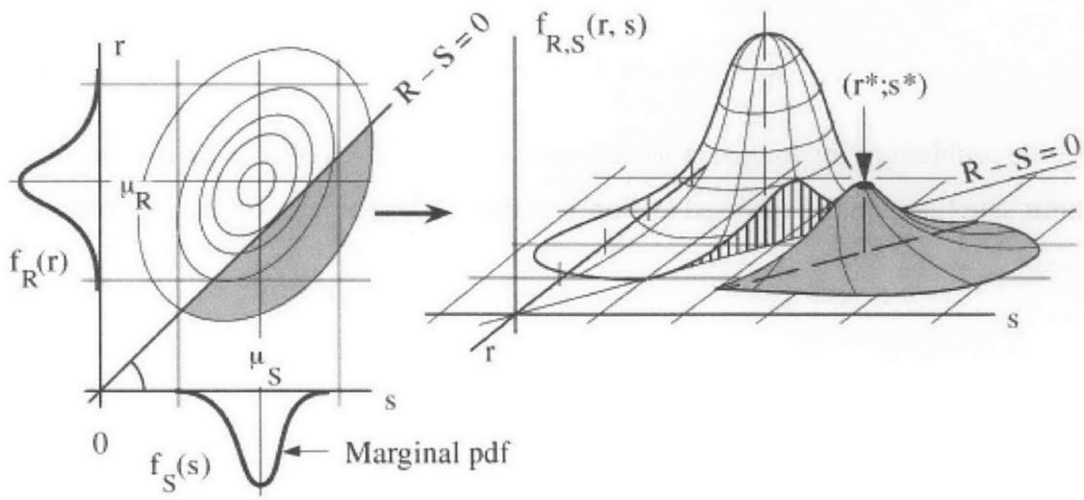

Figure 3.3-Bivariate distribution of resistance and stress variables

To account for the invariance problem discussed in the previous section, random variables *R* and *S* are standardized into variables U_1 and U_2 using Equation 2.9.

$$R = \mu_R + U_1\sigma_R \quad (3.11)$$

$$S = \mu_S + U_2\sigma_S \quad (3.12)$$

Now, as seen in Figure 3.4, the joint probability function is now centered and axially symmetric. The reliability index, can now be seen as the magnitude of the vector starting from the origin to the new design point U_1^* and U_2^* because it is the shortest distance between the origin and limit state surface yielding the greatest volume of the unsafe region and therefore the greatest probability of failure.

$$\beta = \sqrt{(U^*)^T(U^*)} \quad (3.13)$$

And the weight factors discussed in Section 3.1, can now be geometrically seen as the directional cosines of the vector **β** where

$$\alpha_i = \cos\theta_{U_i} = \frac{\sigma_i}{\sqrt{\sum_{i=1}^{n}\sigma_i^2}} \quad (3.14)$$

And the value of the design point as

$$u_i^* = -\beta \cos \theta_{U_i} = -\beta \alpha_i \tag{3.15}$$

Consequently, the problem of invariance is solved since the calculation of β is independent of the way the limit sate function is expressed and only relies on the geometry of the distribution which is constant.

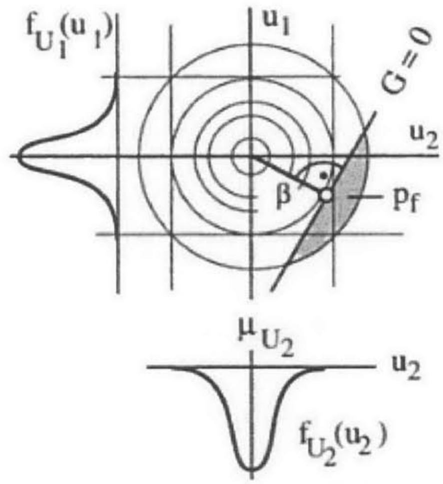

Figure 3.4-Standardized bivariate distribution

However, if the limit state function isn't linear, we must first standardize it and linearize it around the design points in order to calculate β. By doing so, we obtain the following formulas:

$$\beta = \frac{\sum_{i=1}^{n} u_i^* \frac{\partial G}{\partial U_i}|_*}{\sqrt{\sum_{i=1}^{n} \frac{\partial G}{\partial U_i}|_*^2}} \tag{3.16}$$

$$\alpha_i = \frac{\frac{\partial G}{\partial U_i}|_*}{\sqrt{\sum_{i=1}^{n} \frac{\partial G}{\partial U_i}|_*^2}} \tag{3.17}$$

Where $\frac{\partial G}{\partial U_i}|_*$ is the partial derivate of the standardized margin function evaluated at the design point u_i^*.

Additionally, the value of the original design point can be expressed as follows

$$x_i^* = \mu_{X_i} + u_i^* \sigma_{X_i} \tag{3.18}$$

Where

$$u_i^* = -\alpha\beta \tag{3.19}$$

$$x_i^* = \mu_{X_i} - \alpha\beta\sigma_{X_i} \tag{3.20}$$

As seen from the equations, calculating the reliability index is an iterative process. To find the reliability index we follow the algorithm below:

Step 1-Define a limit state function

Step 2-Make an initial guess for the design points u_i^* . Good estimate are the means of the random variables.

Step 3-Standardize random variables by using Equation

Step 4- Calculate $\frac{\partial G}{\partial U_i}|_*$ as well as the directional cosines α_i

Step 5- Find u_i^* in terms of β by using calculated value of α_i and Equation 3.19

Step 6- Insert u_i^* in terms of β into the limit state function and solve for β. Keep in mind that the limit state function evaluated at u_i^* is equal to zero.

Step 7- Use the new value of β from the previous step to calculate a new u_i^*

Step 8- With the new value of u_i^* calculate Steps 4 to 7 again until β converges.

Once β is calculated, we find the probability of failure as such

$$P_f = \Phi(-\beta)$$

4. Reliability Analysis of a bar element

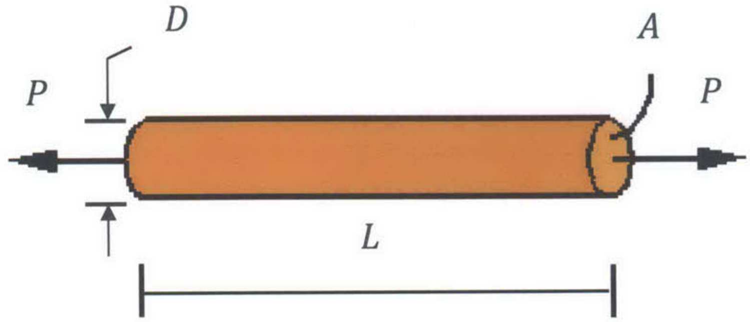

Figure 4.1-Steel bar subjected to axial load

Now that we have covered appropriate methods for evaluating structural reliability, we will apply them to a steel bar element. We first look to define four limit state functions for the bar element. G_0 is based on a buckling limit, G_1 is based on a yielding force limit, G_2 is based on a yielding stress limit, and G_3 is based on a deflection limit.

$$G_0(D,P) = F_{crit}(D) - P = \frac{EI(D)\pi^2}{L^2} - P = \frac{\pi^3 E D^4}{64L^2} - P = a_0 D^4 - P \tag{4.1}$$

$$G_1(D,P) = F_y(D) - P = A_b(D)\sigma_y - P = \frac{\pi D^2}{4} - P = a_1 D^2 - P \tag{4.2}$$

$$G_2(D,P) = \sigma_y - \frac{P}{A_b} = \sigma_y - \frac{4P}{\pi D^2} = \sigma_y - a_2 \frac{P}{D^2} \tag{4.3}$$

$$G_3(D,P) = u_{max} - u_{bar}(P) = \frac{L}{300} - \frac{PL}{EA_b(D)} = \frac{L}{300} - \frac{4PL}{E\pi D^2} = \frac{L}{300} - a_3 \frac{P}{D^2} \tag{4.4}$$

Where

$$a_0 = \frac{\pi^3 E}{64L^2}$$

$$a_1 = \frac{\pi}{4}$$

$$a_2 = \frac{4}{\pi}$$

$$a_3 = \frac{4L}{E\pi}$$

And

$P - Applied\ Axial\ Force$ $D - diameter\ of\ bar$ $E - Young's\ Modulus$

$A_b - cross\ sectional\ area\ of\ bar$ $L - length\ of\ bar$ $\sigma_y - steel\ yielding\ stress$

In the limit state functions, the random variables are the bar's diameter D and the applied axial load P since, in practice, the length L has negligible variability and can be taken as deterministic. The variables are normally distributed as follows

$$P = N(\mu_P, \sigma_P)$$

$$D = N(\mu_D, \sigma_D)$$

Notice that Equations 4.2 and 4.3 are the same limit state equation just expressed differently. Additionally, the first 3 limit state equations are derived from physical limit states and are objective to material or geometry properties whereas the last limit state equation is based on serviceability and is very subjective to the allowable deflection u_{max} .

4.1. Applying FOSM and Hasofer-Lind

We now use the FOSM method discussed in Section 3.2 for the following parameters

$$\mu_D = 2\ in \quad \sigma_D = 0.01\ in$$

$$\mu_P = 100\ kips \quad \sigma_P = 10\ kips$$

$$E = 29{,}000\ ksi \quad L = 47\ in \quad \sigma_y = 36\ ksi$$

These values were calculated to provide adequate resistance R against the solicitation or stress S of each limit equation. The calculations using the FOSM method are explained in detail in Appendix 11.1.

Using FOSM, the following reliabilities and probability of failures are

Limit State Equation	$G(u_D, u_P)$	α_D	α_P	β	P_f
Buckling	1.76 kips	0.898	-0.441	0.078	46.90%
Yielding-Force	13.10 kips	0.749	-0.662	0.868	19.28%
Yielding-Stress	4.17 ksi	0.707	-0.707	0.926	17.72%
Axial Deflection	0.079 in	0.749	-0.662	10.824	0.00%

Table 4.1-Reliabilities for different limit state equations using FOSM

Where $G(u_D, u_P)$ represents the current safety margin and is positive, showing that there is no failure if the variables weren't random. α_D and α_P are the weight factors showing the sensitivity of P_f to each value. Notice that a different P_f is calculated for the force and stress yielding equations even though they represent the same limit state but are just different algebraically. This invariance shows the robustness of the FOSM method and its high reliance on the way the limit state equation is expressed.

To get rid of the invariance problem and acquire a better approximation of the reliability index and P_f we now apply the Hasofer-Lind method.

Limit State Equation	$D^*(in)$	$P^*(kips)$	α_D	α_P	β	P_f
Buckling	1.99	100.3473	0.896	-0.445	0.078	46.89%
Yielding-Force	1.94	105.9056	0.738	-0.675	0.876	19.07%
Yielding-Stress	1.94	105.9045	0.738	-0.674	0.876	19.07%
Axial Deflection	1.44	130.889	0.876	-0.482	6.410	0.00%

Table 4.2-Reliabilities for different limit state equations using Hasofer-Lind Method

We can now see the percentage of error as

Limit State Equation	β_{FOSM}	$\beta_{H,LIND}$	% error
Buckling	0.078	0.078	0.43%
Yielding-Force	0.868	0.876	0.91%
Yielding-Stress	0.926	0.876	5.78%
Axial Deflection	10.824	6.410	68.86%

Table 4.3-Difference in reliabilities using different calculation methods

5. Reliability Based Design of a Bar Element

Now that we have calculated the reliability indexes, we can analyze their values for varying parameters of $\mu_D, \sigma_D, \mu_P, \sigma_{p,}$ and L. By doing this, we can gauge the sensitivity of the reliability to various factors which are not always certain. However, we can also apply this analysis to design the geometry of the steel bar to meet a given reliability index for a given axial load and limit state. In Section 5.1 and 5.2 we develop design methodologies for the buckling and yielding limit states. We ignore the deflection limit state since its calculation is very subjective to the allowable deflection.

5.1.Buckling Limit State

We can design the steel bar's diameter μ_D and length L to meet a certain probability of failure for a given probabilistic load. Assuming the following parameters for the analysis:

$$\sigma_D = 0.10 * \mu_D \tag{5.1}$$

$$\mu_P = 100\ kips \quad \sigma_P = 10\ kips$$

$$E = 29{,}000\ ksi \quad \sigma_y = 36\ ksi$$

Using the Hasofer-Lind method from Section 4.2 and the MATLAB code from the Appendix 7.2, we generate a 3D-plot, shown in Figure 5.1, of the probability of failure P_f according to the buckling limit state (Equation 4.1) as a function of the diameter μ_D and length L.

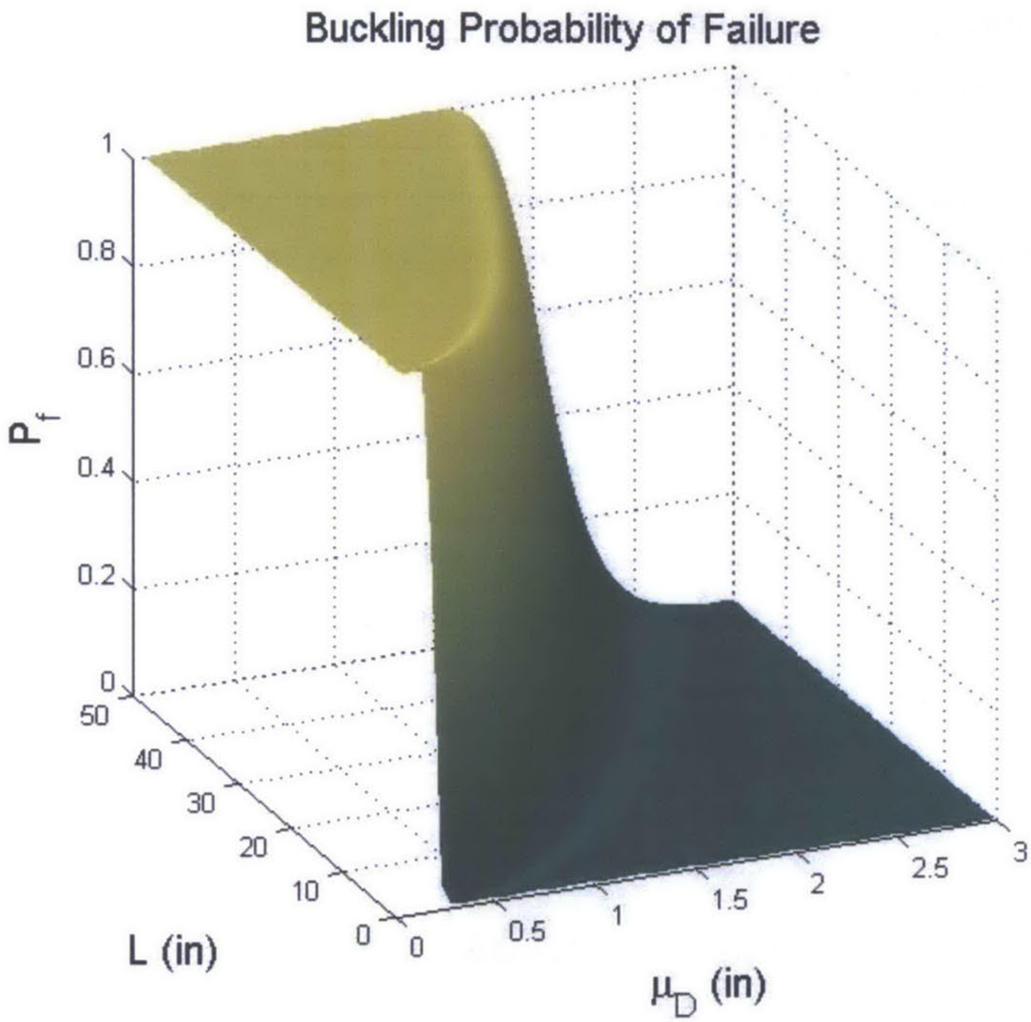

Figure 5.1-Buckling Probability of Failure, *P=100 kips*

The plot in Figure 5.1 resembles a CDF with varying parameters according to the length. From this plot, we notice that a bar with diameter of 1 in and length of 5 in has a P_f of less than 5%. However, since a longer length decreases the critical load needed for buckling, we see that a bar with the same diameter but a length of 25 in has a P_f of more than 90% as expected. Therefore, for any given probability of failure and load we can find a function outlining the relationship between D and L as seen in Figures 5.2 and 5.3.

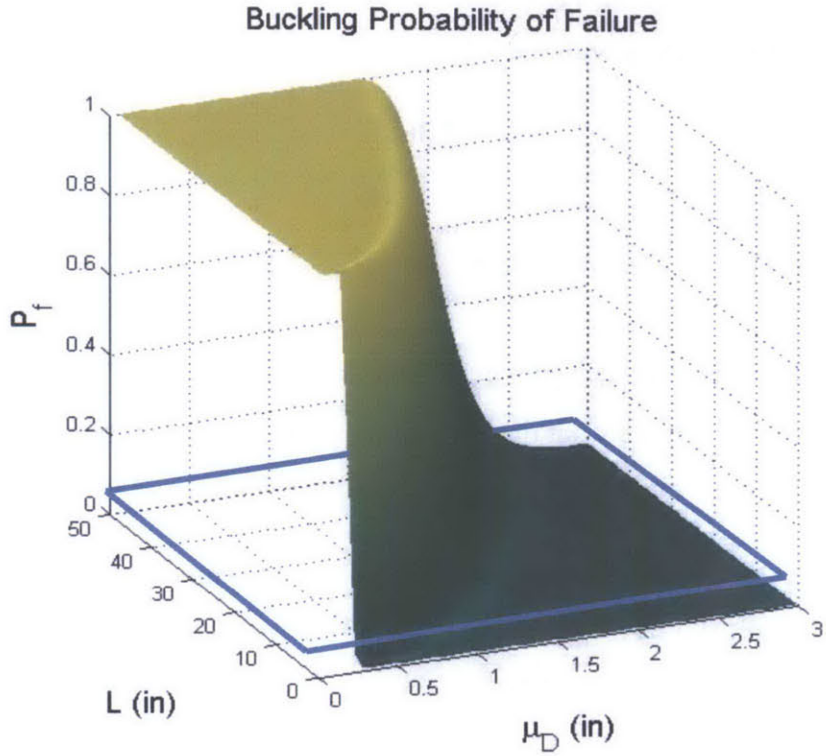

Figure 5.2-Cross Section Reference of Figure 5.1

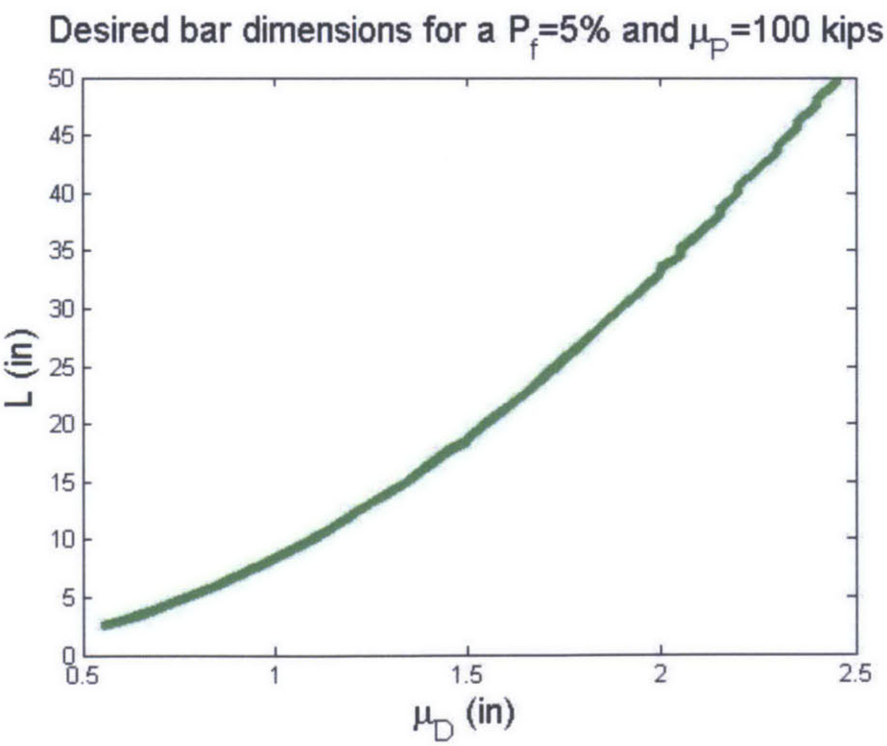

Figure 5.3-Desired bar dimensions for a P_f=95% and P=100 kips

From the cross-section in Figure 5.3, we see that the relationship between D and L can be approximated as linear relationship with the following equations:

$$L \approx 8.30\mu_D^2 \tag{5.2}$$

$$\mu_D \approx \sqrt{\frac{L}{8.30}}$$

Therefore, we can use this result to design the geometry of a bar to have a $P_f = 5\%$ for $\mu_P = 100\ kips$. For example, if we want the bar to have a length of 40 in then it will need to have a diameter of 4.26 in, according to Equation 4.6, to meet the desired P_f.

As you can see, this design methodology is extremely robust and can only be applied for a given load, probability of failure, and limit state. In Section 4.3 we will develop an easier methodology for reliability-based design for the bar element.

5.2. Yielding Limit State

Just as in Section 5.1, we derive a similar design methodology but for the yielding limit state (Equation 4.2) and $\mu_P = 100\ kips$. Since the bar's yielding is independent of the bar's length,we can generate a 2D plot to show the Yielding Probability of Failure as seen in Figure 4.5.

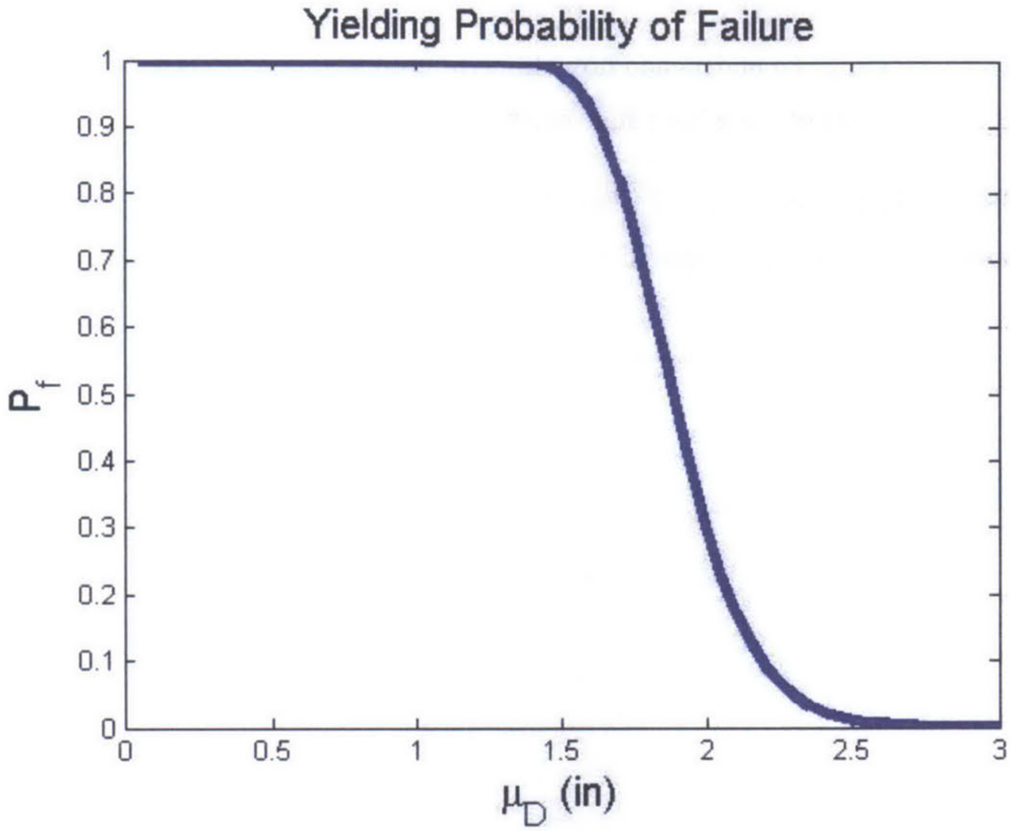

Figure 5.4-Yielding Probability of Failure

We can then fit this relationship with a polynomial to approximate P_f as a function of the diameter μ_D as shown in Equation

$$P_f = -1.0105\mu_D^3 + 8.2041\mu_D^2 - 22.076\mu_D + 19.713 \tag{5.3}$$

Using Equation 5.3, we can design a bar's diameter for a desired yielding probability of failure. For example, if we want a bar with $P_f = 5\%$, Equation 5.3 yields a diameter $\mu_D = 1.9\ in$ with $\mu_P = 100\ kips$.

5.3.Relationships between Probability of Failures

As seen in Section 4.1, different types of limit state functions yield different probability of failures for the same probabilistic geometry and loads of a bar. In Table 4.2, we see that the buckling probability of failure is 46% while the yielding probability of failure is 18% for a specific rod. This discrepancy is of interest because it shows that the governing failure mode for the given bar is currently in buckling. However, in practice, it is favorable for a structural element to yield before buckling. To understand how different limit state functions behave for different probabilistic inputs, we plot the limit state equations in the same design space.

First, we express the Yielding Probability of Failure (Figure 5.4), in a 3D design space with μ_D in the x-axis and L in the y-axis as seen in Figure 5.5.

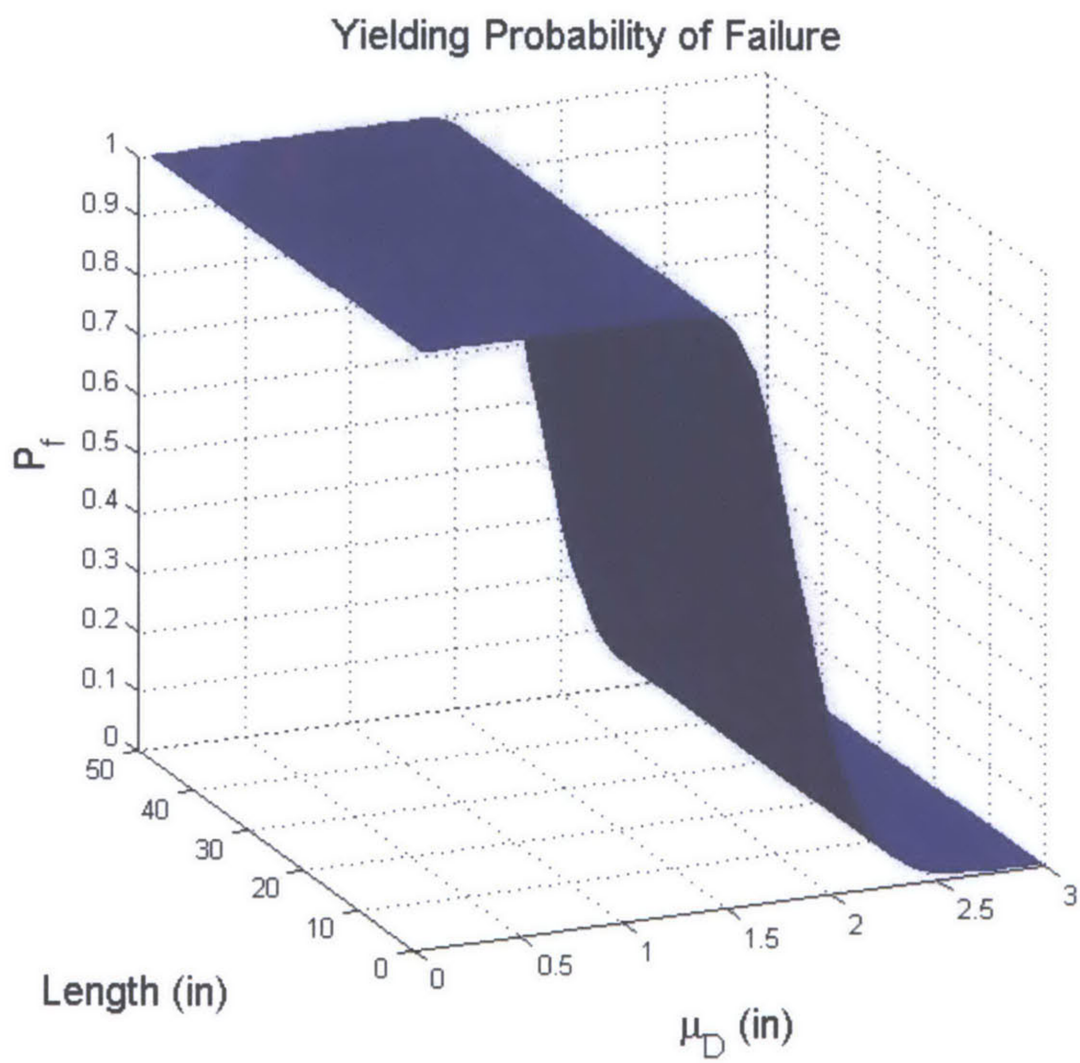

Figure 5.5-Yielding Probability of Failure

We then superimpose the Yielding Probability of Failure surface shown in Figure 5.5 on the same graph as the Buckling Probability of Failure shown in Figure 5.1. The result can be appreciated in Figures 5.6 and 5.7.

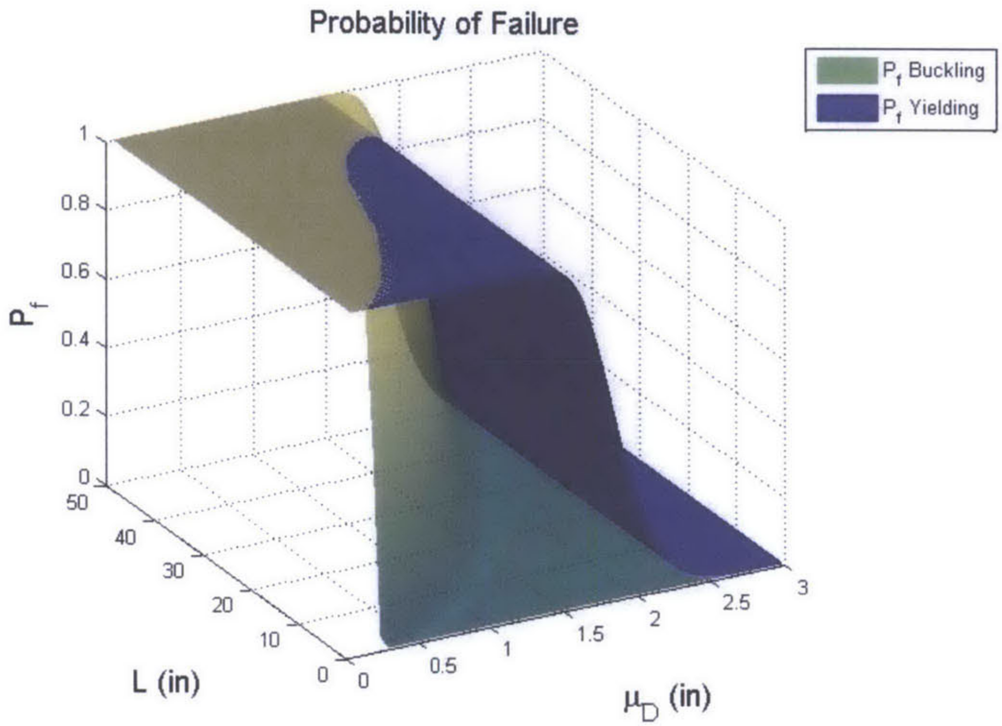

Figure 5.6-Intersection between limit state surfaces

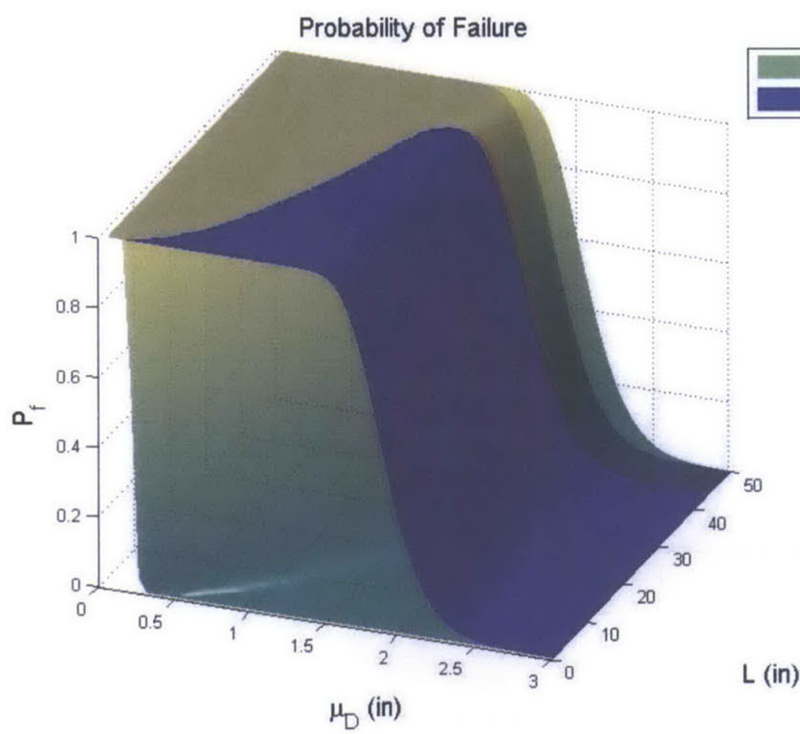

Figure 5.7-Intersection between limit state surfaces

Keeping in mind that Figures 5.6 and 5.7 were generated for $\mu\_P = 100\ kips$, we notice that P_f Yielding is higher or governs above P_f buckling for relatively short lengths ($L < 40\ in$) as expected. However after $L > 43\ in$ we see that P_f buckling begins to govern because of the long length of the bar.

If we look at the intersections between the two failure surfaces, we can find the boundary where one failure mode overtakes the over and focus on the relationship the length L and bar diameter μ_D have at this boundary. Though the intersection varies between *40 in* to *43 in*, it can be approximated as the 2D Yielding Probability of Failure shown in Figure 5.4.

Since this analysis was only done for an axial load with $\mu_P = 100\ k$, we repeat the boundary analysis for different values of μ_P ranging from 25 *kips* to *200 kips* and with $\sigma_P = .10\mu_P$.

By doing so, we create the graph shown in Figure 5.8.

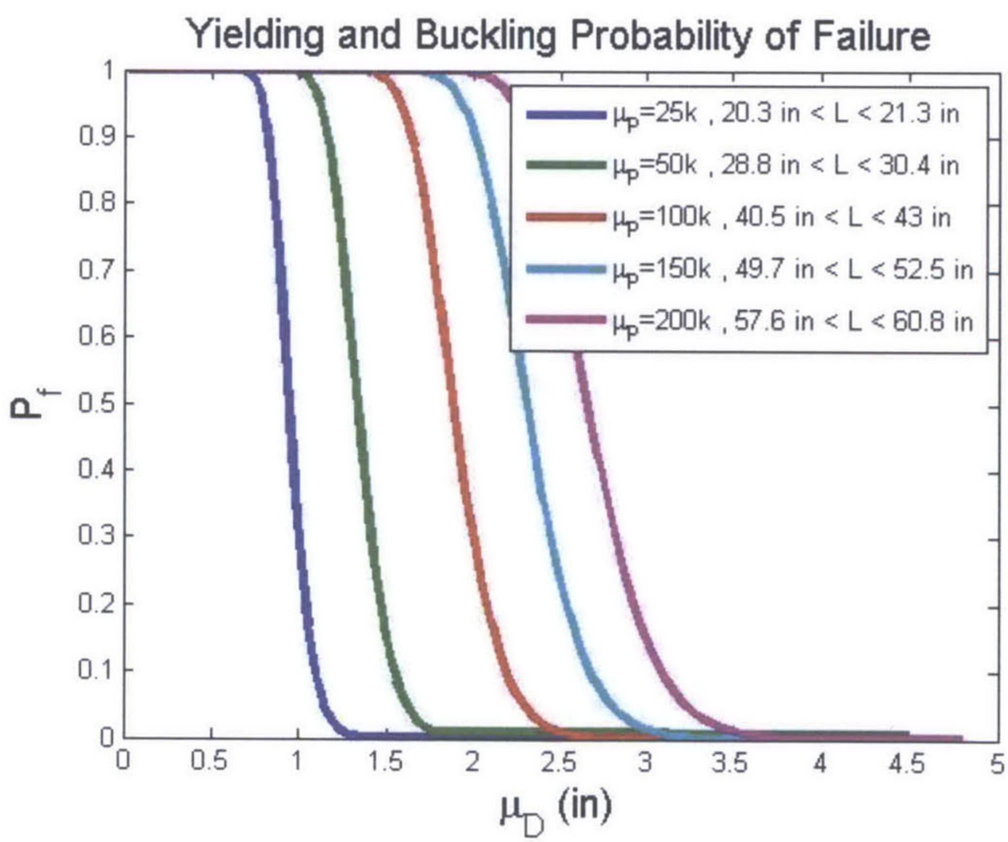

Figure 5.8-Boundaries between Buckling and Yielding limit states

Since the MATLAB code (Appendix 7) used to calculate P_f , takes about 15 minutes to solve per given axial load, we only sampled a group of 5 loads.

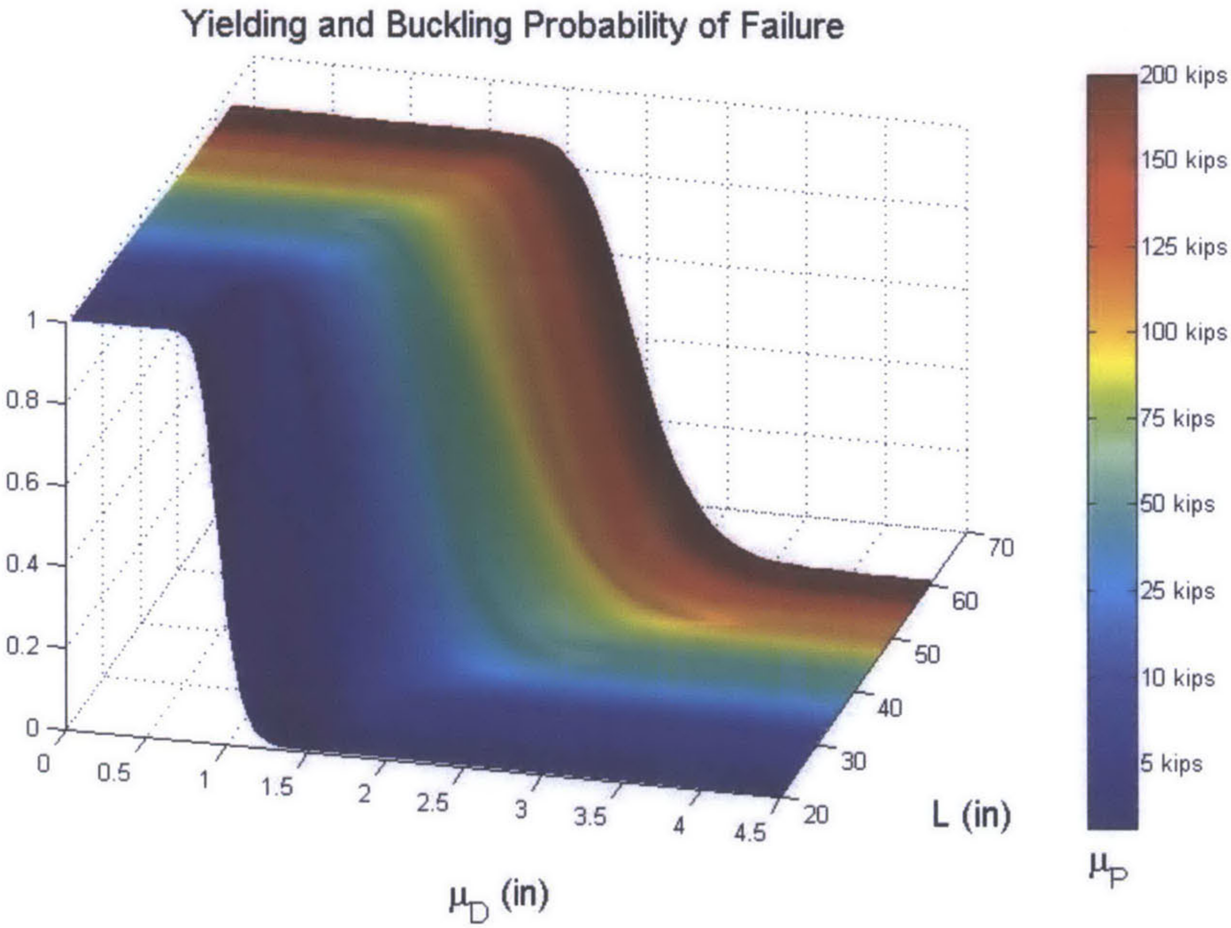

Figure 5.9-Extruded boundaries between Buckling and Yielding limit states

6. Introduction to System Reliability

Now that we have developed a clear understanding on how to analyze and evaluate the reliability of a single element, we will explore evaluating the reliability of a system composed of many elements. This calculation is of special interest, since the reliability of system is usually completely different than that of its elements. Sections 6.1 and 6.2 will introduce different methods used for evaluating the reliability of a system.

6.1.Failure Path

The failure path methodology defines system failure as a sequence of failure of its elements, where each set of failed elements can be defined as a cut-set event. We can demonstrate this methodology through a failure tree seen in Figure 6.1. In our tree, S_0 is the state where all of the elements have yet to fail and the system is fully functional.

If the system is damaged, its state can be shown as S_{ij} , where elements i and j have failed. Consequently, E_{ij}^k is defined as the event where element k fails next if elements i and j have failed.

In general, each node of the failure tree can be seen as the current state of the system with each branch being an event. At the first node of the tree, or undamaged state, various failure events can happen for elements of the system, with each failure event being a branch starting from the first node. At the end of each new branch there is a node and, if it's not a failure state, then more branches will emit to new nodes that might lead to a failure state or continue linking to new nodes. All of the branches move out to new nodes until they all eventually reach failure states which are called terminal nodes.

In Figure 6.1, we can see that the system has different failure sequences which can be seen as the paths from the initial node to the terminal nodes. Each of these failure sequences is a cut-set event, and the failure of the overall system can be defines as occurrence of any of the cut-sets. This is expressed mathematically in Equation 6.1.

$$P(any\ cut\ set\ event\ n) = P\left(\bigcup_{n} \bigcap_{n_1,n_2,\ldots n_i} E_{n_i}^{n_1,n_2,\ldots n_{i-1}}\right) \tag{6.1}$$

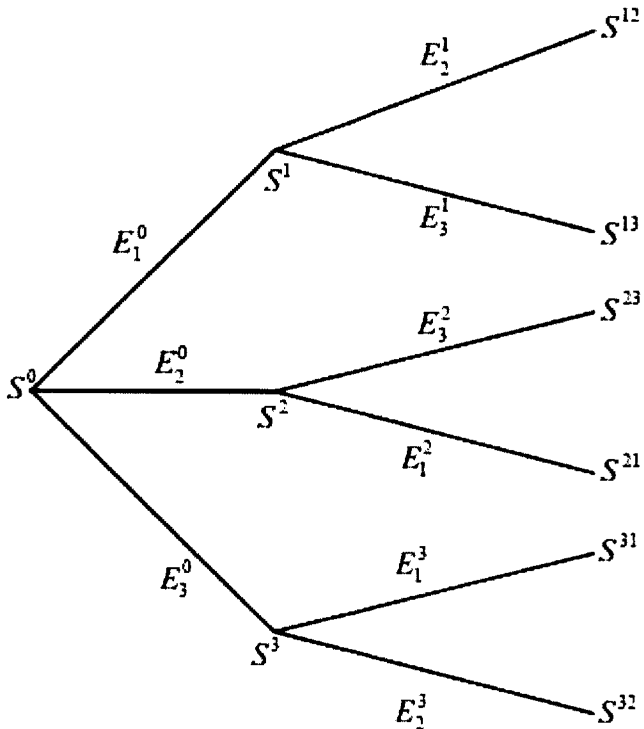

Figure 6.1-Failure Tree

6.2.β-unzipping method

The β-unzipping method is used to calculate the reliability of a system by using the reliability indices of each of the elements of the system. Using this method, a system is modeled as a series system, with the elements of the series system consisting of parallel systems. However, there are many levels of evaluation in the β-unzipping method which yield more defined calculations for the reliability of system. We will cover the first two levels of this method in the following subsections.

6.2.3. Level 0

To calculate our system reliability at level 0, we first calculate the reliability indices for all of the elements in the system. We then assume the system reliability is equal to the reliability of the element with the lowest reliability index as noted by Equation 6.1.

$$\beta_S^0 = \min_{i=1,2,\dots,n} \beta_i \tag{6.2}$$

Therefore, the system has a probability of failure equal to the highest probability of failure of one of its elements. As expected, this calculation is extremely robust and gives a highly conservative estimation of the system reliability index.

6.2.4. Level 1

At level 1, we assume that the failure of a single element can cause the failure of the whole system. In Figure 6.2, the whole system is modelled as a series system of n elements.

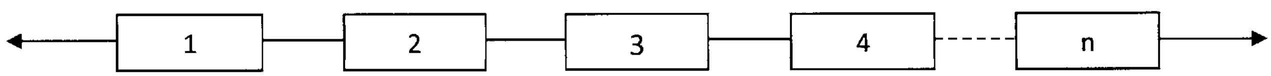

Figure 6.2-Series System

In this level, it's not required to use all n elements to estimate the system reliability. The system reliability can still be calculated with sufficient accuracy by including select failure elements. The elements can be used if they follow within the range $[\beta_{min}, \beta_{min} + \Delta\beta]$,where $\Delta\beta$ is an arbitrarily chosen positive number (the bigger $\Delta\beta$ is, the more accurate the result).

Now, to calculate the probability of failure for the series system, we first let X be a vector containing n random variables which are normally distributed.

$$\bar{X} = (X_1, X_2, \dots, X_n)$$

Then the probability of failure of our system can be approximated as the following linear combination,

$$P_f = P(\alpha_1 X_1 + \alpha_2 X_2 + \cdots + \alpha_n X_n \leq -\beta) = \Phi(-\beta) \tag{6.3}$$

Where $\bar{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ is a vector containing the corresponding directional cosines discussed in the Hasofer-Lind method in Section 4. Therefore, our linearized safety margin in the standardized normal space can be expressed as,

$$G_i(X) = \beta_i + \sum_{j=1}^{n} \alpha_{ij} X_j \tag{6.4}$$

Where, i refers to the i^{th} element and j to the j^{th} random input variable.

Now, we can calculate the probability of failure of our series system $P_{f_{system}}$ as

$$P_{f_{system}} = P\left(\bigcap_{i}^{k} G_i(X)\right) = P\left(\bigcap_{i}^{k} \alpha_i \bar{X} \le -\beta\right) = 1 - P\left(\bigcap_{i}^{k} -\alpha_i \bar{X} < \beta_i\right) = 1 - \Phi_k(\bar{\beta}, \bar{\rho}) \tag{6.5}$$

Where k is the total number of linearized limit states and $\bar{\beta} = \{\beta_1, \beta_2, \dots, \beta_k\}$ is a vector containing k number of reliability indices. Lastly, $\bar{\rho}$ is a vector containing the correlation matrix ρ_{ij} which is calculated as

$$\rho_{ij} = \bar{\alpha_i}^T \bar{\alpha_j} \qquad \text{for all } i \neq j \tag{6.6}$$

Because Equation 6.6 cannot be evaluated directly, we can use upper and lower bounds of the equation to calculate $P_{f_{system}}$. The upper and lower bounds are called Ditlevsen Bounds and are shown in Equation 6.6 below.

$$\max_{i=1,2,\dots,n} \left(\Phi(-\beta_i)\right) \le P_f \le 1 - \prod_{i=1}^{n} P_f = \Phi(\beta_i) \tag{6.7}$$

We use the lower bound for a series system if their elements are either fully correlated, meaning values of ρ_{ij} are closer to 1, and we use the upper bound if the elements are uncorrelated, meaning values of ρ_{ij} are closer to 0.

7. Reliability analysis of a 2D Truss

Now that we have developed a methodology for evaluating the reliability of a system, we look to apply the β-unzipping method to a 2D truss system composed of bar elements analyzed in Section 4. First we look at the following 2D truss shown in Figure 7.1

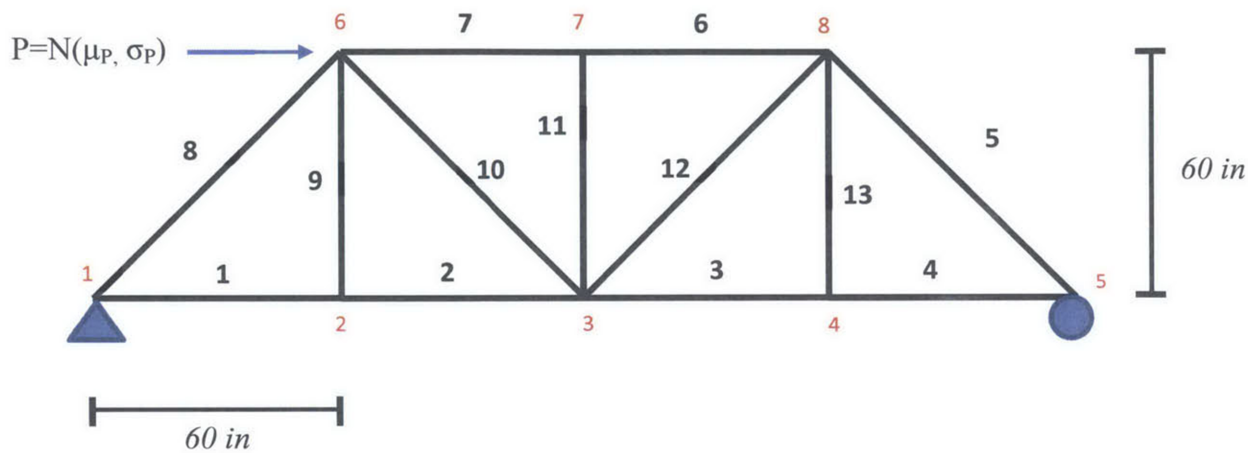

Figure 7.1-2D Pratt Truss

There is a probabilistic load P applied horizontally on node 6 of the truss, where

$\mu_P = 100\ kips$ and $\sigma_P = 10\ kips.$

The truss' steel bars have similar geometrical and probabilistic properties as those in Section 4, and have the following values,

$$D_i = N(\mu_D, \sigma_D)$$

$$\mu_D = 2\ in \qquad \sigma_D = 0.02\ in$$

$$L_i = \begin{cases} 60.0\ in, & i \neq 5,8,10,12 \\ 84.5\ in, & i = 5,8,10,12 \end{cases}$$

where i, is the number of the corresponding steel bar member.

D_i is the diameter of the steel bar following a normal distribution and has the same values for all of the steel bars in the truss. L_i is the length of the steel bar, The axial force at each bar element is noted as

$$F_{ij} = N(\mu_{F_{ij}}, \sigma_{F_{ij}})$$

$\mu_{F_{ij}}$ can be calculated for each bar element using the MATLAB code from [2] which uses a matrix stiffness method and, therefore, relies on D_{ij} to calculate the forces for indeterminate trusses.

Since our truss behaves linearly we know that $\mu_{F_{ij}} \propto \mu_P$ and $\sigma_{F_{ij}} \propto \sigma_P$, therefore, using the basic properties of a normal distribution, we can calculate $\sigma_{F_{ij}}$ once we obtain $\mu_{F_{ij}}$ from our MATLAB code as follows;

$$\sigma_{F_{ij}} = \frac{\mu_{F_{ij}}}{\mu_P} \cdot \sigma_P$$

By doing so, we generate the local normally-distributed axial forces of the truss which are shown in Figure 7.2.

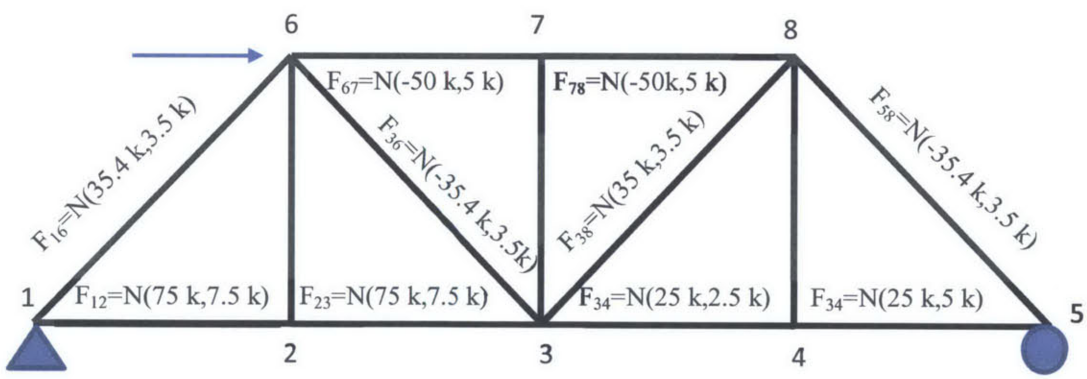

Figure 7.2-Local probabilistic axial loads on 2D truss

The MATLAB code used to generate the axial forces also calculates the local displacements and the deformation profile of the truss as seen in Figure 7.3.

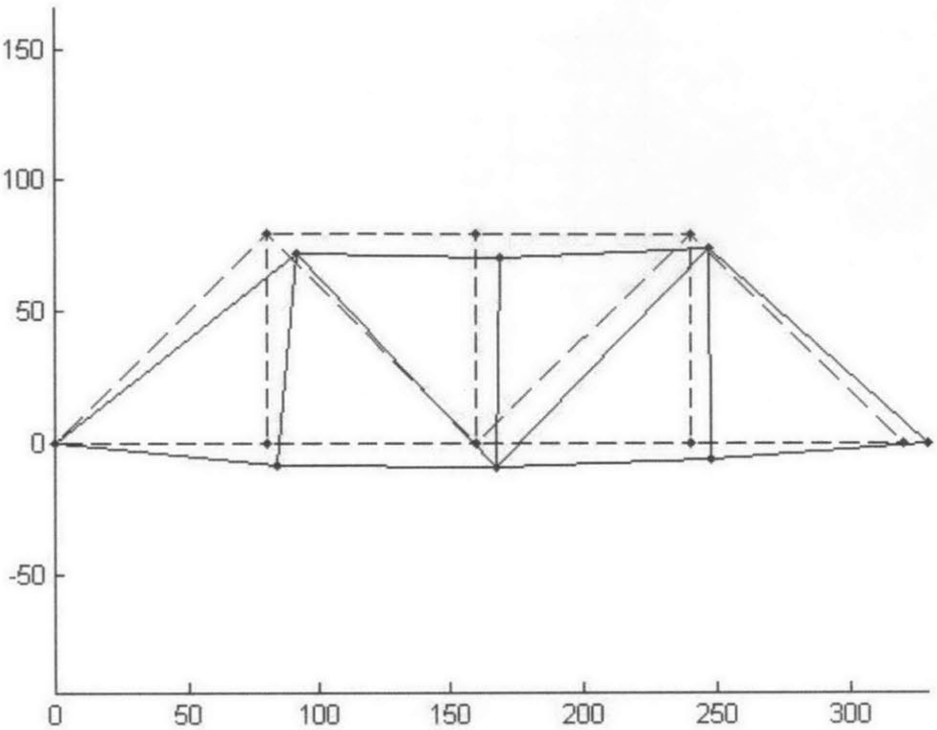

Figure 7.3-Deformation of 2D truss

Now that we have the local axial forces and their probabilistic properties, we can calculate the reliability of each member using the Hasofer-Lind Method applied in Section 4 for a yielding limit state. Our calculated local reliabilities are shown in Table 7.1.

Member	β	P_f
1	1.72	4.242%
2	1.72	4.242%
3	5.17	0.000%
4	5.17	0.000%
5	4.25	0.001%
6	3.19	0.072%
7	3.19	0.072%
8	4.25	0.001%
9	-	-
10	4.25	0.001%
11	-	-
12	4.25	0.001%
13	-	-

Table 7.1-Reliabillities of 2D truss' bar members

7.1.Analysis at Level 0

Using the methodology in Section 6.2.3 we can find the probability of failure of the truss by applying Equation 6.2. This yields

$$\beta_{truss} = 1.72$$

$$P_{f_{truss}} = \Phi(-\beta_{truss}) = 4.24\%$$

In this case, $P_{f_{truss}}$is equal to P_f of members 1 and 2 since they have the highest percentage of failure out of all the elements. This value of $P_{f_{truss}}$ is a robust estimate and is certainly not accurate. To get a more reliable answer, we must go into higher levels in the β-unzipping method.

7.2. Analysis at Level 1

We use the methodology developed in Section 6.2.4 to find $P_{f_{truss}}$ at Level 1 of the β-unzipping method. We first set $\Delta\beta = 2$ to narrow down the number of elements analyzed, giving us a range for β of [1.72, 3.72]. By doing so, our series system is now composed of the following members:

Member	β	P_f
1	1.72	4.242%
2	1.72	4.242%
6	3.19	0.072%
7	3.19	0.072%

Table 7.2-Filtered bar members' reliabilities

We now calculate our correlation matrix ρ using Equation 6.6 and acquire the following result:

$$\rho = \begin{bmatrix} 1 & 1 & 0.998 & 1 \\ 1 & 1 & 0.998 & 1 \\ 0.998 & 0.998 & 1 & 1 \\ 0.998 & 0.998 & 1 & 1 \end{bmatrix}$$

Since it is clear that ρ shows a high correlation between members, we use the lower bound of the Ditlevsen Bounds (Equation 6.7), which yields a truss probability of failure of

$$P_{f_{truss}} = \max_{i=1,2,\ldots,n} \left(\Phi(-\beta_i)\right) = 4.24\%$$

In this case we obtain the same $P_{f_{truss}}$ since our members are highly correlated which lead to a lower bound of Equation 6.5.

8. Reliability-based design of a 2D Truss

After developing a basic understanding on how to evaluate the reliability of a 2D truss system in Section 7, we can use this analysis combined with the design methodology of a steel bar member, developed in Section 5, to design the geometry of 2D truss to meet a certain system reliability.

First, we begin by listing the current local probabilistic loads for our 2D truss shown in Figure 7.2 alongside the local reliability indices listed in Table 7.1.

Member	μ_P (kips)	σ_P $(kips)$	β	P_f
1	75.0	7.5	1.72	4.242%
2	75.0	7.5	1.72	4.242%
3	25.0	2.5	5.17	0.000%
4	25.0	2.5	5.17	0.000%
5	-35.4	-3.5	4.25	0.001%
6	-50.0	-5.0	3.19	0.072%
7	-50.0	-5.0	3.19	0.072%
8	35.4	3.5	4.25	0.001%
9	0.0	0.0	-	-
10	-35.4	-3.5	4.25	0.001%
11	0.0	0.0	-	-
12	35.4	3.5	4.25	0.001%
13	0.0	0.0	-	-

Table 8.1-Local probabilistic loads for 2D truss

Utilizing Figures 5.8 and 5.9 used to design a steel bar, we aim to design the members of our 2D truss to have the same probability of failure for a yielding limit state. This is a favorable goal because it will make $P_{f_{truss}}$ equal to that of a single bar element for both Levels 0 and 1 of the β-unzipping methodology. By changing μ_D for all members to meet a $P_f = 5\%$ using Figures 5.8 and 5.9, we obtain the following values:

Member	μ_P (kips)	σ_P $(kips)$	μ_D (in)	σ_D (in)	$\Delta\mu_D$	P_f
1	75.0	7.5	1.95	0.20	2.5%	5.0%
2	75.0	7.5	1.95	0.20	2.5%	5.0%
3	25.0	2.5	1.40	0.14	30.0%	5.0%
4	25.0	2.5	1.40	0.14	30.0%	5.0%
5	-35.4	-3.5	1.36	0.14	32.0%	5.0%
6	-50.0	-5.0	1.60	0.16	20.0%	5.0%
7	-50.0	-5.0	1.60	0.16	20.0%	5.0%
8	35.4	3.5	1.36	0.14	32.0%	5.0%
9	0.0	0.0	1.35	0.14	32.5%	5.0%
10	-35.4	-3.5	1.36	0.14	32.0%	5.0%
11	0.0	0.0	1.36	0.14	32.0%	5.0%
12	35.4	3.5	1.40	0.14	30.0%	5.0%
13	0.0	0.0	1.36	0.14	32.0%	5.0%

Table 8.2-Design values for bar members of a 2D truss

9. Conclusion

After analyzing the reliability of a bar and a truss, we clearly see the heavy dependence of the probability of failure on the limit state equation used. Since failure is subjective to the definition of the limit state, even though the same probabilistic variables and values are used for all equations, the reliability indexes vary widely depending on the limit imposed. Focusing on the sensitivity of each reliability index, we also see that each random variable does not contribute equally to the calculation of failure, with the most significant variable having a higher order in the respective limit state equation.

Having built a program for reliability analysis, we notice that we can design the geometry of the elements to match desired levels of reliability for a limit state function. At times, we can also design the elements to have the same reliability for various limit state functions if there exists design values for the element.

10. References

1. Ferreira, A. J. M. *MATLAB Codes for Finite Element Analysis: Solids and Structures*. Dordrecht: Springer Science & Business Media, 2009. Print.
2. Murotsu, Y., H. Okada, K. Niwa, and S. Miwa. "Reliability Analysis of Truss Structures by Using Matrix Method." *Journal of Mechanical Design* 102.4 (1980): 749. Web.
3. Schneider, Jörg. *Introduction to Safety and Reliability of Structures*. Zürich, Switzerland: IABSE-AIPC-IVBH, 2006. Print.
4. Thoft-Christensen, Palle, and Michael J. Baker. *Structural Reliability Theory and Its Applications*. Berlin: Springer-Verlag, 1982. Print.
5. Thoft-Christensen, Palle, and Yoshisada Murotsu. *Application of Structural Systems Reliability Theory*. Berlin: Springer-Verlag, 1986. Print.
6. Tichý, Milík. *Applied Methods of Structural Reliability*. Dordrecht: Kluwer Academic, 1993. Print.
7. Zhao, Yan-Gang, and Tetsuro Ono. "Moment Methods for Structural Reliability." *Structural Safety* 23.1 (2001): 47-75. Web.

11. Appendix

11.1. FOSM Calculations

We define the limit state equations as

$$G_0(D,P) = F_{crit}(D) - P = \frac{EI(D)\pi^2}{L^2} - P = \frac{\pi^3 ED^4}{64L^2} - P = a_0 D^4 - P$$

$$G_1(D,P) = F_y(D) - P = A_b(D)\sigma_y - P = \frac{\pi D^2}{4} - P = a_1 D^2 - P$$

$$G_2(D,P) = \sigma_y - \frac{P}{A_b} = \sigma_y - \frac{4P}{\pi D^2} = \sigma_y - a_2 \frac{P}{D^2}$$

$$G_3(D,P) = u_{max} - u_{bar}(P) = \frac{L}{360} - \frac{PL}{EA_b(D)} = \frac{L}{360} - \frac{4PL}{E\pi D^2} = \frac{L}{360} - a_3 \frac{P}{D^2}$$

Where

$$a_0 = \frac{\pi^3 E}{64L^2}$$

$$a_1 = \frac{\pi}{4}$$

$$a_2 = \frac{4}{\pi}$$

$$a_3 = \frac{4L}{E\pi}$$

We now linearize the limit state functions by using the following formula

$$G(X_1, X_2, X_3, \ldots, X_n) = G(x_i^*) + \sum_{i=1}^{n} (X_I - x_i^*) \frac{\partial G}{\partial X_i}|_*$$

Now linearized, the functions take the following forms

$$G_0(D,P) = a_0 \mu_D{}^4 - \mu_P + 4a_0\mu_D^3(D - \mu_D) - (P - \mu_P)$$

$$G_1(D,P) = a_1 \mu_D{}^2 - \mu_P + 2a_1\mu_D(D - \mu_D) - (P - \mu_P)$$

$$G_2(D,P) = \sigma_y - a_2\frac{\mu_P}{\mu_D^2} + 2a_2\frac{\mu_P}{\mu_D^3}(D-\mu_D) - \frac{a_2}{\mu_D^2}(P-\mu_P)$$

$$G_3(D,P) = \frac{L}{360} - a_3\frac{\mu_P}{\mu_D^2} + 2a_3\frac{\mu_P}{\mu_D^3}(D-\mu_D) - \frac{a_3}{\mu_D^2}(P-\mu_P)$$

With means of

$$\mu_0 = G_0(\mu_D, \mu_P) = a_0{\mu_D}^4 - \mu_P$$

$$\mu_1 = G_1(\mu_D, \mu_P) = a_1{\mu_D}^2 - \mu_P$$

$$\mu_2 = G_2(\mu_D, \mu_P) = \sigma_y - a_2\frac{\mu_P}{\mu_D^2}$$

$$\mu_3 = G_3(\mu_D, \mu_P) = \frac{L}{360} - a_3\frac{\mu_P}{\mu_D^2}$$

And standard deviations of

$$\sigma_0 = \sqrt{(4a_0\mu_D^3\sigma_D)^2 + \sigma_P^2}$$

$$\sigma_1 = \sqrt{(2a_1\mu_D\sigma_D)^2 - \sigma_P^2}$$

$$\sigma_2 = \sqrt{\left(2a_2\frac{\mu_P}{\mu_D^3}\sigma_D\right)^2 - \left(\frac{a_2}{\mu_D^2}\sigma_P\right)^2}$$

$$\sigma_3 = \sqrt{\left(2a_3\frac{\mu_P}{\mu_D^3}\sigma_D\right)^2 - \left(\frac{a_3}{\mu_D^2}\sigma_P\right)^2}$$

Now the reliability indexes are as follows

$$\beta_0 = \frac{a_0{\mu_D}^4 - \mu_P}{\sqrt{(4a_0\mu_D^3\sigma_D)^2 + \sigma_P^2}}$$

$$\beta_1 = \frac{a_1{\mu_D}^2 - \mu_P}{\sqrt{(2a_1\mu_D\sigma_D)^2 - \sigma_P^2}}$$

$$\beta_2 = \frac{\sigma_y - a_2 \frac{\mu_P}{\mu_D^2}}{\sqrt{\left(2a_2 \frac{\mu_P}{\mu_D^3} \sigma_D\right)^2 - \left(\frac{a_2}{\mu_D^2} \sigma_P\right)^2}}$$

$$\beta_3 = \frac{\frac{L}{360} - a_3 \frac{\mu_P}{\mu_D^2}}{\sqrt{\left(2a_3 \frac{\mu_P}{\mu_D^3} \sigma_D\right)^2 - \left(\frac{a_3}{\mu_D^2} \sigma_P\right)^2}}$$

11.2. Hasofer-Lind Method Application

To apply the Hasofer-Lind method we first standardize the margin functions by applying the following formula

$$P = \mu_P + \sigma_P P'$$

$$D = \mu_D + \sigma_D D'$$

The limit functions are now expressed as

$$G_0(D',P') = a_0(\mu_D + \sigma_D D')^4 - (\mu_P + \sigma_P P')$$

$$G_1(D',P') = a_1(\mu_D + \sigma_D D')^2 - (\mu_P + \sigma_P P')$$

$$G_2(D',P') = \sigma_y - a_2 \frac{\mu_P + \sigma_P P'}{(\mu_D + \sigma_D D')^2}$$

$$G_3(D',P') = \frac{L}{360} - a_3 \frac{\mu_P + \sigma_P P'}{(\mu_D + \sigma_D D')^2}$$

Taking the partial derivatives

$$\frac{\partial G_0}{\partial D'} = 4a_0\sigma_D(\mu_D + \sigma_D D')^3 \quad \frac{\partial G_0}{\partial P'} = -\sigma_P$$

$$\frac{\partial G_1}{\partial D'} = 2a_1\sigma_D(\mu_D + \sigma_D D') \quad \frac{\partial G_1}{\partial P'} = -\sigma_P$$

$$\frac{\partial G_2}{\partial D'} = 2a_2\sigma_D \frac{\mu_P + \sigma_P P'}{(\mu_D + \sigma_D D')^3} \quad \frac{\partial G_2}{\partial P'} = -a_2 \frac{\sigma_P}{(\mu_D + \sigma_D D')^2}$$

$$\frac{\partial G_3}{\partial D'} = 2a_3\sigma_D \frac{\mu_P + \sigma_P P'}{(\mu_D + \sigma_D D')^3} \quad \frac{\partial G_3}{\partial P'} = -a_3 \frac{\sigma_P}{(\mu_D + \sigma_D D')^2}$$

11.3. Hasofer-Lind Method for Buckling Limit State MATLAB Code

```
%thesis

clear all

E=29000; %ksi
sig_y= 36; %ksi
L= 47; %in

A=(pi^3)*E/(64*L^2); %temp variable

u_D=2; %in INPUT
sd_D=.1; %in INPUT

u_P=100; %kip INPUT
sd_P=10; %kip INPUT


u_Di=u_D; %initial design point
u_Pi=u_P; %initial design point

alpha_D=0; %initializing sensitivity factor for D
alpha_P=0; %initializing sensitivity factor for P

n=0; %number of iterations

btemp=0;

k=true;

while k



alpha_D=(4*A*sd_D*(u_D+sd_D*u_Di)^3)/((4*A*sd_D*(u_D+sd_D*u_Di)^3)^2+sd_P^2)^
.5;

    alpha_P=-sd_P/((4*A*sd_D*(u_D+sd_D*u_Di)^3)^2+sd_P^2)^.5;

    syms b



    [b]=vpasolve(((A*(u_D+sd_D*(-alpha_D*b))^4-(u_P+sd_P*(-alpha_P*b))))==0,
b);

    b1=double(b);

    b_i=double(b(1));
```

```
    u_Di=-alpha_D*b_i;
    u_Pi=-alpha_P*b_i;

    if abs(btemp-b_i)<0.0001

      k=false;

    end

    btemp=b_i;
    n=n+1;

end



m=(A*(u_D+sd_D*u_Di)^4)-(u_P+sd_P*u_Pi)

D_design=u_D+sd_D*u_Di
P_design=u_P+sd_P*u_Pi
```

11.4. Hasofer-Lind Method for Yielding Limit State MATLAB code

```
%Hasofer-Lind, Yielding Limit State

clear all

E=29000; %ksi
sig_y= 36; %ksi
L= 47; %in

A=sig_y*pi/4; %temp variable

u_D=2; %in INPUT
sd_D=.1; %in INPUT

u_P=100; %kip INPUT
sd_P=10; %kip INPUT


u_Di=u_D; %initial design point
u_Pi=u_P; %initial design point

alpha_D=0; %initializing sensitivity factor for D
alpha_P=0; %initializing sensitivity factor for P

n=0; %number of iterations

btemp=0;

k=true;
while k



alpha_D=2*A*(u_D+u_Di*sd_D)*sd_D/((2*A*(u_D+u_Di*sd_D)*sd_D)^2+sd_P^2)^.5;
    alpha_P=-sd_P/((2*A*(u_D+u_Di*sd_D)*sd_D)^2+sd_P^2)^.5;

    syms b

    [b]=vpasolve(A*(u_D-alpha_D*b*sd_D)^2-(u_P-alpha_P*b*sd_P)==0,b);

    b_i=double(b(1));


    u_Di=-alpha_D*b_i;
    u_Pi=-alpha_P*b_i;

    if abs(btemp-b_i)<0.001

      k=false;
```

```
    end

    btemp=b_i;
    n=n+1;

end



%syms b
%solve((A*(u_D^2-alpha_D*sd_D*b)^2-(u_P*alpha_P*sd_P*b))==0,b);

m=A*(u_D+u_Di*sd_D)^2-(u_P+u_Pi*sd_P);

D_design=u_D+sd_D*u_Di
P_design=u_P+sd_P*u_Pi
```

11.5. Hasofer-Lind Method for Deflection Limit State MATLAB code

```
%Hasofer-Lind, Deflection Limit State

clear all

E=29000; %ksi
sig_y= 36; %ksi
L= 47; %in

A=4*L/(E*pi); %temp variable
u_max=L/360;

u_D=2; %in INPUT
sd_D=.1; %in INPUT

u_P=100; %kip INPUT
sd_P=10; %kip INPUT


u_Di=u_D; %initial design point
u_Pi=u_P; %initial design point

alpha_D=0; %initializing sensitivity factor for D
alpha_P=0; %initializing sensitivity factor for P

n=0; %number of iterations

btemp=0;

k=true;

while k


alpha_D=(2*A*sd_D*(u_P+sd_P*u_Pi)/(u_D+sd_D*u_Di)^3)/((2*A*sd_D*(u_P+sd_P*u_P
i)/(u_D+sd_D*u_Di)^3)^2+(-A*sd_P/(u_D+sd_D*u_Di)^2)^2)^.5;
    alpha_P=(-
A*sd_P/(u_D+sd_D*u_Di)^2)/((2*A*sd_D*(u_P+sd_P*u_Pi)/(u_D+sd_D*u_Di)^3)^2+(-
A*sd_P/(u_D+sd_D*u_Di)^2)^2)^.5;

    syms b



    [b]=vpasolve((u_max-A*(u_P+sd_P*(-alpha_P*b))/(u_D+sd_D*(-
alpha_D*b))^2)==0, b);

    b1=double(b);

    b_i=double(b(1));
```

```
    u_Di=-alpha_D*b_i;
    u_Pi=-alpha_P*b_i;

    if abs(btemp-b_i)<0.0001

      k=false;

    end

    btemp=b_i;
    n=n+1;

end



m=u_max-A*(u_P+sd_P*u_Pi)/(u_D+sd_D*u_Di)^2

D_design=u_D+sd_D*u_Di
P_design=u_P+sd_P*u_Pi
```

11.6. Calculation of Probability of Failure Surfaces using MATLAB

```
%Calculation for Buckling and Yielding Limit States surfaces

clear all

E=29000; %ksi
sig_y= 36; %ksi

for q=1:1:100

L1(q)= .35*q; %in47

L=L1(q);


a_1=sig_y*pi/4; %temp variable
a_0=(pi^3)*E/(64*L^2); %temp variable


for w=1:1:60

u_D1(q,w)=.03*w; %in INPUT


u_D=u_D1(q,w);

sd_D=.1*u_D; %in INPUT


u_P=25; %kip INPUT
sd_P=2.5; %kip INPUT


u_Di=u_D; %initial design point
u_Pi=u_P; %initial design point

alpha_D=0; %initializing sensitivity factor for D
alpha_P=0; %initializing sensitivity factor for P

n=0; %number of iterations

btemp=0;

k=true;
while k




alpha_D=2*a_1*(u_D+u_Di*sd_D)*sd_D/((2*a_1*(u_D+u_Di*sd_D)*sd_D)^2+sd_P^2)^.5
;
```

```
    alpha_P=-sd_P/((2*a_1*(u_D+u_Di*sd_D)*sd_D)^2+sd_P^2)^.5;

    syms b

    [b]=vpasolve(a_1*(u_D-alpha_D*b*sd_D)^2-(u_P-alpha_P*b*sd_P)==0,b);

    b_i=double(b(1));


    u_Di=-alpha_D*b_i;
    u_Pi=-alpha_P*b_i;

    if abs(btemp-b_i)<0.001

      k=false;

    end

    btemp=b_i;
    n=n+1;

end

betaforce(q,w)=b_i;


btemp=0;

j=true;

while j



alpha_D=(4*a_0*sd_D*(u_D+sd_D*u_Di)^3)/((4*a_0*sd_D*(u_D+sd_D*u_Di)^3)^2+sd_P
^2)^.5;

    alpha_P=-sd_P/((4*a_0*sd_D*(u_D+sd_D*u_Di)^3)^2+sd_P^2)^.5;

    syms b



    [b]=vpasolve(((a_0*(u_D+sd_D*(-alpha_D*b))^4-(u_P+sd_P*(-
alpha_P*b))))==0, b);

    b1=double(b);

    b_i=double(b(1));
```

```
    u_Di=-alpha_D*b_i;
    u_Pi=-alpha_P*b_i;

    if abs(btemp-b_i)<0.0001

      j=false;

    end

    btemp=b_i;
    n=n+1;

end

betabuckling(q,w)=b_i;

%syms b
%solve((A*(u_D^2-alpha_D*sd_D*b)^2-(u_P*alpha_P*sd_P*b))==0,b);

%m(w)=a_1*(u_D(w)+u_Di*sd_D)^2-(u_P+u_Pi*sd_P);

end

end

p_f_buckling=normcdf(-betabuckling);

surf(u_D1,L1,p_f_buckling,'FaceColor','interp',...
   'EdgeColor','none',...
   'FaceLighting','phong')
title('Buckling Probability of Failure','FontSize',15)
xlabel('Diameter (in)','FontSize',15)
ylabel('Length (in)','FontSize',15)
zlabel('P_f','FontSize',15)

hold on

surf(u_D1,L1,normcdf(-betaforce),'FaceColor','interp',...
   'EdgeColor','none',...
   'FaceLighting','phong')

[row,col]=find(p_f_buckling>.94 & p_f_buckling<.95);

for i=1:length(row)
sub_pf(i)=p_f_buckling(row(i),col(i));
sub_L(i)=L1(row(i));
sub_D(i)=u_D1(row(i),col(i));
end
```

11.7. Plotting of Probability of Failure Surfaces using MATLAB

```
colormap summer


p1=surf(u_D1,L1,p_f_buckling,'FaceColor','interp',...
    'EdgeColor','none',...
    'FaceLighting','phong');

alpha(0.8)

title('Probability of Failure','FontSize',15)
xlabel('\mu_D (in)','FontSize',15)
ylabel('L (in)','FontSize',15)
zlabel('P_f','FontSize',15)

hold on
camlight('right')



p_f_force=normcdf(-betaforce);


p2=surf(u_D1,L1,normcdf(-betaforce),'FaceColor','interp',...
   'EdgeColor','none',...
  'FaceLighting','phong');


set(p2,'FaceColor',[0 0 1],'FaceAlpha',1);

% title('Yielding Probability of Failure','FontSize',15)
% xlabel('Diameter (in)','FontSize',15)
% %ylabel('Length (in)','FontSize',15)
% ylabel('P_f','FontSize',15)
% camlight('right')
legend('P_f Buckling','P_f Yielding')

[row,col]=find(p_f_buckling>.05 & p_f_buckling<.06);

for i=1:length(row)
sub_pf(i)=p_f_buckling(row(i),col(i));
sub_L(i)=L1(row(i));
sub_D(i)=u_D1(row(i),col(i));
end
%
% plot(sub_L,sub_D)
% title('Yielding Probability of Failure','FontSize',15)
% ylabel('Diameter (in)','FontSize',15)
% xlabel('Length  (in)','FontSize',15)
%
n=1;
```

```
clear intx inty p3 int_pf int_L int_D i j k L_max L_min

for j=1:size(p_f_buckling,1)
    for k=1:size(p_f_buckling,2)

    if ((p_f_buckling(j,k)>=p_f_force(j,k))&&
(p_f_buckling(j,k)<(p_f_force(j,k)+.01)) && (p_f_buckling(j,k)<.97)&&
(p_f_buckling(j,k)>.1))
        intx(n)=j;
        inty(n)=k;
        n=n+1;
    end
    end

end



for i=1:length(intx)
int_pf(i)=p_f_buckling(intx(i),inty(i));
int_L(i)=L1(intx(i));
int_D(i)=u_D1(intx(i),inty(i));
end

L_max=max(int_L);
L_min=min(int_L);


p3=plot3(int_D,int_L,int_pf);
set(p3,'Color',[1 0 0]);

% pf_200k=p_f_force(1,:);
% D_200=u_D1(1,:);
% L_max200=L_max;
% L_min200=L_min;


figure
plot(D_25,pf_25k,D_50,pf_50k,D_100,pf_100k,D_150,pf_150k,D_200,pf_200k,'LineW
idth',3)
title('Yielding and Buckling Probability of Failure','FontSize',15)
xlabel('\mu_D (in)','FontSize',15)
ylabel('P_f','FontSize',15)
legend(['\mu_P=25k , ' num2str(L_min25,3) ' in < L < ' num2str(L_max25,3) '
in'],...
    ['\mu_P=50k , ' num2str(L_min50,3) ' in < L < ' num2str(L_max50,3) '
in'],...
    ['\mu_P=100k , ' num2str(L_min100,3) ' in < L < ' num2str(L_max100,3) '
in'],...
    ['\mu_P=150k , ' num2str(L_min150,3) ' in < L < ' num2str(L_max150,3) '
in'],...
    ['\mu_P=200k , ' num2str(L_min200,3) ' in < L < ' num2str(L_max200,3) '
in'])
```

```
L_bound=[L_min25 L_max25 L_min50 L_max50 L_min100 L_max100 L_min150 L_max150
L_min200 L_max200];

for i=1:60
L_bound2(:,i)=[25 25 50 50 100 100 150 150 200 200];

end

D_bound(1,:)=D_25;D_bound(2,:)=D_25;D_bound(3,:)=D_50;D_bound(4,:)=D_50;
D_bound(5,:)=D_100;D_bound(6,:)=D_100;D_bound(7,:)=D_150;D_bound(8,:)=D_150;
D_bound(9,:)=D_200;D_bound(10,:)=D_200;

pf_bound(1,:)=pf_25k;pf_bound(2,:)=pf_25k;pf_bound(3,:)=pf_50k;pf_bound(4,:)=
pf_50k;
pf_bound(5,:)=pf_100k;pf_bound(6,:)=pf_100k;pf_bound(7,:)=pf_150k;pf_bound(8,
:)=pf_150k;
pf_bound(9,:)=pf_200k;pf_bound(10,:)=pf_200k;

figure
colormap jet
surf(D_bound,L_bound,pf_bound,L_bound2,'FaceColor','interp',...
  'EdgeColor','none',...
 'FaceLighting','phong')

title('Yielding and Buckling Probability of Failure','FontSize',15)
xlabel('\mu_D (in)','FontSize',15)
ylabel('L (in)','FontSize',15)
zlabel('P_f','FontSize',15)

camlight('right')

c=colorbar('YTickLabel',{'5 kips', '10 kips','25 kips', '50 kips', '75
kips','100 kips','125 kips', '150 kips','200 kips',})

xlabel(c,'\mu_P','FontSize',15)

figure
plot(sub_D,sub_L,'LineWidth',4,'Color',[0 .8 0])
title('Desired bar dimensions for a P_f=5% and \mu_P=100 kips','FontSize',15)
xlabel('\mu_D (in)','FontSize',15)
ylabel('L (in)','FontSize',15)

figure
plot(u_D1(1,:),p_f_force(1,:),'LineWidth',4,'Color',[0 0 .8])
title('Yielding Probability of Failure','FontSize',15)
xlabel('\mu_D (in)','FontSize',15)
ylabel('P_f','FontSize',15)
```

11.8. Supplementary code from *MATLAB Codes for Finite Element Analysis*

```
%..........................................................

% MATLAB codes for Finite Element Analysis
% problem5.m
% antonio ferreira 2008

% clear memory
clear all

% E; modulus of elasticity
% A: area of cross section
% L: length of bar
E=70000;   A=300;    EA=E*A;

% generation of coordinates and connectivities
elementNodes=[ 1 2;1 3;2 3;2 4;1 4;4 7;3 4;3 6;4 5;4 6;3 5;5 6;6 7];
nodeCoordinates=[ 0 0;0 3000;3000 0;3000 3000;6000 0;6000 3000;3500 4000];
numberElements=size(elementNodes,1);
numberNodes=size(nodeCoordinates,1);
xx=nodeCoordinates(:,1);
yy=nodeCoordinates(:,2);

% for structure:
    % displacements: displacement vector
    % force : force vector
    % stiffness: stiffness matrix
GDof=2*numberNodes;
U=zeros(GDof,1);
force=zeros(GDof,1);
% applied load at node 2
force(4)=-20000;
force(8)=-80000;
force(12)=-50000;

% computation of the system stiffness matrix
 [stiffness]=...
    formStiffness2Dtruss(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,xx,yy,EA);

% boundary conditions and solution
prescribedDof=[1 2 10]';

% solution
displacements=solution(GDof,prescribedDof,stiffness,force);
us=1:2:2*numberNodes-1;
vs=2:2:2*numberNodes;

% drawing displacements

figure
L=xx(2)-xx(1);
%L=node(2,1)-node(1,1);
XX=displacements(us);YY=displacements(vs);
```

```
dispNorm=max(sqrt(XX.^2+YY.^2));
scaleFact=2*dispNorm;
clf
hold on
drawingMesh(nodeCoordinates+scaleFact*[XX YY],...
    elementNodes,'L2','k.-');
drawingMesh(nodeCoordinates,elementNodes,'L2','k.--');

% output displacements/reactions
outputDisplacementsReactions(displacements,stiffness,...
    GDof,prescribedDof)

% stresses at elements
stresses2Dtruss(numberElements,elementNodes,...
    xx,yy,displacements,E)
```

11.9. System reliability calculation of a 2D Truss using MATLAB

```
% clear memory
clear all

% E; modulus of elasticity, ksi
% A: area of cross section, in^2
% L: length of bar, in

E=29000;

% generation of coordinates and connectivities
elementNodes=[ 1 2;1 6;2 3;2 6;3 4;3 6;3 7;3 8; 4 5; 4 8;5 8;6 7;7 8];
nodeCoordinates=[ 0 0;80 0;160 0;240 0;320 0;80 80; 160 80; 240 80];


numberElements=size(elementNodes,1);
numberNodes=size(nodeCoordinates,1);
xx=nodeCoordinates(:,1);
yy=nodeCoordinates(:,2);

% for structure:
    % displacements: displacement vector
    % force : force vector
    % stiffness: stiffness matrix
GDof=2*numberNodes;
U=zeros(GDof,1);
force=zeros(GDof,1);
sdforce=zeros(GDof,1);

% applied load at node 2
force(11)=100;

sdforce(11)=10;

%diameters
diameter=zeros(numberElements);
sddiameter=zeros(numberElements);

diameter(:)=1.36; %in
diameter(1)=1.95;
diameter(3)=1.95;
diameter(5)=1.4;
diameter(8)=1.4;
diameter(9)=1.4;
diameter(12)=1.6;
diameter(13)=1.6;

sddiameter(:)=.1*diameter(:); %in

A=(pi/4)*(diameter.^2);

% computation of the system stiffness matrix
```

```
 [stiffness]=...
    formStiffness2Dtruss2(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,xx,yy,E,A);

% boundary conditions and solution

prescribedDof=[1 2 10]';

j=find(force~=0);

coeff=zeros(size(force,1),numberElements);

for n=1:size(force,1)
forcetemp=zeros(size(force));

if(force(n)~=0)
forcetemp(n)=force(n);
% solution
displacements=solution(GDof,prescribedDof,stiffness,forcetemp);

% stresses at elements
sigma=stresses2Dtruss(numberElements,elementNodes,...
    xx,yy,displacements,E);

coeff(n,:)=(A(round(n/2))/force(n))*sigma;

end
end
% drawing displacements

displacements=solution(GDof,prescribedDof,stiffness,force);

% stresses at elements
sigma=stresses2Dtruss(numberElements,elementNodes,...
    xx,yy,displacements,E);


us=1:2:2*numberNodes-1;
vs=2:2:2*numberNodes;

figure
L=xx(2)-xx(1);
%L=node(2,1)-node(1,1);

XX=displacements(us);YY=displacements(vs);
dispNorm=max(sqrt(XX.^2+YY.^2));
scaleFact=200*dispNorm;
clf
hold on
drawingMesh(nodeCoordinates+scaleFact*[XX YY],...
    elementNodes,'L2','k.-');
drawingMesh(nodeCoordinates,elementNodes,'L2','k.--');

% output displacements/reactions
```

```
outputDisplacementsReactions(displacements,stiffness,...
    GDof,prescribedDof)


%reliability of each element


%[beta,rowij,m_i]=bucklingReliability(numberElements,elementNodes,...
   %xx,yy,E,sigma,sdforce,coeff,2,.1)

 [beta,rowij,m_i]=forceReliability2(numberElements,sigma,...
     sdforce,coeff,diameter,sddiameter,36)

%---Level 0----



level0=normcdf(-min(beta))

%---Level 1---

deltaB=1;


k=find(beta>(min(beta)+deltaB));

beta1=beta;
beta1(k)=[];

rowij2=rowij;

rowij2(:,k)=[];
rowij2(k,:)=[];


temp=reshape(triu(rowij2,+1),1,[]);
temprow1=temp(find(temp~=0));

temp2=reshape(tril(rowij2,-1),1,[]);
temprow2=temp2(find(temp2~=0));

failure=normcdf(-beta)'

if (mean(temprow1)+mean(temprow2))/2 > .8

level1=max(normcdf(-beta1))

else

level1=1-sum(normcdf(beta1))

end
```

Accompanying function forceReliability.m

```
function
[b_i,rowij,m_i]=forceReliability2(numberElements,stresses,forcesd,coeff,diame
ter,sddiameter,sig_y )

SD=(forcesd.^2)'*(coeff.^2);
A=sig_y*pi/4; %temp variable


for e=1:numberElements

u_D=diameter(e);
sd_D=sddiameter(e);

u_P=abs(stresses(e))*pi*(u_D^2)/4; %kip INPUT

sd_P= (SD(e))^.5;

%sd_TP*u_P/u_TP ; %kip INPUT


u_Di=u_D; %initial design point
u_Pi=u_P; %initial design point

alpha_D=0; %initializing sensitivity factor for D
alpha_P=0; %initializing sensitivity factor for P

n=0; %number of iterations

btemp=0;

b_i(e)=0;

k=true;

if (u_P==0)
k=false;
b_i(e)=50;
end

while k


    alpha_D=2*A*(u_D+u_Di*sd_D)*sd_D/((2*A*(u_D+u_Di*sd_D)*sd_D)^2+sd_P^2)^.5
    alpha_P=-sd_P/((2*A*(u_D+u_Di*sd_D)*sd_D)^2+sd_P^2)^.5

    syms b

  [burr]=vpasolve(A*(u_D-alpha_D*b*sd_D)^2-(u_P-alpha_P*b*sd_P)==0,b);

    b1=double(burr)
```

```
    b_i(e)=b1(1);


    u_Di=-alpha_D*b_i(e)
    u_Pi=-alpha_P*b_i(e)


    if abs(btemp-b_i(e))<0.001

      k=false;

    end


    btemp=b_i(e);
    n=n+1;

end

alpha(1,e)=alpha_D;
alpha(2,e)=alpha_P;

m_i(e)=A*(u_D+u_Di*sd_D)^2-(u_P+u_Pi*sd_P);


end

rowij=alpha'*alpha;
```