ADYTRACK: A MODEL FOR STRUCTURAL ANALYSIS OF RAILROAD TRACKBED USING RANDOM FINITE ELEMENT METHOD

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Asif Arshid

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Civil and Environmental Engineering

November 2019

Fargo, North Dakota

North Dakota State University

Graduate School

Title

ADYTRACK: A MODEL FOR STRUCTURAL ANALYSIS OF RAILROAD TRACKBED USING RANDOM FINITE ELEMENT METHOD

By

Asif Arshid

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Ying Huang
Chair

Dr. Denver Tolliver (Co-Advisor)

Dr. Dinesh Katti

Dr. Annie Tangpong

Dr. Pan Lu

Dr. Ravi Yellavajjala

Approved:

November 15, 2019
Date

Dr. David R. Steward
Department Chair

ABSTRACT

Railroads are playing pivotal role to the economic growth of United States and trackbeds ensure their safe and smooth operations. However, reliable trackbed performance prediction has always been challenging due to many reasons, for instance materials characterization, deteriorations of materials and geometries due to railways operation and environmental changes etc. All these factors exhibit varying levels of intrinsic variabilities and uncertainties. These variations and uncertainties are completely ignored in most of the state-of-the-practice problems due to lack of availability of robust models that can characterize variations in materials, geometries, and/or loadings. In this study, a Random Finite Element based three-dimensional numerical model, named ADYTrack, is developed for structural analysis of railroad trackbeds. Uniqueness of this model is the inclusion of materials' intrinsic variabilities, geometric imperfections and/or uncertainties in axle loadings. The ADYTrack results, when compared with the analytical solution of a cantilever beam model, produced a maximum percentage difference of 0.7%; and 6% difference when compared with ANSYS software results for a single layer trackbed model; and a range of 5-20% difference was observed when validated against the actual field measurements. Sensitivity studies using RFEM based ADYTrack revealed that with the increasing variations in input parameters, measured by coefficient of variations (COV), the variations in output parameter also increased, and generally followed a bilinear trend with first linear component relatively insensitive up to around 30% COV of input parameters. However, beyond this limit, a considerable increase was observed in COVs of output parameters. For a COV of 80% in subgrade resilient modulus, a COV of 65% in vertical stress at the top of subgrade layer was observed. Additionally, the performance of any substructure layer found to be more sensitive to the variations in its own resilient modulus values. Furthermore, resilient modulus of subgrade layer

was found to be the most influential input parameter, as revealed by many other studies, and so was its variations. To conclude, ADYTrack model can serve as a robust supplemental tool for railroad trackbed analysis, especially at locations that exhibit higher degrees of uncertainties and thus pose higher risk of public or infrastructure safety.

ACKNOWLEDGEMENTS

Most of the studies as part of this dissertation are conducted at the Upper Great Plains Transportation Institute (UGPTI) at North Dakota State University (NDSU) under the advising of Dr. Ying Huang and co-advising of Dr. Denver Tolliver. Dr. Huang has always supported me on academic, personal, and professional fronts. Dr. Tolliver always appreciated my ideas and provided a valuable feedback and encouraged me in attending and presenting at various conferences.

The financial supports received from USDOT Mountain Plains Consortium (MPC) through UGPTI is gratefully acknowledged.

My appreciations extend to my colleagues at the North Dakota State University for their friendship, company and support. Special thanks go to Dr. Ravi Yellavajjala, Assist. Prof. at NDSU for his help and suggestions on this thesis. In addition, I am thankful to my committee members Dr. Dinesh Katti, Dr. Annie Tangpong, Dr. Pan Lu, and Dr. Ravi Yellavajjala for their time to review and feedback on my dissertation.

I am deeply thankful for my parents, who always pampered me emotionally and psychologically during ups and downs of this journey of PhD. The unconditional love of my wife and two sons gave me strength and a sense of purpose of keep moving.

DEDICATION

To my dearest parents, Arshad Mahmood and Shamim Arshad; O Lord, have mercy on them as they brought me up in my childhood.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDIX FIGURES

1. INTRODUCTION

1.1. Research Background

Railroad has always been considered as an efficient mode of transportation due the fact that it reduces friction between the moving wheels and rails. This consideration led the invention of animal driven carts and wagons and their use date back to 6th century BC in Greece. The invention of steam engine laid the foundation of modern days' railways in early 19th century in Great Britain. Railways soon became the most preferable mood of transportation, in big metropolitans for public transportation and in industries like mining or ports for freight. Even in modern days' transportation, railway is the most fuel efficient, environment friendly, safest, time saving and reliable mode of transportation.

Safe, economical and reliable operation of railways heavily depends on construction quality and subsequent maintenance. In both these processes, estimation of material responses due to train movement is a considerable challenge. This challenge was first countered using some rules of thumb and empirical relationships. These empirical relationships were then improved in the light of theory, field and laboratory testing. Many full-scale field and laboratory studies conducted over the course of time to study the railroad trackbed performance in more realistic manner. Broad categories of solutions used in the railroad analysis and design are empirical (or semi-empirical), analytical and numerical solution. Each of these solutions have their own merits and demerits.

Empirical solutions are usually site specific therefore have limited widespread applications. Analytical solutions are relatively easy and reliable to apply but these solutions cannot accommodate geometric changes. Numerical solutions generally have a widespread application in almost all fields, mainly due to their ability to characterize different materials, flexibility to generate different geometric shapes, and producing results with acceptable accuracy.

In addition, it allows to study the responses at any given location within the model which is sometimes not practically possible in the field or laboratory studies. As the railroad trackbed consists of widely different materials including steel, wood, concrete, and soils, and experiences complex interactions at their interfaces, numerical solutions are generally preferred over other methods.

1.2. Literature Review

1.2.1. Structural Modeling of Railroad Trackbeds

Many researchers have proposed different models to predict the stresses and strains (or displacements) in different components of the railroad trackbed structure. The Beam on Elastic Foundation (BOEF) theory provided the earliest theoretical solution framework [1]–[5] for analysis and design of pavements. Winkler [6] used the Euler-Bernoulli beam supported by elastic foundation. He assumed the reaction forces are function of beam deflection at any given point along the beam under the application of externally applied loads. Burmister then introduced multilayer elastic theory [7], which facilitated many researchers to model the substructure with different materials.

The BOEF theory upgraded by incorporating the multilayer elastic theory [7], brought the earliest railroad numerical models including MULTA [8]. MULTA model used Burmister's multilayers elastic theory in conjunction with structural analysis models to solve a three-dimensional model for tie-ballast reactions [9]. Some of its limitations included the inability to allow relative displacement between tie and ballast and all forces were in vertical direction only ignoring shear forces.

Another breakthrough was the introduction of Finite Element (FE) methods [10]–[14] and its applications for pavement designs. Chang et. al. built a model, named PSA, on the fundamentals

of FE methods with prismatic element types for substructure [15]–[18]. It also separated the substructure from superstructure for the response calculations while maintaining the continuity conditions, baseline of FE analysis. This offered some advantage over MULTA including allowing to change the material properties along the tie and across the rail beam and computational economical when compared with models using brick elements.

In addition, ILLITRACK [19], [20] combined the two-dimensional analysis and longitudinal direction followed by traverse direction two-dimensional analysis, thus formulating a quasi-three-dimensional finite element analysis. The model attempted to analyze the non-linearity and stress dependent response of materials to simulate the physical problem more accurately. The resilient modulus (Er), a ratio of cyclic stress to the corresponding recoverable strain, as given by Equation (1.1), was used to model the nonlinearity of ballast and subballast [21], [22]:

$$E_r = K_1 \theta^{K2} \tag{1.1}$$

where, K1 and K2 are soil parameters obtained from the laboratory testing. The major drawback of the model was its pseudo-three-dimensional assumption.

Later, GEOTRACK was proposed by Chang et al.[23], which was a multilayer theory based three-dimensional model which was recently upgraded GEOTRACK for railroad track analysis [24] with Graphical User Interface (GUI) features. GEOTRACK was built on the fundamentals of multilayer theory with quasi-dynamic loading conditions [25], [26]. It also considered the nonlinear and stress dependent behavior of materials and kept ties separated from the substructure. The model's primary focus was on the geotechnical response of the trackbed. The model considered eleven ties with wheel load applied at the mid-tie, assuming complete distribution of applied stresses by the fifth tie [27]. Rail and ties were modelled as linear elastic

beams, whereas substructure was modelled as linear elastic layers. This model was developed based on PSA and MULTA code while introducing some improvements.

Huang et al. introduced KENTRACK based on same multilayer theory and FEA to calculate stresses and strains in substructure. KENTRACK is finite element based multi-layered elastic model developed in the University of Kentucky [28]–[34]. The model was capable of prediction not only the response of the trackbed but also the cumulative damage caused by the cyclic loadings of the train operations. This model was also capable of analyzing three different types of trackbeds: a) all granular layers trackbed (typical), b) asphalt layered trackbed (replacing subballast with asphalt), and c) combined (subballast plus asphalt) layered trackbed. The failure criteria for the design procedure was cumulative vertical stresses at the top of subgrade or tensile stain at the bottom of asphalt layer, whichever occurred first. The material properties were considered as stress dependent nonlinear as presented by Equation (1.1). The bottom most layer was assumed to be incompressible to simulate the bedrock conditions.

3D20N is another three-dimensional linear elastic FEA based model [35]. This model also considered the full geometry of the railways track. This model used 20-noded isoparametric hexahedral (aka brick) elements for substructure layered materials, whereas rail and ties were modelled as 1-D beam elements. All the interfaces were modelled as zero thickness 16-noded surface elements to allow relative movement between different materials and surfaces. The model used only one fourth of the model due symmetric loading conditions and geometry and spans over five ties only. The boundary conditions were set such that the surfaces along axis of symmetries and the bottom most surface were constraint for normal movements while allowing movement in other two directions.

Huang [36] compared several models for their advantages, disadvantages and their predictions. Of all these models, GEOTRACK and KENTRACK are the most common among researchers and practitioners. Both these programs have their merits and demerits. However, some of their limitations based on Huang's study [36] are as below:

1) While using GEOTRACK, it is likely to miss the maximum stresses due the fact that loads are applied directly above the supports;
2) GEOTRACK assumes rail as a beam of finite length and do not consider the jointing effects;
3) Neither of these models account for time dependent response of materials;
4) None of these models consider dynamic effects of rail operations;
5) Use of linear elastic models to similar soil behaviour can cause considerable errors;
6) The effects of lateral forces are neglected altogether.

The variability of different materials has gain momentum over the past four to five decades. And many studies have reported and compiled the statistical parameters of a variety of material properties. The use of probabilistic analysis in civil engineering is very limited primarily due to its complexity, high computational demand (significant number of iterations), and lake of amount of data available (e.g., number of tension tests on steel for Young's modulus) to perform reliable statistical analysis. Many attempts have been made to propose simplified versions of different methods but at the cost of accuracy of the results. Much work has been done to apply probabilistic methods (Monte Carlo) on the design of retaining wall and slope stability problems, which can be solved analytically in few steps.

1.2.2. Model Evaluation and Validation

To evaluate and validate the performance of a new structural railroad track model, there are many ways which can be used. Generally, these methods can be categorized into three major groups: a) analytical solutions, b) numerical solutions (commercial software's), and c) other well-known similar models and/or full scale filed test measurements.

Keeping in view the significance of the evaluation of any newly developed model, many researchers have proposed some standard set of problems and their analytical solutions [37]–[40]. These tests include patch test, straight cantilever beam, curved beam, twisted beam, rectangular plate, spherical shell and thick-walled cylinder. Among those standard tests, patch test and straight cantilever beam model under variety of different in-plane and out-of-plane loading conditions, are considered quite reliable among the researchers' due to their simplicity and versatility. By varying the loading directions and elements shapes, different deformation modes can be examined.

Another means of evolution to validate a newly developed model is using commercial software that has widely been accepted and used. These programs must have passed all kinds of rigorous evaluations and validations. There is a long list of such software in the market and some commonly used names are Abaqus, ANSYS, Advance Design, DEFORM, Nastran, LS-DYNA, SAP2000, AutoDesk Mechanical, RFEM, Visual FEA etc. among these programs, Abaqus and ANSYS are the most common programs, especially among the researchers working with FE models.

1.2.3. Soil Variability

There are various methods available in the literature to compute the intrinsic variability of the materials using standard deviation, depending upon the amount of data available. Most commonly used four methods are briefly explained below.

1.2.3.1. Direct Estimation Method

The direct estimation method uses the basic definition of the standard deviation to estimate the soil property. This method is applicable when there is enough and reliable data available. When there is not enough data available to estimate standard deviation, or correlations, or judgements are used to determine the values of the variables, then it is convenient to look for published literature for estimates of coefficient of variance (COV). It is common to report variables' COV instead of standard deviation, as it is normalized by the mean, thus making it a dimensionless quantity. Duncan [41] has compiled a list of COV values suggested by many researchers for different soil properties and presented in

Table 1.1. Values of coefficient of variance (COV) for various soil properties and in-situ tests (Source: [41])

Property or in situ test result	Coefficient of Variation (COV)	Source
Unit weight (γ)	3–7%	Harr (1984), Kulhawy (1992)
Buoyant unit weight (γ_b)	0–10%	Lacasse and Nadim (1997), Duncan (2000)
Effective stress friction angle (ϕ')	2–13%	Harr (1984), Kulhawy (1992)
Undrained shear strength (S_u)	13–40%	Harr (1984), Kulhawy (1992), Lacasse and Nadim (1997), Duncan (2000)
Undrained strength ratio (Su /σ')	5–15%	Lacasse and Nadim (1997), Duncan (2000)
Compression index (Cc)	10–37%	Harr (1984), Kulhawy (1992), Duncan (2000)
Preconsolidation pressure (P_p)	10–35%	Harr (1984), Lacasse and Nadim (1997), Duncan (2000)
Coefficient of permeability of saturated clay (k)	68–90%	Harr (1984), Duncan (2000)
Coefficient of permeability of partly saturated clay (k)	130–240%	Harr (1984), Benson et al. (1999)
Coefficient of consolidation (c_v)	33–68%	Duncan (2000)
Standard penetration test blow count (N)	15–45%	Harr (1984), Kulhawy (1992)
Electric cone penetration test (q_c)	5–15%	Kulhawy (1992)
Mechanical cone penetration test (q_c)	15–37%	Harr (1984), Kulhawy (1992)
Dilatometer test tip resistance (q_{DMT})	5–15%	Kulhawy (1992)
Vane shear test undrained strength (S_v)	10–20%	Kulhawy (1992)

1.2.3.2. ***Three Sigma Rules***

When there is not enough data available that direct estimation method can be applied, then the Three Sigma Rule is quite helpful. This rule was first introduced by Dai and Wang [42], who used a fact from probabilistic theory that 99.72% of all values of a normally distrusted parameter fall within plus or minus three standard deviations (three sigma) from the average value. They

proposed that standard deviation can be estimated by dividing the difference of potentially extreme high and low values by six Equation (1.2) as below:

$$\sigma = \frac{(HCV - LCV)}{6} \tag{1.2}$$

where HCV and LOCV are highest and lowest conceivable values, respectively. Accuracy of this estimation will certainly depend on the accuracy of HCV and LCV values.

1.2.3.3. N-Sigma Rules

Estimation of HCV and LCV is not an easy task, as it first appears [43], as the ranges of these values cover leave only 0.27% possibility that a value can fall outside of this range. Keeping in mind of this difficulty, Foye et al., [44]suggested a modified version of the Three Sigma Rule and named it "N-Sigma Rule" as shown in Equation (1.3) below.

$$\sigma = \frac{(HCV - LCV)}{N\sigma} \tag{1.3}$$

where N_σ is the number smaller than 6 that reflects the fact that estimates of LCV and HCV cannot be expected to span $\pm\sigma$. There is not any specific value proposed for N_σ, however, $N\sigma = 4$ seems appropriate for many conditions [45].

1.2.4. Random Field Generators

Various random field generator algorithms have been proposed with varying level of accuracies and ease of implementations. The most commonly used algorithms are Moving Average (MA), Discrete Fourier Transformation (DFT), Covariance Matrix Decomposition (CMD), Fast Fourier Transformation (FFT), Turning Band (TB), and Local Average Subdivision (LAS). The first three methods (MA, DFT, and CMD) are potentially more accurate but computationally expensive. On the other hand, other three methods (FFT, TB, and LAS) are more convenient to

implement but at the cost of accuracy loss. The short description of all these methods are presented below.

1.2.4.1. Moving Average (MA) method

The MA method is very accurate in terms of accuracy of results but at the cost of high computational power requirements. In this method, a random field Z(x) is constructed as a weighted average of a white noise process defined by Equation (1.4).

$$Z(x) = \int_{-\infty}^{\infty} f(\xi) dW(\xi) \tag{1.4}$$

where $dW(\xi)$ is the mean zero of the incremental white-noise process, df is the variance of the process, and f is the weighting function. The size and mesh density of this field will define the accuracy of the field generated. More information on the method is available [46], [47].

1.2.4.2. Discrete Fourier Transformation (DFT) method

The DFT method produces a continuous random filed (Z(x)) using spectral representation of homogeneous mean square, as formulated in Equation (1.5).

$$Z(x) = \int_{-\infty}^{\infty} e^{ix\cdot\omega} W(d\omega) \tag{1.5}$$

where $W(d\omega)$ is the mean zero of the interval white-noise process, and $d\omega$ is the variance of the process. During implantation, the number of integrals will be equal to the number of dimensions. The summation is evaluated at each node of the random filed mesh, which will add to the accuracy of the results but computations become very slow for a reasonable size and/or higher dimensional problems [48].

1.2.4.3. Covariance Matrix Decomposition (CMD) method

The CMD method generates a homogeneous random filed using a direct approach by defining a covariance structure, also known as spatially correlation function. Some commonly used spatially correlation functions are presented in section 4.2.2. If **B** is representing a positive definite covariance matrix, such that $B_{ij} = \rho(\tau) = \rho(x_j - x_i)$, then the required random field (**Z**) can be generated using Equation (1.6).

$$Z = LU \tag{1.6}$$

where **L** is the lower triangular matrix that satisfy $\mathbf{LL}^T = \mathbf{B}$ (generally Cholesky decomposition is used for this purpose), and **U** is a matrix of values with zero mean and unit standard deviation for a given distribution. Apparently this method, in its standard formulation, is very simple and accurate but requires a lot of computation memory, for instance for a field of 100x100x100 size in 3-D space will have a covariance matrix of size $100^2 \text{x} 100^2 \text{x} 100^2 = 10^{12}$ elements, which would be not only resource intensive and time consuming but also add to the round-off errors [49].

1.2.4.4. Fast Fourier Transformation (FFT) method

The FFT method employs the Fourier Transformation like Discrete Fourier Transformation, but by introducing some additional assumptions to make it more efficient. The most important assumption is the to assume the field as mean zero, real and discrete [50]. This helps improving the efficiency of implementing the algorithm and generating the field with relatively less computational effort and time but at the cost of compromising on accuracy.

1.2.4.5. Turning Band (TB) method

The TB method was originally introduced by Matheron [51]. This method can only be implemented in two- or higher dimensional problems, along the lines that are crossing the field domain. During the implementation, the field is generated along the lines using 1-D FFT algorithm.

The final output field depends on the number of these lines, a greater number of lines produce high quality field, as is evident in Figure 1-1.

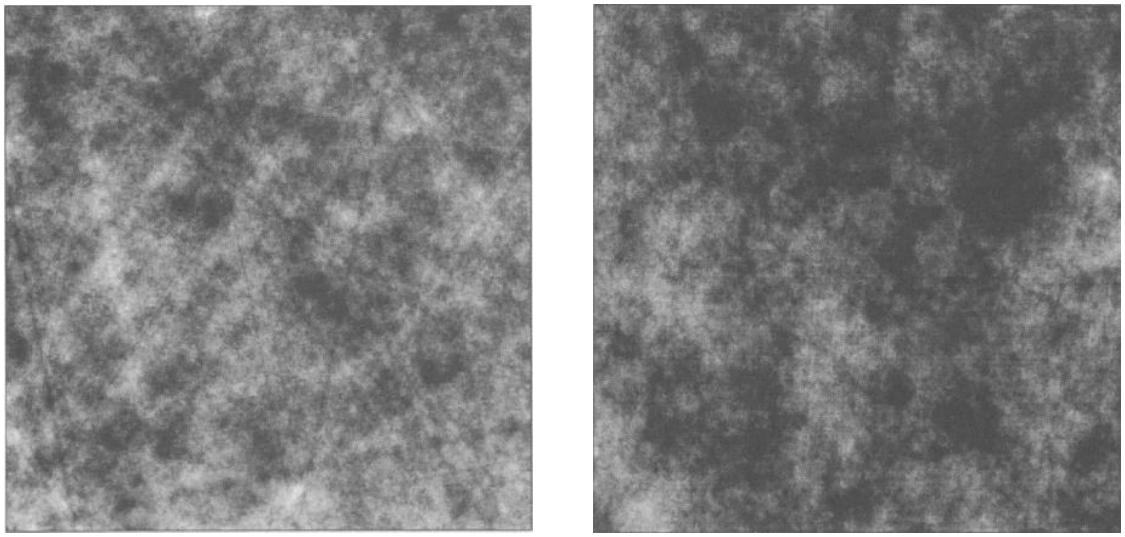

Figure 1-1. (Left) Sample function of 3D field via TBM using 16 lines. (Right) Sample function of 3D field via TBM using 64 lines. (Source: Fenton, 1994)

1.2.4.6. Local Averaging Subdivision (LAS) method

The LAS method is believed to be the most difficult to implement [49]. In this method, local averaging is implemented in a top-down repetitive fashion in multiple stages as shown in Figure 1-2.

Stage-0 is started with given average (or mean) for the field, which is then subdivided into two regions while maintaining the global (or parent) average same. This process continues until the required level of accuracy and precision. More detailed description on the method can be found at [52] and [53].

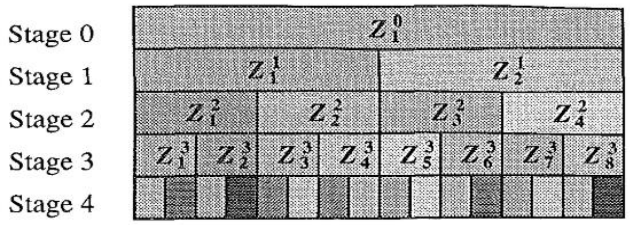

Figure 1-2. Top-Down approach to implement LAS to generate random field (Source: Fenton & Vanmarcke, 1990)

1.2.5. Risk and Reliability Analysis

1.2.5.1. Basics and Fundamentals

To perform risk and reliability analysis, there are some basic yet fundamental terminologies and definitions which are explained below in the context of civil or geotechnical applications.

i. **Variables:** Any physical, geometric or even loading condition can be a variable, for instance density of subgrade or wheel load or thickness of subgrade layer.

ii. **Correlated and Uncorrelated Variables:** Correlated variables are those which are not independent of each other, meaning changes in one variable affect the magnitudes of the other variable too. Uncorrelated variables are independent of each other.

iii. **Standard Deviation (σ):** It is a measure of scatter in the values of a variable. Mathematically, it is defined as square root of the average of squared values of the difference between each of the measured values (x_i) and the average (μ) and is expressed as Equation (1.7).

$$\sigma = \sqrt{\frac{\sum_{i=}^{n}(x_i - \mu)}{n-1}} \tag{1.7}$$

iv. **Coefficient of Variation (COV):** It is measure of the degree of scatter in the values of a variable. Mathematically, it is a ratio of standard deviation to the average Equation (1.8).

$$COV = \frac{\sigma}{\mu} \tag{1.8}$$

v. **Probability Density Function (PDF):** This is also known as Probability Distribution Function. This is a continuous distribution that indicates the probability of occurrence of any value of the variable, within the range covered by the distribution. The two most commonly used distributions are normal distribution and log-normal distribution.

vi. **Cumulative Density Function (CDF):** It is the integral of the PDF or in other words, it is the area under PDF curve.

vii. **Probability of Failure:** It is the probability that failure will occur.

viii. **Reliability:** It is the additive complement of the probability of failure. For example, if the probability of failure for a given condition is 5% then the reliability will be 95%.

ix. **Reliability Index (β):** It is the number of standard deviations between a value and its mean or most likely value. Mathematically, it can be expressed as Equation (1.9).

$$\beta = \frac{x - \mu}{\sigma} \tag{1.9}$$

1.2.5.2. Methods for Reliability Analyses

Three are numerous reliability analysis methods available in the literature but here only four most common methods will be discussed., namely Taylor Series Method, Point Estimate Method, Hasofer Lind Method, Monte Caro Simulation analysis. These methods differ primarily on their underlying assumptions and number of calculations required.

i. Taylor Series Method

The Taylor Series method is fundamentally based on first-order second moment (FOSM) analysis, which means only first two moments (the mean and standard deviations) are used in this analysis. Many researchers have described this method in the literature [41], [54]–[56]. This method is widely used due to its simplicity, ease to use and lesser calculation cycles requirement. This method assumes the distribution of the output variable (usually the factor of safety, settlement or stress). This method requires only 2N+1 calculation cycles, where N is number of variables involved. In this method, COV of the output variable is first computed and then its probability of failure (or conversely reliability) is calculated using assumed distribution. The complete calculation steps are explained in the following four steps:

1) Estimate the standard deviations of the input variables involved using an appropriate method;
2) Estimate the standard deviation of the output variable using Equation (1.10),

$$\sigma_F = \sqrt{\left(\frac{\Delta F_1}{2}\right)^2 + \left(\frac{\Delta F_2}{2}\right)^2 + \ldots\ldots\ldots + \left(\frac{\Delta F_n}{2}\right)^2} \tag{1.10}$$

$$V_F = \frac{\sigma_F}{F_\mu} \tag{1.11}$$

where $\Delta F_1 = (F_1^+ - F_1^-)$. F_1^+ and F_1^- are the outputs calculated using the value of the first parameter (variable) increased and decreased by one standard deviations, respectively. The subscripts 1, 2, … n represent that there are "n" number of variables involved in the analysis;

3) Calculate the mean most likely value of the output (F_μ) variables by performing deterministic analysis, i.e., using the mean values of all the variables;

4) With both mean and standard deviation of the output know, and assumed distribution, probability of failure can be estimated using either of the Equation (2.1) or (2.2).

ii. Point Estimate Method (PEM)

The PEM method is also a FOSM reliability method and was first introduced by Rosenblueth [57] whereas its application in geotechnical engineering was presented by Baecher and Christian [54], Harr [56] and Wolff [58]. This method is an extension of Taylor Series Method in which output variable is calculated using input variables such that they are either means plus one standard deviation or mean minus one standard deviation. By employing all possible combinations for N variables, there will be 2^N calculations cycles. This method also assumes the distribution of the output variable. Step by step procedure to apply this method is presented below:

1) Estimate the standard deviations of the input variables involved, using an appropriate method;
2) Calculate both values mean plus one standard deviation and mean minus one standard deviation for all the variables. Then perform the analysis for 2^N cases satisfying all the possible combinations of these values;
3) Assign the probability weight to each case. For uncorrelated variables, all cases will carry the same weight, i.e., 1/N;
4) Calculate the mean or most likely value of the output (F_μ) variables by performing deterministic analysis, i.e., using the mean values of all the variables;
5) Calculate the standard deviation of the output variable using the basic definition Equation (1.2). The mean output values are calculated in step-4, whereas individual output values are calculated in step-2;

6) Probability of failure can be computed using Equation (2.1) or (2.2) depending on the assumed distribution for the output variable.

iii. Hasofer Lind Method

The Hasofer Lind method was first introduced by Hasofer Lind [59] but this method could not get much attention due to its complication and more number of cycle requirements. On the other hand, however, Filz and Vavin [60] found this method more accurate than Taylor Series and Point Estimate Methods. This method is relatively more accurate since it does not assume the distribution of the output variable, which is hard to predict and thus subject to much variations. However, it assumes the distribution of the inputting variable, which is relatively easier to predict and assume with reasonable accuracy. A brief description of the step by step procedure of this method is presented below:

1) Estimate the standard deviations of the input variables involved;
2) Assume the distribution of each input variable and indicate their category, load or resistance. Variables categorized as load will use negative value of standard deviations in the analysis;
3) Stage one of this method use an initial trial value of the reliability index (β) to calculate the input variables using Equation (1.12) (for normally distributed variables). Use these values to calculate the output variable. If the output variable comes out to be greater than one, then a higher value of the b should be used next trial and vice versa. Repeat this process until factor of safety becomes equal to 1,

$$x_i = -\beta * \sigma_{xi} + \mu_{xi}\,; \tag{1.12}$$

4) At stage first select the final values of all the variables at the end of stage one and increase and decrees the load and resistance related variables, respectively, by 10%.

Apply this change on one variable in each category at one time. And then perform the analysis and calculate the output variable for each variation in input variables. Now calculate the change (gradient=dg/dx_i) in output variable for a given change in input variable using Equation (1.13) and (1.14),

$$dg/dx_i = \Delta F/\Delta x_i' \tag{1.13}$$

$$x_i' = \frac{x_i - \mu_{xi}}{\sigma_{xi}}; \tag{1.14}$$

5) This stage (3rd) is like first stage where reliability index value is assumed, and values of input variables are then calculated but his time using Equation (1.15). Then calculate the factor of safety. If the factor of safety is not equal to one, then keep reliability index changing and repeating the process until factor of safety becomes equal to one,

$$x_i = -\alpha_i \beta * \sigma_{xi} + \mu_{xi} \tag{1.15}$$

$$\alpha_i = \frac{\left(dg/dx_i'\right)}{\sqrt{\sum\left(dg/dx_i'\right)^2}}; \tag{1.16}$$

6) The final value of reliability index in stage 3 will represent the final reliability index of the failure mechanics. Probability of failure can then be calculated using probability charts.

iv. Monte Carlo Method

The Monte Carlo method is quite different from other three methods presented earlier in three ways [45], including a) it require way more calculation cycles than any other method (in the range of N=5000 or even 10,000), b) it assigns a specific distribution to each input variable, and c) special program/codes are required to apply this method automatically due

to very high number of calculation cycles. This method randomly picks the values of each input variable based on the distribution assigned to it and perform the analysis. The analysis is performed for required number of iterations while keeping the track of the values of output variable. At the end of the simulations, there will be N values of output variable which can be used to determine mean, standard deviation and best fit distribution.

Accuracy of the results strongly depends on the accuracy of the input parameters of the variables and number of iterations. Reliable input parameters will certainly produce much more reliable results than that of uncertain parameters. Similarly, higher the number of iterations will produce better and consistent results. It is seen that for a given problem, probabilities of failure do not vary much after certain number of iterations. Considering the required consistency which would be needed for practical study in railroad trackbed analysis, in this study, the Monte Carlo Method is selected to perform the reliability analysis based on the new developed railroad trackbed model.

1.3. Problem Statement and Significance of This Study

Based on all the review above, it can be seen that structural analysis of railroad trackbed is one of the unique problems in civil engineering due to several factors, for instance, that it consists of a diverse group of materials, spatial extent of the structure, geological changes, environmental changes and loading conditions are just to name a few. Currently, finite element (FE) method is widely used to perform structural analysis for a wide range of problems involving continuums with acceptable level of accuracy. Many researchers have developed software that can perform deterministic analysis the railroad trackbed using FE method. However, the current challenges yet to be investigated still exist, including:

1) Most of the commercial programs assume a single representative or most likely value of the material properties, geometric dimensions or loading magnitudes. However, there exists a wide range of variations and heterogeneities in all these quantities which are completely ignored in all these programs by assuming a uniform, homogeneous and isotropic materials.
2) Limited evaluation and validation were performed for models that will be developed to perform structural analysis of railroad trackbed while using principles of random field generation and probabilistic methods.
3) Few studies on performing structural analysis using material, geometric and loading variabilities while employing risk and reliability principles to study a more realistic performance of a railroad trackbeds.

Keeping in view of the shortcoming of deterministic analysis and considerable variations in substructure materials of railroad trackbed, there is a great need to build a robust, efficient numerical model in 3-D space that can take the advantage of FE method and employ material variability through probabilistic method (random field and Monte Carlo) to simulate the model as close to real life conditions as practical with improved accuracy and judgement.

1.4. Objectives and Organization of This Dissertation

In this study, the main objective is to develop a numerical model based on RFEM method to analyze railroad trackbed using probabilistic method(s) to incorporate the material variabilities. To achieve this objective, following specific tasks have been delineated:

1) Incorporate the random field generation of the ADYTrack model using random finite element method.

2) Develop the basic numerical model of the ADYTrack based on FE method to perform deterministic analysis of railroad tracked with a user-friendly graphical user interface (GUI).
3) Evaluate and validate the newly developed (FE component) numerical model.
4) Conduct a sensitivity study to determine the quantitative response evaluation of structural performance railroad trackbed under more realistically assumptions of heterogeneous substructure layers (ballast, subballast, and subgrade).

The organization of this dissertation is such that Chapter 1 explains the introduction and literature review of railroad trackbed models, problem statement and objectives of this study; Chapter 2 describes the random finite element method of the ADYTrack and generates the random fields, Chapter 3 details the development of the basic numerical model of the ADYTrack, its salient features, mechanics involved, and graphical user interface; Chapter 4 presents evaluation and validation of the ADYTrack against analytical solutions, commercial numerical software predictions, and field test results; Chapter 5 presents the sensitivity study, conducted to evaluate the structural performance of railroad trackbed under heterogeneously modelled substructure layers under varying levels and combinations of intrinsic variations; Chapter 6 summarizes the contributions of this study, the research findings and recommends the future work.

2. RANDOM FINITE ELEMENT METHOD AND RANDOM FIELD GENERATION

Random Finite Element method (RFEM), also known as Stochastic FEM (SFEM), is an extension of standard FE method that include the uncertainties in the material properties, loading conditions, and geometric imperfections in the problems. RFEM employs the principles of probability theories and reliability analyses to model the inherent variations in the input parameters to determine the uncertainty in the response parameters. This method is generally coupled with Monte Carlo simulations to enhance the confidence in the results as well as to avoid any biasedness in the input parameters.

The usage of RFEM is gaining momentum in recent years and is being recognized as a powerful tool to challenge the very basic assumption of uniformity and homogeneity in parameters to model various materials, loading conditions and geometries. Based on its widespread acceptance, Oden et. al. [61] argue that it will be gaining more importance and interest in next decade. A state of the art review of its past, present and future is provided by [62] with some future directions. A wide range of problems in civil engineering has been published where intrinsic variations in material properties, geometric imperfections, and/or uncertainties in applied loading conditions were included [58], [63], [72]–[81], [64], [82]–[91], [65], [92]–[101], [66], [102]–[109], [67]–[71].

In this study, a finite element model based on the RFEM method and probability analysis, named ADYTrack model, is developed to include the variations of the sublayers' material properties for the performance evaluation of the railroad trackbed. A functional RFEM analysis will include three steps, including generating random fields, establishing a standard FEM, and performing Monte Carlo simulations for probability analysis. To generate random fields, first of all, a probability function needs to be selected for each random finite element; second, a spatial

correlation function is required to correlate the random finite elements together; and third, a random field generator algorithm is needed to generate the random fields.

For railroad track, the most material variation comes from the variance of the subgrade, subballast, and ballast. The steel rail, when compared to other components in a railroad track is very small. Due to the fact that this study is focused on setting up a RFM analysis prototype railroad trackbed analysis, the random field generation of various sub-layers beneath the steel rail including subgrade, subballast, and ballast are described in the following sections.

2.1. Probability Distributions

To generate a random finite element for the sub-layers beneath the steel rail for a railroad track, probability distributions need to be considered for each finite element. Normal and lognormal distributions are very common types of continuous probability distributions in many natural and social sciences. Most soil/ballast related variables follow either normal or lognormal distribution which are introduced as follows.

2.1.1. Normal Distribution

Normal distribution is also known as Gaussian or Laplace-Gaussian Distribution. It is symmetric bell-shaped curve, with its peak at the mean value while standard deviation defines its lateral spreading. Mathematically, it is expressed in Equation (2.1).

$$Normal\ p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma} \right)^2 \right] \tag{2.1}$$

where p(x) is the probability of occurrence a value of x with its mean μ and standard deviation σ.

2.1.2. Log-Normal Distribution

The lognormal distribution is another common type of continuous probability distribution. It is unsymmetrical (left skewed) bell-shaped curve, with its peak at the left of the mean value while standard deviation defines its lateral spreading. Mathematically, it is calculated using Equations (2.2) – (2.4).

$$Lognormal\ p(x) = \frac{1}{\varsigma x \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\ln x - \lambda}{\varsigma} \right)^2 \right] \tag{2.2}$$

$$\varsigma = \sqrt{\ln(1 + COV^2)} \tag{2.3}$$

$$\lambda = \ln(\mu) - \frac{1}{2}\varsigma^2 \tag{2.4}$$

where p(x) is the probability of occurrence of a value of x with its natural logarithmic mean λ and natural logarithmic standard deviation ς (zeta).

In this study, the Log-Normal distribution is selected to generate the material variations of each random finite element for the sub-layers of railroad trackbed.

2.2. Spatial Correlation Functions

Although random numbers generated using any distributions are generally uncorrelated, however, the distribution of parameter, say density of soil in the field, is not uncorrelated. In reality, although each finite element of the sub-layers for a railroad track is random, they still follow certain spatial distribution. For instance, the elements within a very small area will not vary too significant to each other. Therefore, there is a need to introduce the spatial correlation function to correlate the magnitudes of soil parameter(s) over space and/or time. These parameters are correlated over a specified distance or time, known as Spatial Correlation Length or Spatial

Correlation Time. As in this study, only spatial extent is targeted, therefore, only spatial correlation length and corresponding functions will be discussed hereafter.

The correlation length is a measure of rate of fluctuation in the input parameters for generating a random field. Larger the correlation length is, coarser (compared with FE mesh) the random field (RF) discretization mesh will be. In other words, larger correlation length will reduce the number of RF nodes, which will subsequently affect the results of random field [96], [110]–[112]. The parameter '*a'* in Equations (2.5) - (2.8) is spatial correlation length. Many correlation functions can be used to generate the correlation length of numbers of random elements, which have been introduced in the literature. However, four most commonly used analytical models are presented below.

2.2.1. Triangular Correlation Function

The simplest correlation function (ρ) is triangular function, which decreases linearly from 1 to 0 as presented in Equation (2.5):

$$\rho(\tau) = \begin{cases} 1 - \dfrac{|\tau|}{a}, & |\tau| \le a \\ 0 \quad , & |\tau| \ge a \end{cases} \tag{2.5}$$

where τ the distance between two points in random space, and *a* is the spatial correlation length.

2.2.2. Exponential (Markovian) Correlation Function

The exponential correlation function is one of the most commonly used spatial correlation functions, also known as Markovian correlation function. This is calculated using Equation (2.6):

$$\rho(\tau) = e^{-|\tau|/a} . \tag{2.6}$$

2.2.3. Second Order Correlation Function

The second order spatial correlation function is associated with second order autoregressive processes and is calculated using Equation (2.7):

$$\rho(\tau) = \left[1 + \frac{|\tau|}{a}\right] e^{-|\tau|/a} . \tag{2.7}$$

2.2.4. Gaussian (Squared Exponential) Correlation Function

The gaussian correlation function is another very commonly used spatial correlation functions, also known as Squared Exponential correlation function and can be calculated using Equation (2.8):

$$\rho(\tau) = e^{-(|\tau|/a)^2} . \tag{2.8}$$

In this study, the exponential correlation function is used to generate the spatial correlation between the random finite elements due to its ease to use and good stability.

2.3. Random Field Generation

With selected probability model and spatial correlation function, the random field can then be generated following specific generation algorithms. As shown in Chapter 1, various random field generator algorithms have been proposed with varying level of accuracies and ease of implementations. The most commonly used algorithms are Moving Average (MA), Discrete Fourier Transformation (DFT), Covariance Matrix Decomposition (CMD), Fast Fourier Transformation (FFT), Turning Band (TB), and Local Average Subdivision (LAS). The first three methods (MA, DFT, and CMD) are potentially more accurate but computationally expensive. On the other hand, other three methods (FFT, TB, and LAS) are relatively convenient to implement but at the cost of accuracy loss. A brief description of all these methods is presented in section 1.2.4.4 to 1.2.4.6.

Following a detailed comparison of complexity of implementation, accuracy, computational and time requirements, Covariance Matrix Decomposition (CMD) is incorporated in the newly developed ADYTrack model. CMD is one of the most accurate method, however, it required huge computational memory. This shortcoming was overcome by a recently proposed Stepwise CMD (SW-CMD) method [113]. The correlation matrix $\boldsymbol{R}$ in 3-D space is disassembled into three 1-D correlation matrices $\boldsymbol{R_x}$, $\boldsymbol{R_y}$, $\boldsymbol{R_z}$ by a Kronecker product using Equation (2.9). $\boldsymbol{R_x}$, $\boldsymbol{R_y}$, $\boldsymbol{R_z}$ are much smaller 1-D vectors as compared global correlation matrix $\boldsymbol{R}$. Following this disassemble, a matrix of random field, $\boldsymbol{X}$ can be calculated using same Equation (1.6).

$$R = R_x \otimes R_x \otimes R_x \tag{2.9}$$

For the same 100x100x100 field in 3-D space, the size of covariance matrix in case of SW-CMD would be (100x100x100) x 1 = 10^6 elements, as opposed to 10^{12} elements in standard CMD. This substantial reduction in matrix size requirement, higher accuracy of this method, ease of implementation, and easier handling of matrices in MATLAB are the reasons to adopt this method for ADYTrack model. The MATLAB code for the implementation of this SW-CMD is shown in Figure 2-1.

```
U = randn(nx, ny, nz);          % nx-ny-nz array
X = Lx*reshape(U, nx, ny*nz); % nx-ny*nz matrix
X = reshape(X, nx, ny, nz);    % nx-ny-nz array
X = permute(X, [2, 3, 1]);     % ny-nz-nx array
X = Ly*reshape(X, ny, nz*nx); % ny-nz*nx matrix
X = reshape(X, ny, nz, nx);    % ny-nz-nx array
X = permute(X, [2, 3, 1]);     % nz-nx-ny array
X = Lz*reshape(X, nz, nx*ny); % nz-nx*ny matrix
X = reshape(X, nz, nx, ny);    % nz-nx-ny array
X = permute(X, [2, 3, 1]);     % nx-ny-nz array
```

Figure 2-1. Implementation of stepwise CMD in MATLAB (Source: Li et.al., 2019)

Spatial correlation length and coefficient of variation play significant role in the generation of random fields. Using an example mean modulus of sub-layers of 2,000 MPa (the actual modulus of the sub-layers will change based on the actual measurements or literatures if any), the effect of spatial correlation length on the random field generation is investigated, as shown in Figure 2-2, Figure 2-3, and Figure 2-4 for spatial correlation lengths of 0.01, 1.0, and 2.5, respectively, while keeping the COVs constant at 30%. These figures have two plots, the upper plots show the variation in magnitudes along the x-direction at the middle of y-direction, as shown by the red line in lower contour maps. The lower colored contour maps show the variation in the random parameter in 2-space. These figures are for illustration purpose only. The actual random field generations in ADYTrack model to incorporate the variation in substructure layers is in 3-D space.

Figure 2-2 shows a contour map with almost no sign of correlation in space domain due to relatively smaller spatial correlation length (i.e., 0.01). However, the variation in the magnitudes of random parameter (upper plot) is quite significant due to considerable value of COV of 30%.

Some signs of correlation among random field parameter can be seen on the contour plot with relatively moderate value of spatial correlation length 1.0 in Figure 2-3. Also, the variations in the magnitudes of random parameter (upper plot) are of the same order as in Figure 2-2 since same intrinsic variations are considered in both the cases.

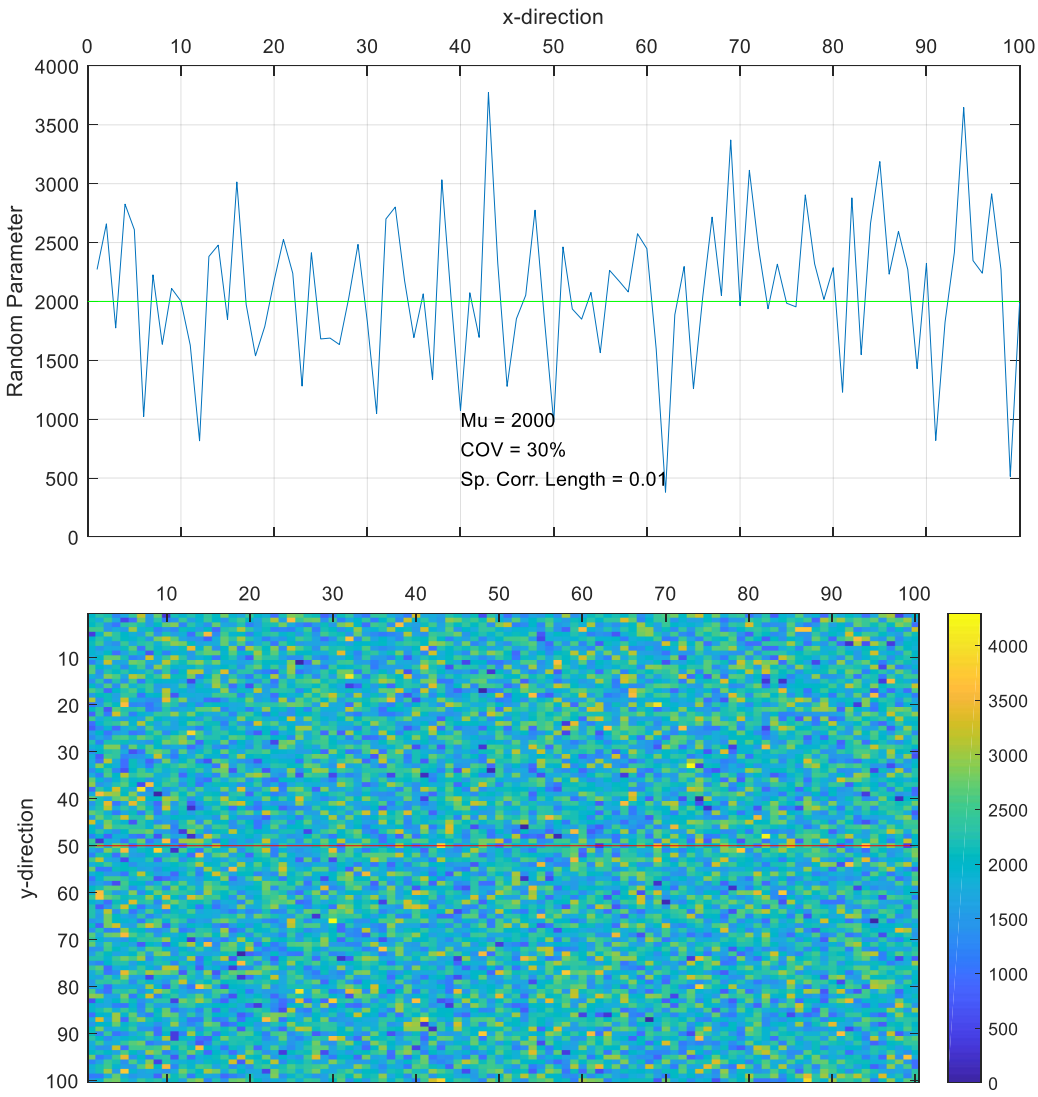

Figure 2-2. Effect of spatial correlation length *(Lc)* on the random field generation with COV = 30% and *Lc* = 0.01.

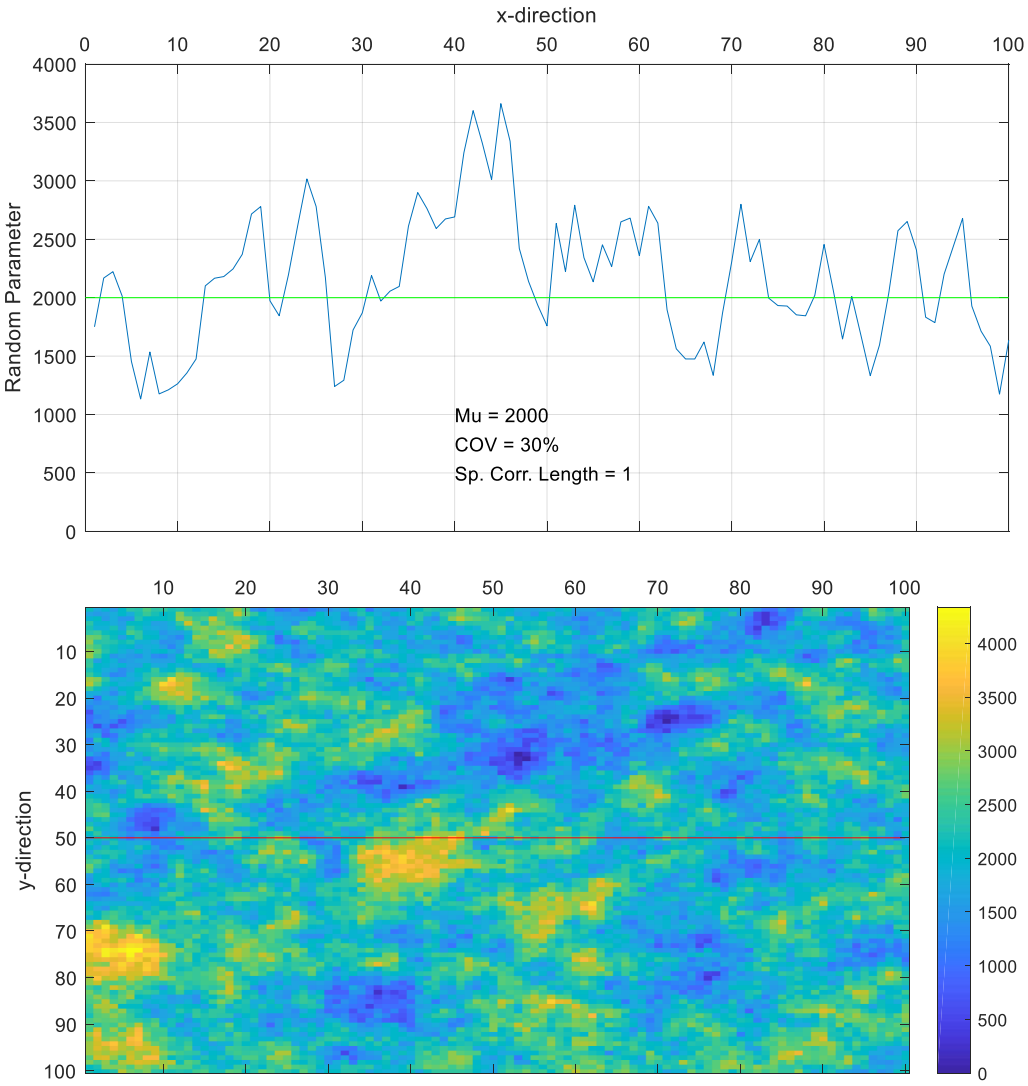

Figure 2-3. Effect of spatial correlation length *(Lc)* on the random field generation with COV = 30% and *Lc* = 1.0.

The effect of spatial correlation length is much more pronounced in Figure 2-4 with relatively higher value of 2.5. In this case, there exists a stronger correlation in space with relatively bigger pockets of similar values of random parameter, as can be seen in the contour map.

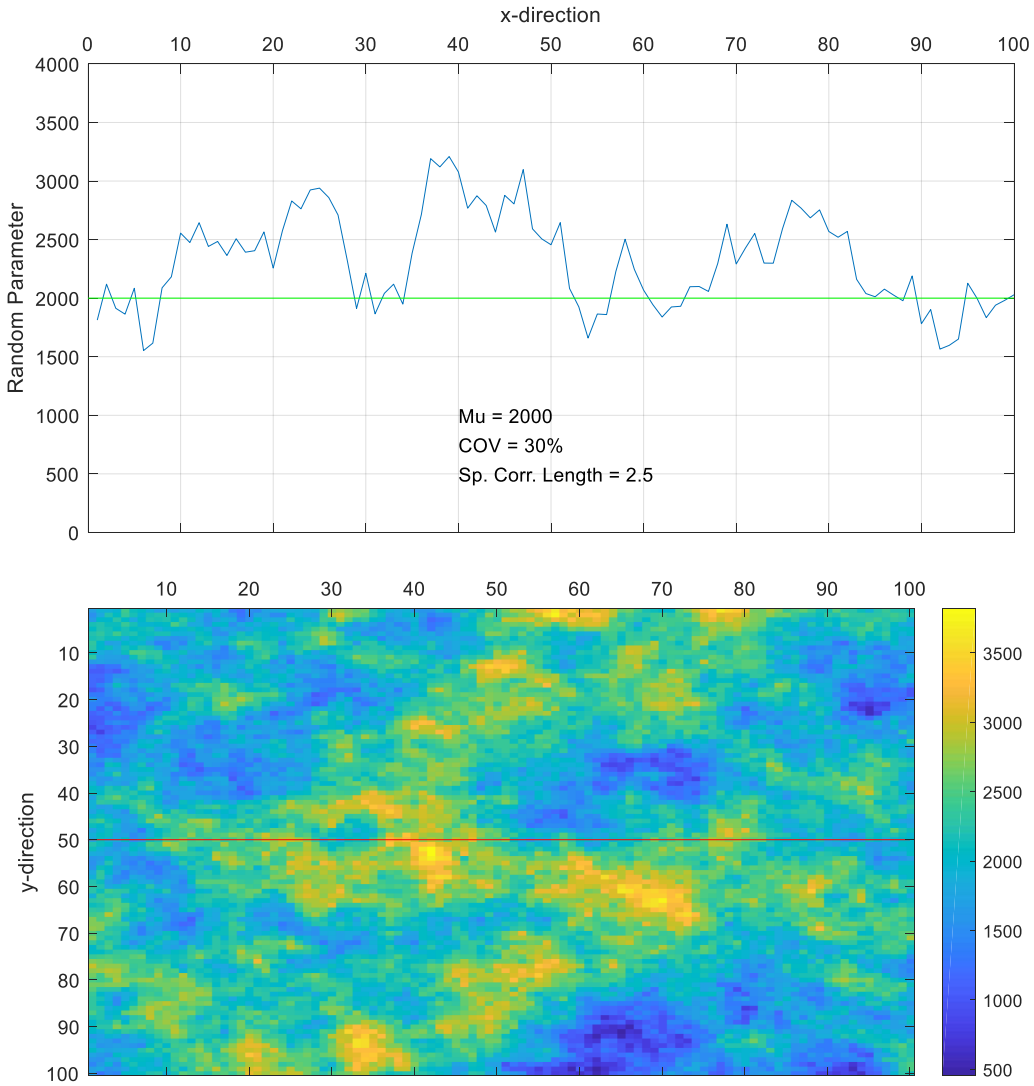

Figure 2-4. Effect of spatial correlation length *(Lc)* on the random field generation with COV = 30% and *Lc* = 2.5.

Similarly, the effect of COV on the random field generation is presented in Figure 2-5, Figure 2-6, and Figure 2-7, while keeping the spatial correlation length constant at a relatively moderate value of 1.0. A random field with almost negligible variation (COV = 1%), as can been in the line chart (upper plot) is presented in Figure 2-5. Whereas the contour map is considerably correlated in 2-D space.

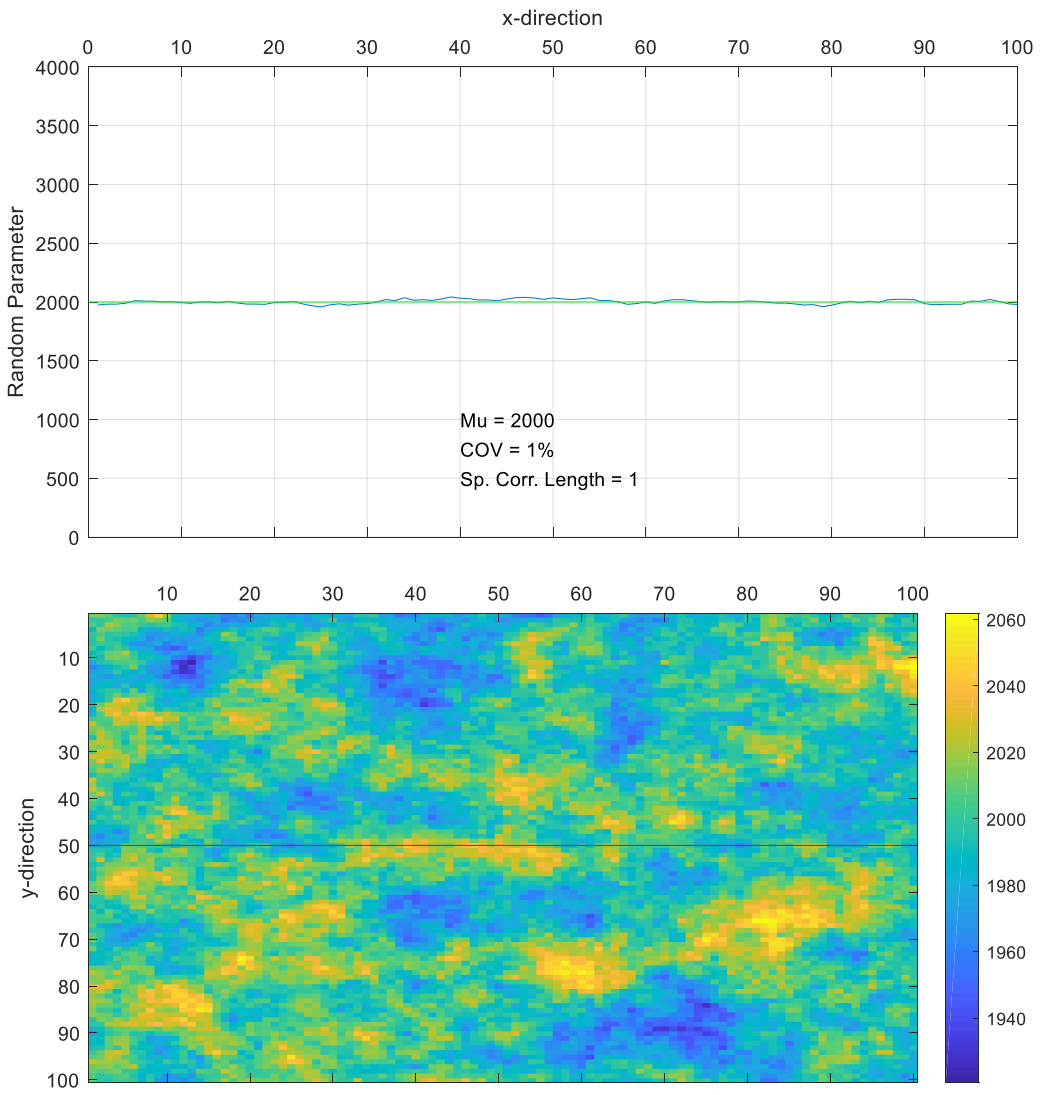

Figure 2-5. Effect of spatial correlation length *(Lc)* on the random field generation with COV = 1% and *Lc* = 1.0.

The upper plot in Figure 2-6 shows a considerable variation in random field magnitudes as compared to upper plot of Figure 2-5 due to higher COV value considered in this case. However, the contour maps in both these figures are much different, as both have same spatial correlation length of 1.0.

Figure 2-7 shows the random field generated with even higher value of COV of 50%, and therefore higher crests and deeper troughs can be seen in the upper plot. Whereas, the contour map is again similar to that of Figure 2-5 and Figure 2-6 except with stronger colors that represent the higher orders of magnitudes of random parameter.

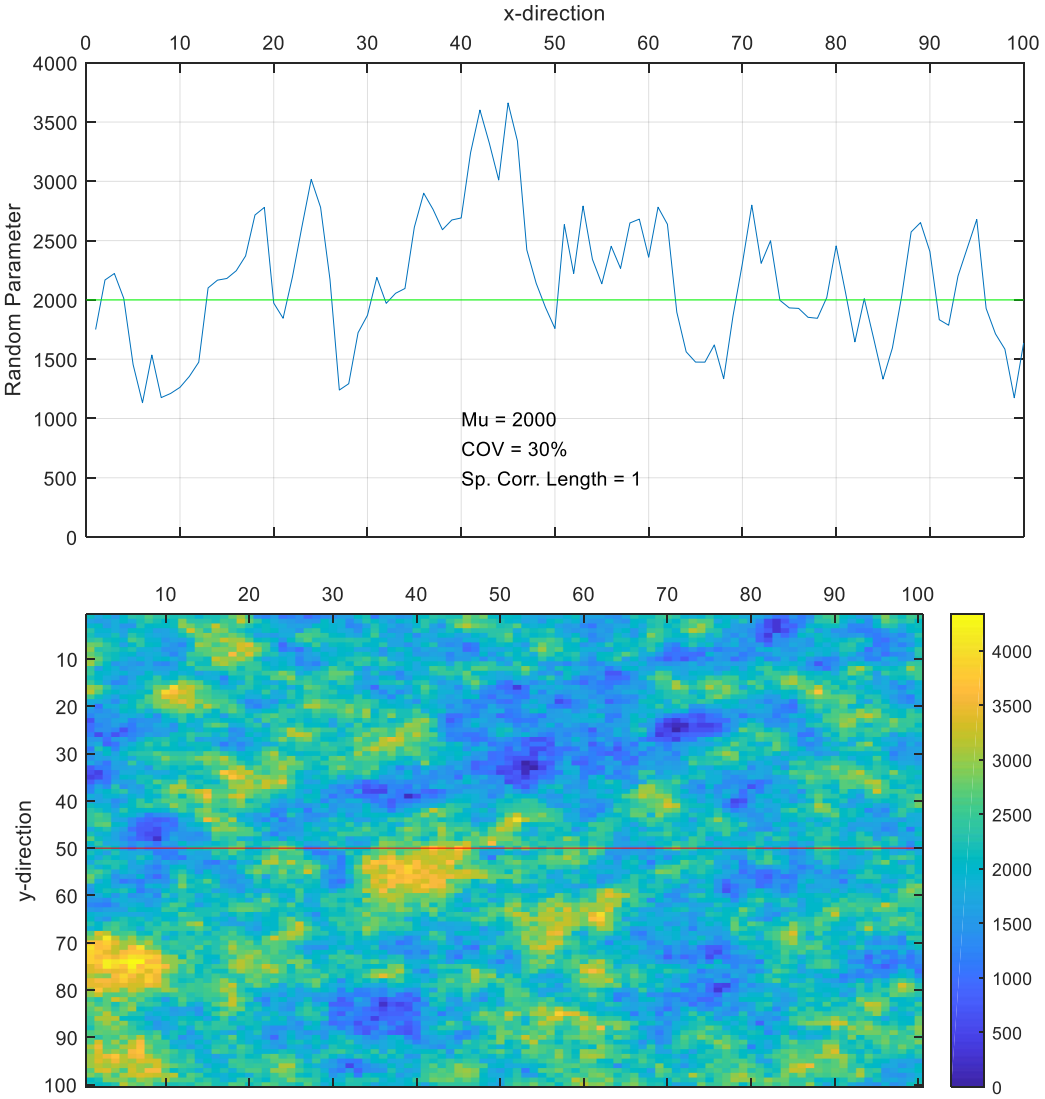

Figure 2-6. Effect of spatial correlation length *(Lc)* on the random field generation with COV = 30% and *Lc* = 1.0.

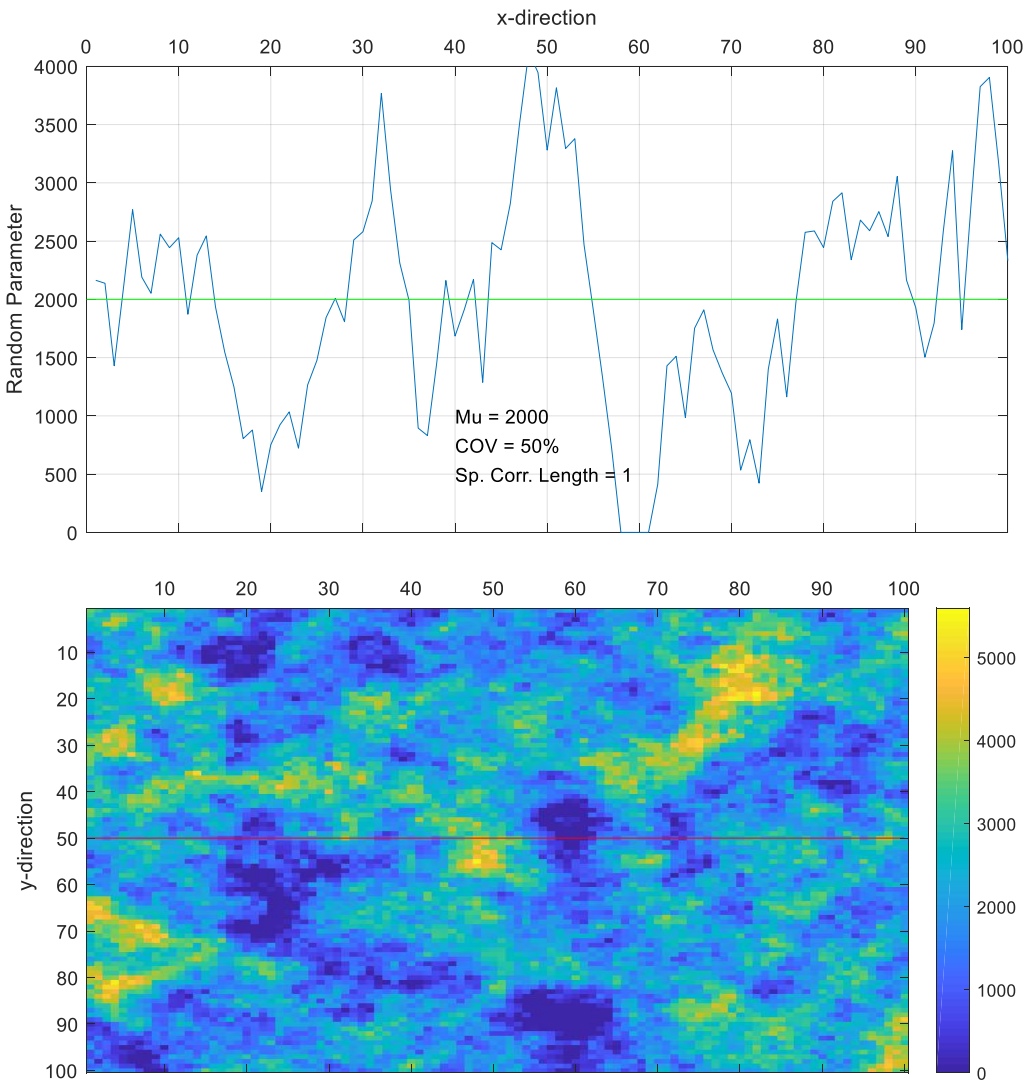

Figure 2-7. Effect of spatial correlation length *(Lc)* on the random field generation with COV = 50% and *Lc* = 1.0.

2.4. Selection of Random Field Mesh and its Overlapping with Finite Element Mesh

The size of random field mesh plays an important role, like finite element (FE) mesh, to the quality and accuracy of the random filed generated. The random field mesh size is primarily depending on the spatial correlation length parameter, whereas, the FE mesh usually depends on the geometry and response (stress or strain) concentrations. Der Kiureghian & Ke [114] suggested

that as both these meshes are depending on different criteria, so two different meshes should be defined, if possible, and this may improve the efficiency of the results.

For strongly correlated random field, the random mesh should be considerably coarser than the FE mesh, which will significantly reduce the number of random variables in the model. This will make the model simpler and less computationally demanding. Ideally, a mesh should accurately describe the geometry, gradients of the responses (stresses and strains), and intrinsic variability in the substructure layers, all at the same time.

Der Kiureghian & Ke [114] suggests that the element size of a random field should be one half to one quarter of the correlatio length parameter. They proposed this range based on repeated evalution of reliability index of their numerical experiments on a randomly distribuited rigid beam. This proposed range also confirmed by Li and Der Kiureghian [115] and Zeldin and Spanos [110] based on the their evalutions of power spectra of the random fields.

In the devloped ADYTrack model, a fraction of 1/4 of spatially correlation length is adopted to generate random field mesh, which will produce finer mesh to produce better quality random field, using the leverage of efficient Stepwise Covariance Matrix Decomposition (SW-CMD) method in 3-D space. In addition, the user is provided with an option to change the fraction in x-, y-, and z-directions dpending on the on-site conditions.

2.5. Summary

This chapter covers the introduction and fundamental components of random finite element analysis. In addition, it describes in details and highlights the advantages of the selected probability distributions, spatial correlation function and random field generators, to include in the ADYTrack model. With the successful incorporation of random finite element analysis capability, the random fields of sub-layers of railroad trackbed can be generated.

3. DEVELOPMENT OF ADYTRACK MODEL

After generating the random fields of the sublayers' material properties of railroad trackbed, a standard railroad track finite element model needs to be set up to interpolate the random fields for the sublayers, followed by the Monte Carlo simulation (MSC) for probability analysis. This chapter describes the development of the basic FEM based numerical model of the ADYTrack, which can be used to study the structural response of railroad trackbeds. The basic FEM model of the ADYTrack can generate a full geometry of the trackbed including side slopes and shoulders for each layer. It is completely based on continuum mechanics including substructure unlike many other well-known models that are based on uniform half space theory. In addition, there is no constraints in defining the extent of the model, however, longer and wider model would require more computational power. Furthermore, the model can perform analyses for special cases like track on horizontal and/or vertical curves with little modification in the program. With the development of the basic FEM model, the generation the random fields of the materials, the Monte Carlo simulation is applied to perform a valid probability analysis based on the ADYTrack. In addition, the development of its graphical user interface (GUI) is also performed. The details of underlying theories in the development of the model are presented in the following sections.

3.1. ADYTRACK Basic Finite Element Model Setup

The ADYTrack is based on a three-dimensional (3-D) model with a typical geometry (the detail code of the model and the user guideline are provided in Appendix A and Appendix B). Its cross section with all its components is shown in Figure 3-1. A user can choose any finite number of substructure layers and ties, material and cross-sectional properties of rail and tie, gauge length, tie spacing, wheel load application location, shoulder width, side slopes etc. to model a trackbed.

The model is built with half geometry while taking advantage of track center line symmetry to save computational efforts.

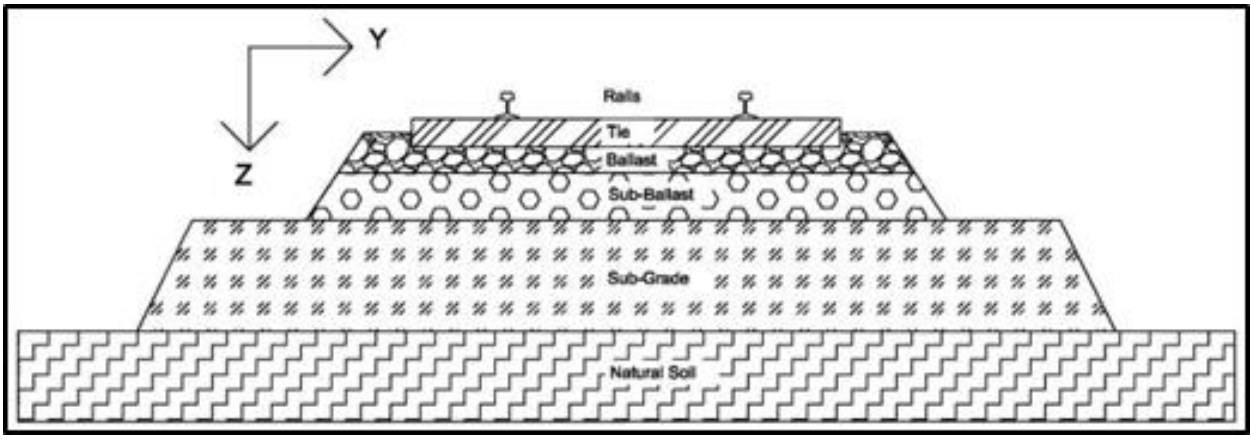

Figure 3-1. Cross section of the trackbed

The model is fundamentally based on finite element analysis, the details of which can be found in the literature [Chandrupatla and Belegundu (1997)]. The rails and ties are modelled as 2-nodal beam elements with six degrees of freedom (u_x, u_y, u_z, σ_x, σ_y, σ_z) at each node. The rail elements transmit the load to ties through a 1-D spring element with single degree of freedom, capable of withstanding both tensions and compressions.

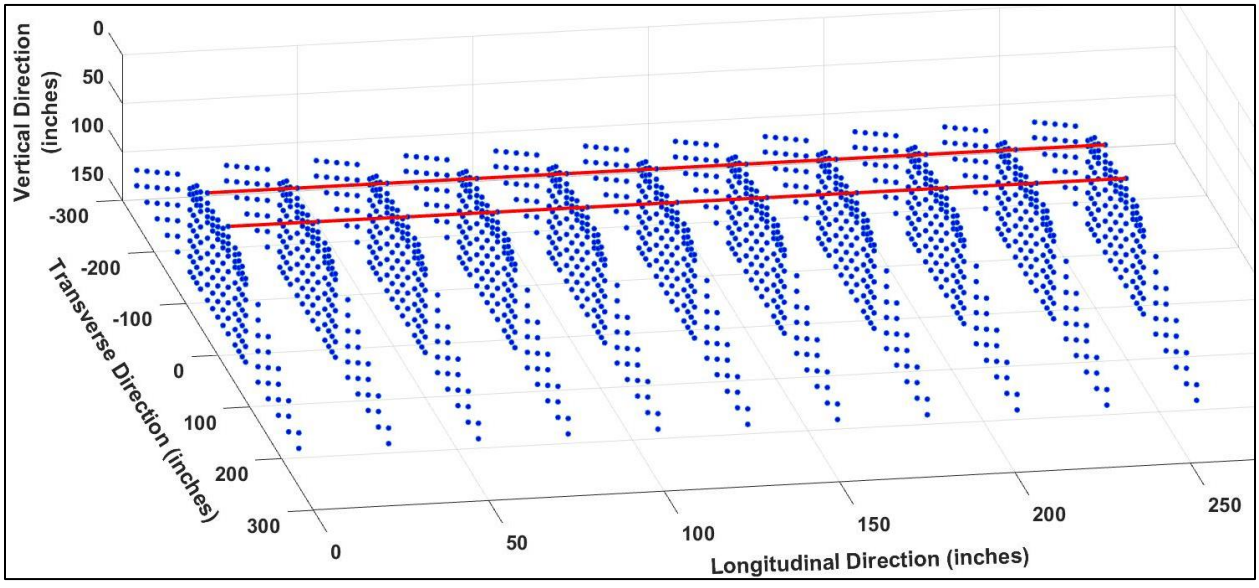

Figure 3-2. Full model considered by the ADYTrack after discretization

All the sub granular layers (ballast, subballast, subgrade and natural soil) are modelled as 8-nodal isoperimetric hexahedral (aka brick) elements. User can define as many main layers (ballast, sub-ballast etc.) as he needs to and can further uniformly divide each main layer into any number of sublayers. The thickness of each sublayer is considered as the thickness of brick elements in that corresponding sublayer. User can also choose material properties, including Young's modulus of elasticity and Poison's ratio, of these layers as linear or non-linear and elastic. Instead of assuming homogeneous half space for all these layers, a more realistic geometry is considered including shoulder width at the top and side slope for each main layer (Figure 3-1). A full scaled model considered by the ADYTrack after discretization is shown in Figure 3-2.

Node numbering plays an important role in the generation of stiffness matrix. An efficient node numbering can reduce the band width of the stiffness matrix which help reduce the computational efforts in solving the system of equations. ADYTrack creates nodes and number them such as to generate the global stiffness matrix with min bank width. First, the nodes are created along the rail then along the first tie (in y-direction at x=0) then next tie and so on. After creating nodes along rail and ties, they are then created for the complete first cross section (along y-direction at each sublayer and at x=0) and then for the next cross section and so on. Local (corner) nodes of a brick element are numbered systematically to minimize the efforts of constructing node-element connectivity table and is shown in Figure 3-3.

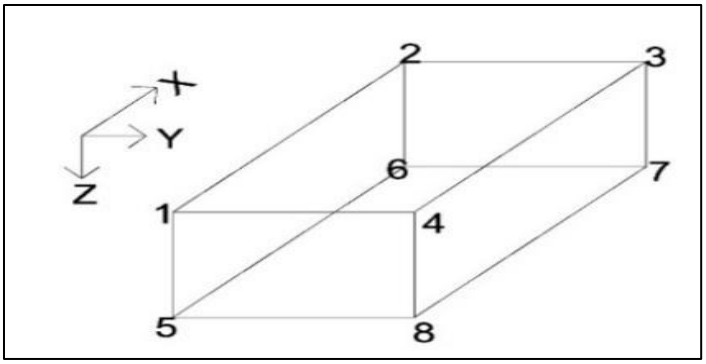

Figure 3-3. Node numbering of brick element

Like node numbering, element connectivity and their corresponding numbering is equally important to reduce the stiffness matrix band width and thus computational effort. Element connectivity followed the same route as node numbering did. Starting from rail, then ties and then cross sections in a sequential manner along each cross section.

3.2. Stiffness Matrices

Two different types of elements are used in the ADYTrack, beam and brick elements with six and three degrees of freedoms at each node respectively. Global stiffness matrix for beam element is calculated using Equation (3.1).

$$K = L^T K' L, \tag{3.1}$$

where, L is the coordinate transformation matrix from local to global coordinate system and K' is beam element stiffness matrix in local coordinates system which can be calculated using Equation (3.2) and (3.3), respectively.

$$\mathbf{L} = \begin{bmatrix} \boldsymbol{\lambda} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \boldsymbol{\lambda} & \mathbf{0} & \mathbf{0} \\ & & \boldsymbol{\lambda} & \mathbf{0} \\ & & & \boldsymbol{\lambda} \end{bmatrix} \tag{3.2}$$

where, $\boldsymbol{\lambda} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix}$; $l_1 = \frac{x_2 - x_1}{l_e}$; $m_1 = \frac{y_2 - y_1}{l_e}$; and $n_1 = \frac{z_2 - z_1}{l_e}$

$$\mathbf{K'} = \begin{bmatrix} AS & 0 & 0 & 0 & 0 & 0 & -AS & 0 & 0 & 0 & 0 & 0 \\ & a_z' & 0 & 0 & 0 & b_z' & 0 & -a_z' & 0 & 0 & 0 & b_z' \\ & & a_y' & 0 & -b_y' & 0 & 0 & 0 & -a_y' & 0 & -b_y' & 0 \\ & & & TS & 0 & 0 & 0 & 0 & 0 & -TS & 0 & 0 \\ & & & & c_y' & 0 & 0 & 0 & b_y' & 0 & d_y' & 0 \\ & & & & & c_z' & 0 & -b_z' & 0 & 0 & 0 & d_z' \\ & & & & & & AS & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & a_z' & 0 & 0 & 0 & -b_z' \\ & & & & & & & & c_y' & 0 & b_y' & 0 \\ & & & & & & & & & TS & 0 & 0 \\ & & & & & & & & & & c_y' & 0 \\ S & y & m & m & e & t & r & i & c & & & c_z' \end{bmatrix} \quad (3.3)$$

where AS=EA/le, le=length of the element, TS=GJ/le, a_z'=12EIz'/le^3, b_z'=6EIz'/le^2, c_z'=4EIz'/le, d_z'=2EIz'/le, a_y'=12EIy'/le^3, and so on. In these expressions, E, A, G, J, Iy', Iz' are Young's elastic modulus, cross sectional area, shear modulus, polar moment of inertia, second moment of area along y- and z-axis respectively.

The element stiffness matrix for brick elements is calculated using Equation (3.4) as below:

$$\mathbf{K}^{(e)} = \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \sum_{k=1}^{p_3} \omega_{ijk} \mathbf{B}_{ijk}^{\mathbf{T}} \mathbf{E} \mathbf{B}_{ijk} J_{ijk} \quad (3.4)$$

where, p_1, p_2 and p_3 denote the number of Gauss points (using a product rule of integration and same number of points in every direction, i.e., 2) in the neutral axis directions respectively, whereas w_{ijk}, B_{ijk}, E and J_{ijk} are weight of Gauss integration, Strain-Displacement matrix, Stress-Strain matrix and determinant of Jacobian matrix respectively.

Equation (3.5) presents the strain-displacement relationship which is used to calculate element strains after the whole system is solved for nodal displacements. The B matrix is calculated by taking partial derivatives of shape function with respect to global coordinates (x, y, and z). These partial derivatives are calculated using Equation (3.6).

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1^e}{\partial x} & 0 & 0 & \frac{\partial N_2^e}{\partial x} & 0 & 0 & \dots & \frac{\partial N_8^e}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1^e}{\partial y} & 0 & 0 & \frac{\partial N_2^e}{\partial y} & 0 & \dots & 0 & \frac{\partial N_8^e}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1^e}{\partial z} & 0 & 0 & \frac{\partial N_2^e}{\partial z} & \dots & 0 & 0 & \frac{\partial N_8^e}{\partial z} \\ \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} & 0 & \frac{\partial N_2^e}{\partial y} & \frac{\partial N_2^e}{\partial x} & 0 & \dots & \frac{\partial N_8^e}{\partial y} & \frac{\partial N_8^e}{\partial x} & 0 \\ 0 & \frac{\partial N_1^e}{\partial z} & \frac{\partial N_1^e}{\partial y} & 0 & \frac{\partial N_2^e}{\partial z} & \frac{\partial N_2^e}{\partial y} & \dots & 0 & \frac{\partial N_8^e}{\partial z} & \frac{\partial N_8^e}{\partial y} \\ \frac{\partial N_1^e}{\partial z} & 0 & \frac{\partial N_1^e}{\partial x} & \frac{\partial N_2^e}{\partial z} & 0 & \frac{\partial N_2^e}{\partial x} & \dots & \frac{\partial N_8^e}{\partial z} & 0 & \frac{\partial N_8^e}{\partial x} \end{bmatrix} \tag{3.5}$$

$$\begin{bmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & \frac{\partial \mu}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \mu}{\partial y} \\ \frac{\partial \xi}{\partial z} & \frac{\partial \eta}{\partial z} & \frac{\partial \mu}{\partial z} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \\ \frac{\partial N_i^e}{\partial \mu} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \\ \frac{\partial N_i^e}{\partial \mu} \end{bmatrix} \tag{3.6}$$

Jacobian matrix (J), presented in Equation (3.7) is a product of nodal values presented in Table 3.1 and partial derivatives of shape function w.r.t. to natural axis (ξ, η, μ).

$$\mathbf{J} = \begin{bmatrix} x_i \frac{\partial N_i^e}{\partial \xi} & y_i \frac{\partial N_i^e}{\partial \xi} & z_i \frac{\partial N_i^e}{\partial \xi} \\ x_i \frac{\partial N_i^e}{\partial \eta} & y_i \frac{\partial N_i^e}{\partial \eta} & z_i \frac{\partial N_i^e}{\partial \eta} \\ x_i \frac{\partial N_i^e}{\partial \mu} & y_i \frac{\partial N_i^e}{\partial \mu} & z_i \frac{\partial N_i^e}{\partial \mu} \end{bmatrix} \tag{3.7}$$

$$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \end{bmatrix} \begin{bmatrix} N_1^e \\ N_2^e \\ \dots \\ N_8^e \end{bmatrix} \tag{3.8}$$

The shape function (N) of brick elements in its indicial form is presented in Equation (3.9), whereas values of i^{th} node (corner) of the elements are presented in Table 3.1.

$$N_i^e = \frac{1}{8}(1+\xi\xi_i)(1+\eta\eta_i)(1+\mu\mu_i) \tag{3.9}$$

Table 3.1. Natural coordinates at the corners of brick elements

Node	ξ	η	μ	Node	ξ	η	μ
1	-1	-1	-1	2	+1	-1	-1
3	+1	+1	-1	4	-1	+1	-1
5	-1	-1	+1	6	+1	-1	+1
7	+1	+1	+1	8	-1	+1	+1

The stress-strain matrix (E) is a symmetric material matrix which remains constant for each element and can be calculated using Equation (3.10).

$$\mathbf{E} = \frac{E_c}{(1+\upsilon)(1-2\upsilon)} \begin{bmatrix} 1-\upsilon & \upsilon & \upsilon & 0 & 0 & 0 \\ \upsilon & 1-\upsilon & \upsilon & 0 & 0 & 0 \\ \upsilon & \upsilon & 1-\upsilon & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5-\upsilon & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5-\upsilon & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5-\upsilon \end{bmatrix} \tag{3.10}$$

where, E_c is Young's modulus (or Elastic modulus) and v is Poison's ratio. Finally, the global stiffness matrix is constructed, which will be used for solving nodal displacements using the equation, F=Kd.

3.3. Loading and Boundary Conditions

User can define the wheel load of any magnitude and can also specify the location of its application in terms of tie numbering. Boundary conditions are applied such that x-direction movement (u_x) is restricted at all cross sectional nodes of first tie (x=0) and last tie ($x=x_{max}$);

movement in y-direction (u_y) is restricted at the center line plane (XZ-plane at y=0) and the far right end of the model ($y=y_{max}$); and z-directional movement (u_z) is for the bottom most nodes ($z=z_{max}$).

3.4. Solution and Post Processing

Various built-in functions of MATLAB were tried to solve the system of linear equations including back operator (\), matrix inverse function (inv), LU factorization (lu), Cholesky factorization (chol), however the back operator (\) provided the most computational economical and accurate solution. After solving the system of linear equations for nodal displacements, stress in brick elements are calculated using Equation (3.11).

$$S^{(e)} = \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \sum_{k=1}^{p_3} \omega_{ijk} \mathbf{EB}_{ijk} d \tag{3.11}$$

where, **E**, **B**, and ***d,*** are stress-strain, strain-displacement and deformation matrices, respectively. Young's modulus of all substructure materials is modelled using Equation (3.12).

$$E_c = K_1 \theta^{K2}, \tag{3.12}$$

where K1 and K2 are material constants and $\theta = (\sigma_x+\sigma_y+\sigma_z)/3$.

In case of non-linear analysis, next step is to check the assumed Young's modulus (Ec) for each non-linear material and see if it falls within the user defined tolerance limit. In case this check fails, Ec will be recalculated and the whole systems will be solved again with modified Ec values. And this process repeat until Ec falls within the user defined tolerance limit.

After successfully passing the non-linearity check, the vertical displacements and stresses are plotted along three lines, (a) along the depth the below, (b) along the tie and (c) along the rail. User can choose any specific tie number and/or layer to see the displacement and/or stress plot.

3.5. Monte Carlo Simulations (MCS)

After setting up a standard finite element model, the random fields described in Chapter 2 can be interpolated into the standard finite element model. However, for a valid probability analysis related to random fields, large numbers of case analysis (also known as repeatability analysis which can obtain the deterministic approach multiple time, N_{sim}) are needed to be analyzed to cover presence of various random events. The MCS method is considered the most powerful and robust tool to solve complex nature of probabilistic and stochastic problems. More details on the MCS method please refer to [116][92][62][95]. In this method, equations of equilibrium are solved using deterministic approach multiple times (N_{sim}) while generating random fields of variable for specified input parameter(s) in each cycle of calculations. This generates a vector of output variables of size N_{sim}, which is then used to extract statistics of the output variables, for instance, unbiased estimate of mean, standard deviation and/or coefficient of variation (COV). Larger value of N_{sim}, e.g., N_{sim} ~500, yields good results with higher confidence level but require more computational resources [117]. One of the drawbacks of MCS is its resource intensiveness, which is covered by increased computational powers of latest computers with higher memories and processor speeds. In this study, N_{sim} =5,000 is used, which is sufficiently large size for reproducibility and to avoid any biasedness in the results.

3.6. Flow Chart

Figure 2-4 shows the flow chart of the developed ADYTrack model analysis of railroad track'e performances. The ADYTrack follows a sequential processing by first reading the user inputs which include the basic material property of the track, the COV to be analyzed, and all the geometric parameters such as tie length, size, depth of ballast and subballast, etc., then creating coordinates, nodal numbering and node-elements connectivity table. The code then applies user-

defined loading and pre-defined boundary conditions. For the first cycles of calculation, user defined Ec will be assumed, and the random field generated based on the assigned mean and COV for the sublayers and continued to solve for displacements and stress calculations. Then non-linearity check is applied to see if assumed Ec falls within tolerance limits. In case of linear analysis or passing the check, the program proceeds to results, as shown in Figure 3-4. After the completion of one cycle analysis, Monte Carlo Simulation will be performed to include 5,000 times of analysis to generate the probability analysis of the ADYTrack.

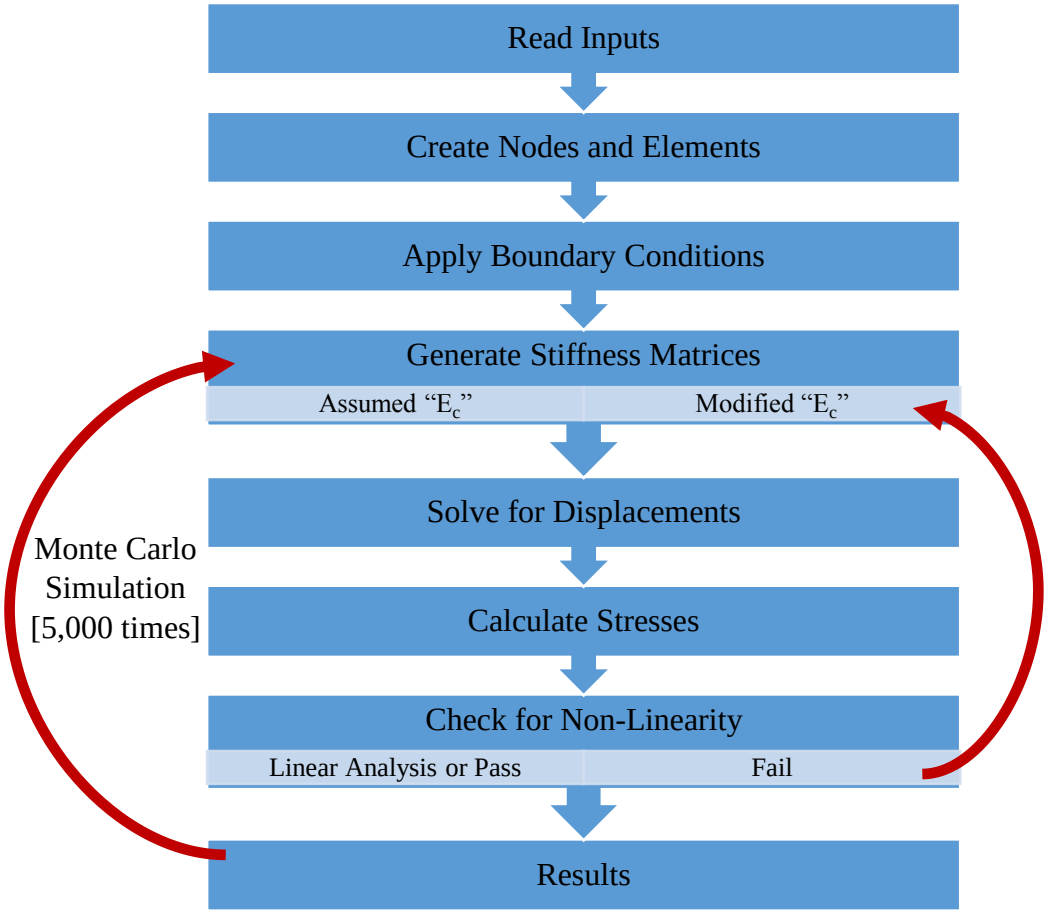

Figure 3-4. Flow chart of the ADYTrack model analysis

3.7. User Interface

In this era of technological advancement, the ease of using a model is equally important as that of its accuracy and reliability of results. In this regards, graphical user interfaces (GUI) plays an important role. GUIs act like a bridge to bypass the requirements of in-depth knowledge of technical theories or programing logics. These interfaces simply take the user input and returns the desired outputs in the form of numerical figures, graphs, charts, contour plots or even animations. Figure 3-5 shows an example of the initial version of the GUI of the model with inputs and outputs. The final inputs and outputs will be changed based on the feedback from the industry and the GUI interface will be improved to be more user friendly.

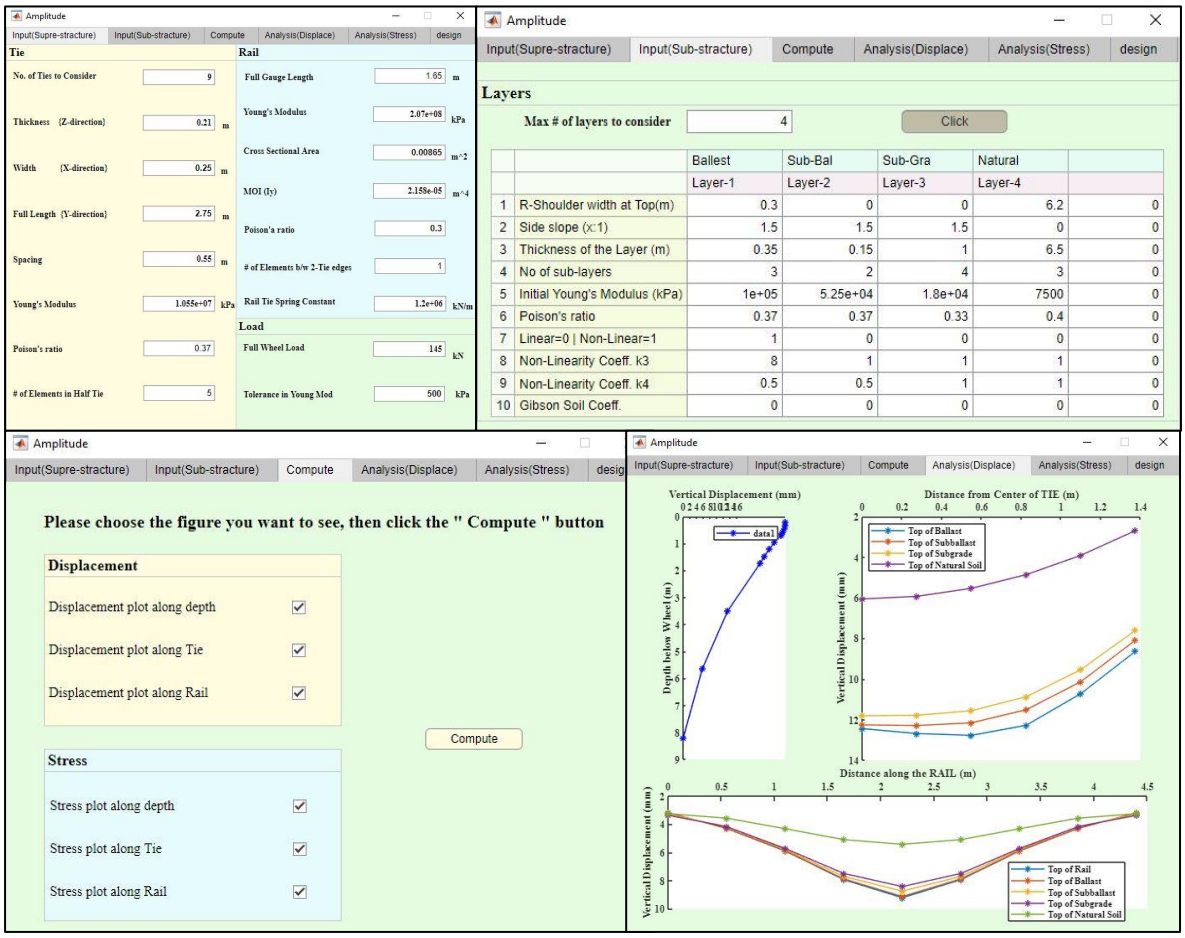

Figure 3-5. Initial version of the graphical user's interface of ADYTrack model

3.8. Summary

The development of ADYTrack is presented in this chapter in details. The basic FEM model of the ADYTrack is built such that the user can build a typical geometry of trackbed of any practical size. The model can generate quarter, half or even full trackbed geometry by taking advantage of axis of symmetry along the track centerline and along traverse direction. Rail and ties are modelled as 2 nodal beam elements with six degrees of freedom at each node, whereas granular layers are modelled as 8 nodal brick elements. Users can define the meshing density and can see the results along the depth below the wheel and along the ties and rail at the top of any sub-layer. The base code of ADYTrack and its user manual with detailed commentary are presented in Appendix-A and -B, respectively.

4. VALIDATION OF ADYTRACK MODEL COMPARED TO ANALYTICAL SOLUTIONS, COMMERCIAL SOFTWARE, AND FIELD MEASUREMENTS

Evaluation and validation of any newly developed program is an essential first step before its widely acceptance and usage. Various methods to evaluate and validate a program are presented in Chapter 1. Three different methods, namely analytical, numerical and field measurements are employed to evaluate the ADYTrack performance and validate its credibility. Straight cantilever beam model is picked for the evaluation of ADYTrack using analytical methods. Whereas academic version of ANSYS software is used to compare the numerical solutions of two models, one relatively simpler and other full scaled trackbed model. Lastly, ADYTrack predictions are compared with field measurements as well as against the prediction of other well-known models of similar nature.

4.1. Evaluation of ADYTrack against Analytical Solutions

MacNeal and Hazader [39] suggested a set of problems, the brief description of which is provided in Chapter 1, to satisfactorily evaluate a newly developed FEM based algorithm. The straight cantilever beam model is selected to test the ADYTrack performance using three different element shapes including rectangular, trapezoidal and parallelogram, as shown in Figure 4-1.

The length, width (our-of-plane) and depth (in-plane) of the beam are 6.0, 0.2 and 0.1 respectively. The material properties are selected as modulus of elasticity equals 1.0×10^7 Pa and poisons ratio equals 0.30 with standard mesh of 6×1. The beam is tested for three types of loadings, namely extension, in-plane shear and out-of-plane shear, applied at the tip of the beam.

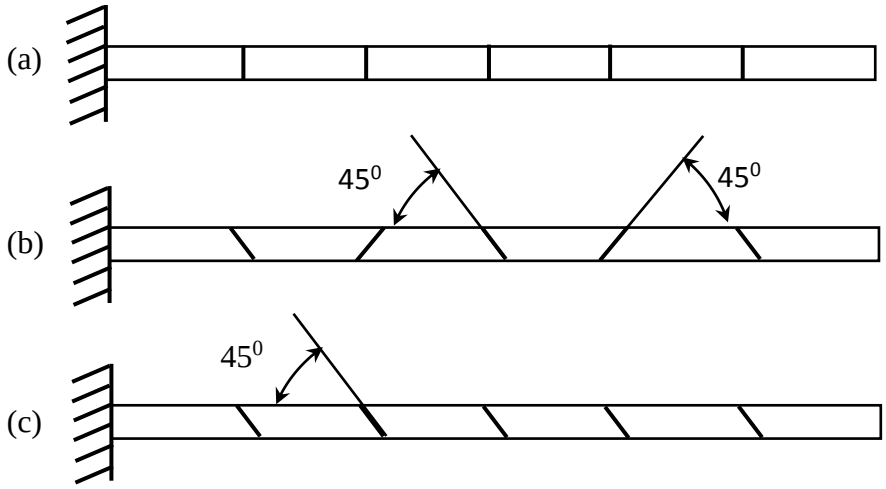

Figure 4-1. Geometry and element shapes of benchmark cantilever beam model [41]. (a) Rectangular brick elements, (b) Trapezoidal brick elements and (c) Parallelogram brick elements.

Table 4.1 summarizes the normalized tip displacements calculated by ADYTrack using hexahedral 8-noded brick elements, MSC/NASTRAN using HEXA (8) element [39] and ANSYS (APDL) using Solid185 element type. The results showed that ADYTrack model captured the response with more than 99% accuracy with all three types of elements against axial loadings. Also, both ADYTrack and HEXA unreliably predicted (with less than 20%) the tip deformations against shear loadings using non-rectangular element shapes with the standard mesh. However, ADYTrack and APDL heavily under predicted the response with rectangular element and non-extension loading conditions, i.e., 41.2% and 2.6% for in-plane shear and 10.3% and 10% for out-of-plane shear respectively. This under-prediction substantially jumped to more than 75% by slightly refining the mesh of ADYTrack model, for both cases of shear loadings, which reveals that the convergence capability of ADYTrack is relatively slower in shearing dominated environment.

Table 4.1. Normalized tip displacement in the direction of the load

Loading Condition	ADYTrack		HEXA (8)	APDL
Rectangular				
Extension	0.993	0.993	0.988	0.995
In-Plane Shear	0.760*	0.412	0.981	0.026
Out-of-Plane Shear	0.798**	0.103	0.981	0.100
Trapezoidal				
Extension	0.998	0.998	0.989	
In-Plane Shear	0.135*	0.093	0.069	
Out-of-Plane Shear	0.048**	0.014	0.051	
Parallelogram				
Extension	0.998	0.998	0.989	
In-Plane Shear	0.369*	0.206	0.080	
Out-of-Plane Shear	0.164**	0.025	0.055	

* meshing size (7 x 1 x 1)
** meshing size (10 x 1 x 1)

4.2. Evaluation of ADYTrack Against Commercial Numerical Solutions

Academic version of ANSYS (APDL) is used to gauge ADYTrack performance reliability in the category of commercial numerical solutions. Two models were developed, a simple unit cube with four different loading and boundary conditions and a full-fledged single layered railroad trackbed model.

Unit cube model was built in APDL using an 8-nodal isotropic hexahedral solid element (SOLID185) identical to ADYTrack brick element with three degrees of freedom (u_x, u_y, u_z) at each node. Young's modulus is taken as 100,000 Pa and poison's ratio as 0.3. The units of these properties do not matter for the sake of comparisons. Four set of calculations are presented in Table 4.2, where two different loading conditions (uniform and non-uniform) and two different boundary

conditions (bottom completely constraint and bottom optimally constraint) are applied to the model. A uniform load of 25 was applied in z-direction at nodes 1 to 4 for Case 1 and 2. Whereas a non-uniform loading conditions with magnitudes of 20, 25, 30 and 35 in z-direction were applied at nodes 1 to 4 respectively for Case 3 and 4. On the other hand, fully constrained conditioned, in which all bottom nodes (5-8) were restricted to move in any direction, were applied to Case 1 and 3. Whereas optimal boundary conditions, in which all bottom nodes were restricted to move in z-direction and additional restrictions included node 5 in x and y-direction and node 6 y-direction, were applied in Case 2 and 4. Contour plot of vertical stress in z-direction for loading case 4 in APDL is shown in Figure 4-2.

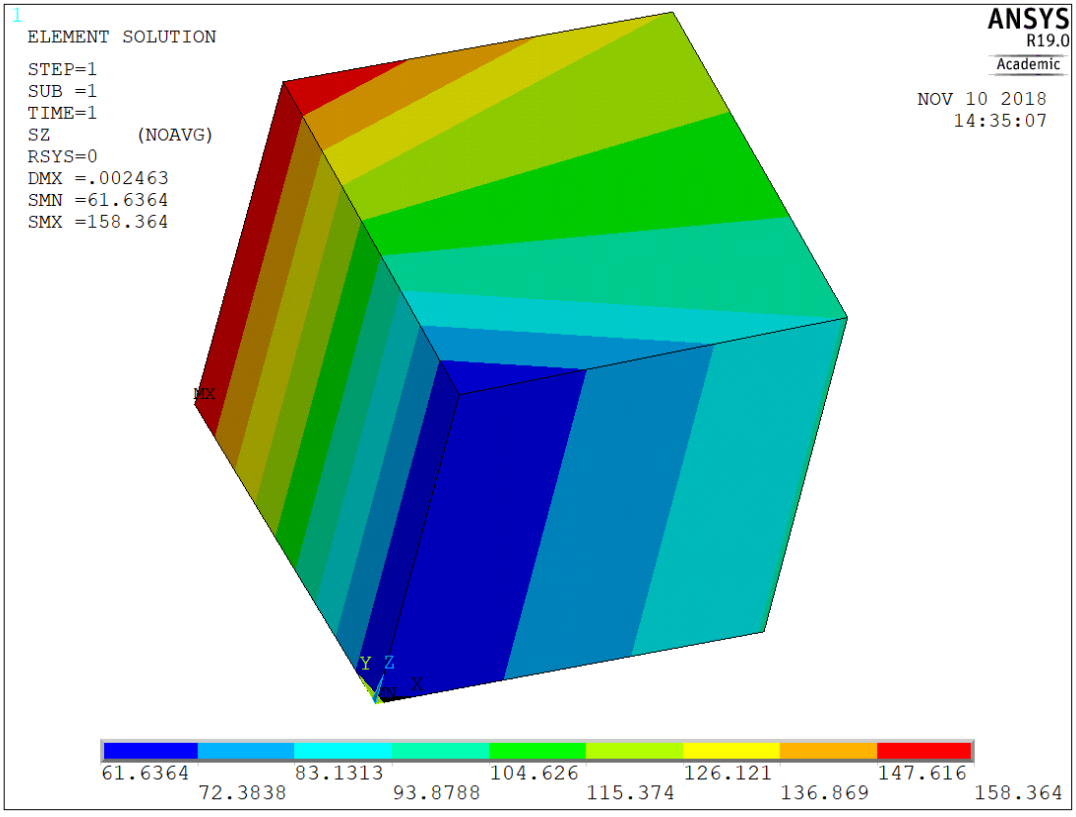

Figure 4-2. Contour plot of vertical stress in z-direction for loading case 4 (non-uniform loading and optimal boundary conditions) using APDL.

Table 4.2. Normalized tip displacement in the direction of the load

Case No.	Loading Condition	Boundary Condition	Max. Displacements (10^{-3})			Max. Stress		
			ADYTrack	APDL	% Diff.	ADY-Track	APDL	% Diff.
1	Uniform*	Fully	u_{Z4}=0.91	0.97	–6%	σ_{Z4}=113	113	0
2	Uniform*	Optimally##	u_{Z4}=1.00	1.00	0%	σ_{Z4}=100	100	0
3	Non-Uniform**	Fully#	u_{Z4}=1.59	2.04	-22%	σ_{Z4}=162	168	4%
4	Non-Uniform**	Optimally##	u_{Z4}=1.74	2.15	-19%	σ_{Z4}=147	158	–7%

* Uniform –Loading of magnitude of 25 in z-direction at node 1 to 4 ($F_{Z1} = F_{Z2} = F_{Z3} = F_{Z4} = 25$)
** Non-Uniform – Loading of magnitude of 20, 25, 30 and 35 in z-direction at node 1, 2, 3 and 4, respectively ($F_{Z1} = 20$, $F_{Z2} = 25$, $F_{Z3} = 30$, $F_{Z4} = 35$)
# Fully Constraint – Restricted movement in all three directions at nodes 5 to 8 ($u_x = u_y = u_z = 0$)
##Optimally Constraint – Restricted movements in all directions for node 5, in z-direction for nodes 6 to 8, and in y-direction for node 6. ($u_{x5} = u_{y5} = u_{z5} = u_{y6} = u_{z6} = u_{z7} = u_{z8} = 0$)

Table 4.2 also provides a direct comparison of maximum displacement and stresses produced by APDL and ADYTrack models. It is evident from the table that uniform loading conditions produced minimum percentage difference in the results, whereas non-uniform loading conditions tend to cause higher percentage differences in the results. Percentage difference of -6% was observed in maximum displacement calculations by both the models for Case 1 while keeping same different equal to 0% for maximum stress calculations. The maximum percentage differences were observed for Case 4 (Non-Uniform loading and Optimal boundary conditions), where -19% difference was observed for maximum displacements and -7% for maximum stress calculations. Case 2 (Uniform loading and Optimal boundary conditions) produced absolutely no difference in the results of both the models.

Second model in this category was a single layered yet full-fledged railroad trackbed model, developed on identical grounds in both the programs. The model was spanned on 9 ties with 0.55 m of tie spacing and 1.65 m gauge length. Main components of the model and its geometry as drawn in APDL software is shown in Figure 4-3. Rail and ties are modelled as 2-node beam elements with six degrees of freedom at each node (BEAM188). Whereas rail-tie links were modelled as spring elements, transmitting only uniaxial forces (COMBIN14) and substructure layer was modelled using 8-node isotropic hexahedral elements with three degrees of freedom at each node (SOLID185).

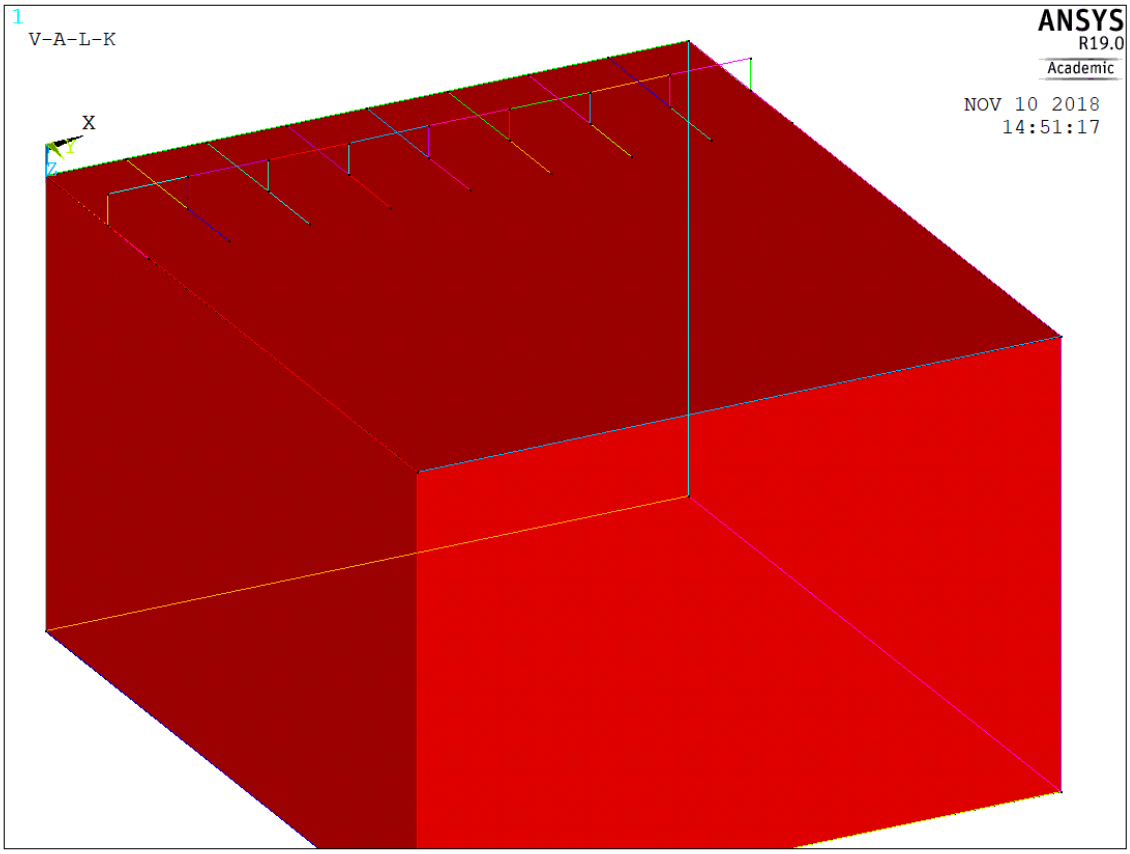

Figure 4-3. Main components and geometry of the model using APDL.

Table 4.3 summarizes other important geometric and material properties of the track used in both the models. A comparison of vertical displacements below the wheel load along the depth below the rail, as calculated by both the models is presented in Figure 4-4. It is evident from the figure that vertical displacement calculated by both the models are identical with a maximum percentage difference of 5.3%.

Table 4.3. Important track properties for evaluation of single layer model in ADYTrack

Property	**Magnitude**
Rail	
Modulus, Er	207,000 MPa
Poison's Ratio	0.30
Tie	
Modulus, Er	10,550 MPa
Poison's Ratio	0.37
Spacing	550 mm
Length	2750 mm
Width	250 mm
Thickness	210 mm
Substructure Layer	
Depth	3025 mm
Modulus, Es	480 MPa
Poison's Ratio	0.37
Rail-Tie Spring Constant	1,200 kN/mm
Wheel Load	145 kN

Figure 4-5 represents a comparison of vertical stress below the wheel load along the depth calculated by both the models. Again, it is clear from the figure that both the models calculated the vertical stress with a maximum percentage difference of 6 %. The stresses were calculated at the mid depth of each element by taking the average of all the upper and lower nodes along the depth below the wheel loading.

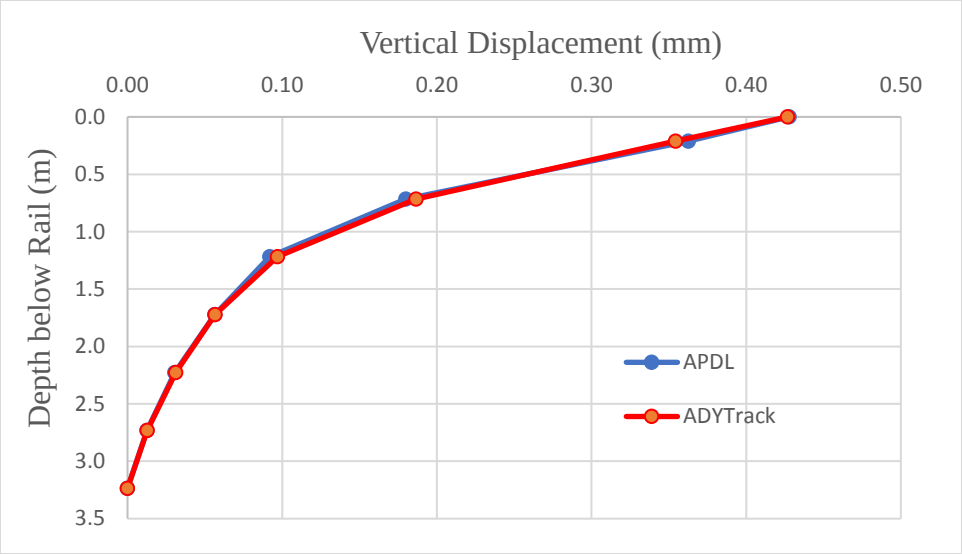

Figure 4-4. Comparison of vertical displacement below the wheel load along the depth.

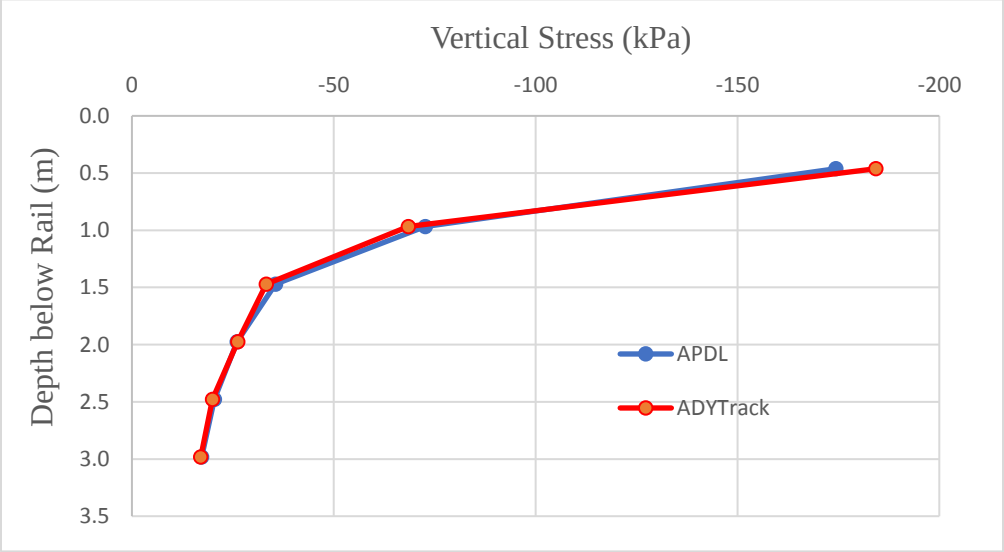

Figure 4-5. Comparison of vertical stress below the wheel load along the depth.

A contour plot of the vertical displacement at the intersection of below the rail plane and below the loaded tie plane using APDL is shown in Figure 4-6. The figure clearly helps understand the distribution of displacement in all three directions. This plot showed the displacement variations in substructure layer only and not in the rail, tie or rail-tie link elements.

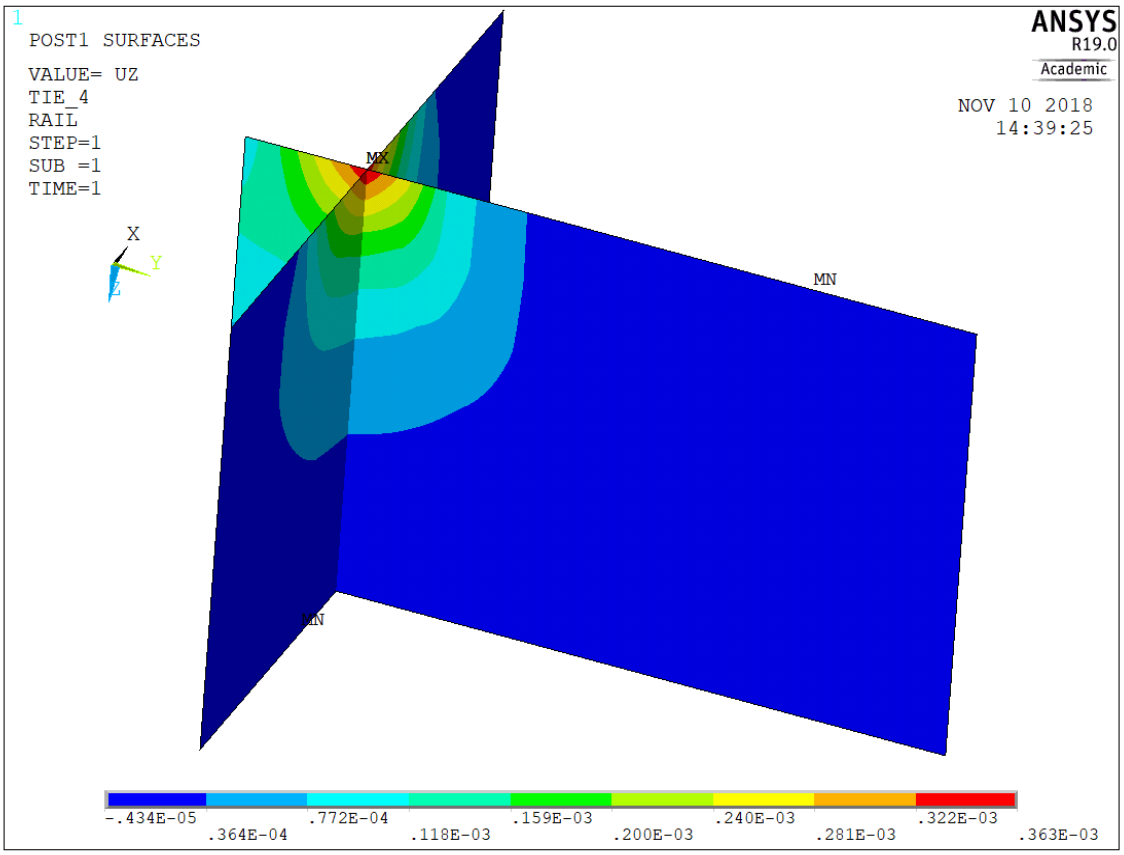

Figure 4-6. Contour plot of vertical displacement at the intersection of below the rail plane and below the loaded tie plane using APDL.

4.3. Validation of ADYTrack Against Field Measurements and Other Models Predictions

In this section, validation of the ADYTrack model is presented against two full scale field tests performed in different substructure conditions and published in the literature [23], [24], [28], [35]. Its results are further compared with the predictions of other well-known models (GEOTRACK, MULTA, 3D20N) in the railways industry. For this purpose, two separate models, named Model 1 and Model 2 were developed in ADYTrack. Summary of the material properties used in both these models are summarized in Table 4.4. Whereas the magnitudes of material properties, used in others' model calculations can be found in their respective publications.

Table 4.4. Summary of magnitudes of material properties used in the ADYTrack model

Properties	Model-1	Model-2
Rail		
Modulus, Er (MPa)	207,000	207,000
Poison's Ratio	0.3	0.3
Gauge Length (mm)	1,435	1,650
Tie		
Modulus, Er (MPa)	10,340	10,550
Poison's Ratio	0.37	0.37
Spacing (mm)	550	500
Length (mm)	2,500	2,750
Width (mm)	225	250
Thickness (mm)	175	210
Ballast		
Depth (mm)	350	350
Modulus, Es (MPa)	207	400
Poison's Ratio	0.37	0.37
Subballast		
Depth (mm)	150	150
Modulus, Esb (MPa)	138	200
Poison's Ratio	0.37	0.37
Subgrade		
Depth (mm)	2,000	1,000
Modulus, Esg (MPa)	35	90
Poison's Ratio	0.33	0.4
Rail-Tie Spring Constant		
Wheel Load (kN)	160	145

4.3.1. Case Study 1

Selig et. al [8], and Steward and Selig [118] measured stresses and deformations using an extensive instrumentation program at a test track section at the Facility for Accelerated Service Testing (FAST) Pueblo Colorado. This study helped creating models and comparing their prediction with the actual measurements. On the same lines, Shahu et. al. [35] also evaluated their model, 3D20N, against the same field measurements and predictions by Selig et. al [8] using MULTA model. Similarly, the ADYTrack model results are compared with both these model results, which include the vertical stress and vertical deformation along the depth and vertical stress along the loaded tie.

First of all, an identical geometry is created in ADYTrack as provided in Shahu et al. [35], including size of rail, gauge length; spacing, type and size of tie, substructure layer thicknesses and applied loading. However, authors analyzed this track section as a four layered substructure with one addition layer of natural soil, extending up to 10 m depth with Young's modulus of 35 MPa and poison's ratio of 0.4. Then the model was calibrated against vertical displacements along the depth as predicted by other models results (MULTA and 3D20N) using hit and trial method by varying the layer properties and mesh size. And the final values of the layer properties are summarized in Table 4.4. Then vertical stress along the depth and tie are calculated using the same mesh and material properties.

Figure 4-7 shows vertical displacements below the wheel and along the depth of the test track, predicted by MULTA, 3D20N and ADYTrack models. As this plot was used to calibrate the ADYTrack results, the predictions are simply duplicating.

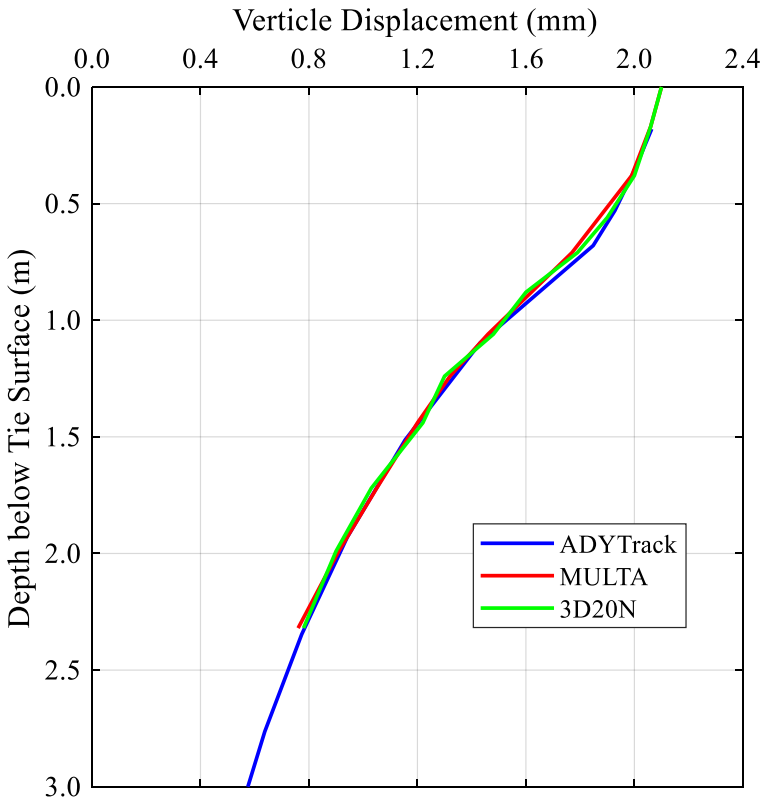

Figure 4-7. Vertical displacement below the wheel and along the depth of the trackbed.

Figure 4-8 explains the variations of vertical stresses below the wheel and along the depth of the trackbed right as predicted by all three models. ADYTrack calculations followed the typical trend of vertical stress reduction along the depth. In addition, ADYTrack overestimated the maximum vertical stress by almost 15% at the ballast surface. One of the possible reasons of this overestimation can be the 8-nodal brick elements, representing the layer structure, that make the structure relatively rigid.

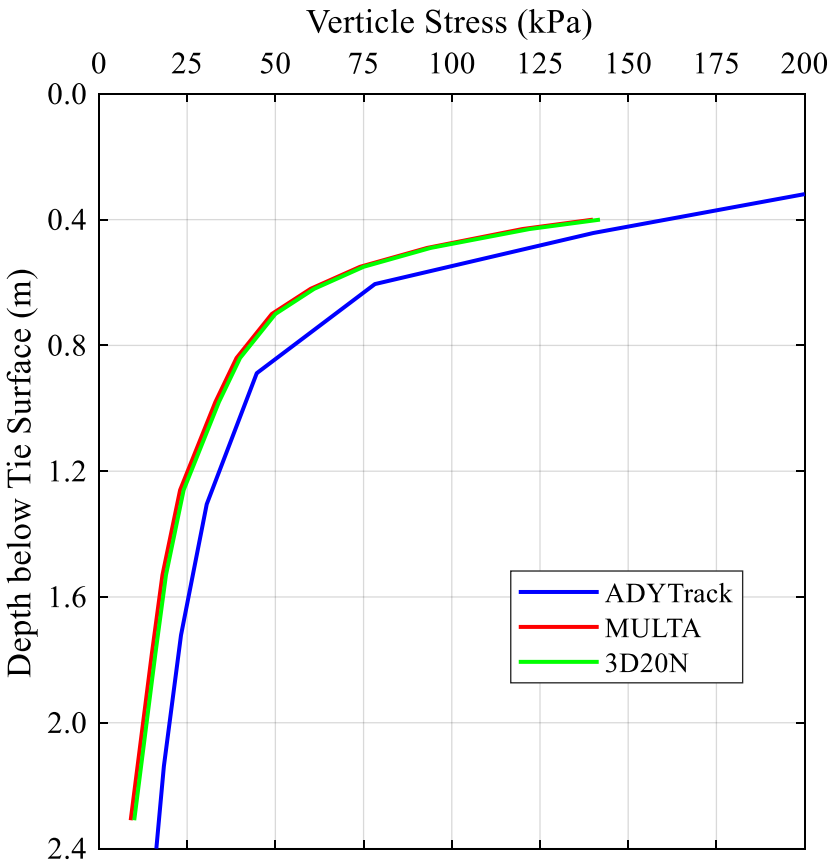

Figure 4-8. Vertical stress below the wheel and along the depth of the trackbed.

Vertical stresses calculated along the tie and below the wheel at two different depths, i.e., at the top of ballast (18 cm below the wheel load) and at the top of subgrade layer (71cm below the wheel load) are shown in Figure 4-9. ADYTrack successfully captured the overall response and distributed the vertical stress along the tie at different depths. The maximum vertical stress below the rail beam at the top of ballast layer, calculated by MULTA, 3D20N and ADYTrack are 215 MPa, 233 MPa and 230MPa respectively. The percentage difference in the estimation of the maximum stress at the ballast surface by ADYTrack was less than 10%. The figure further highlights the fact that though ADYTrack underestimated the stress at the top of ballast, whereas it slightly overestimated the stress at the top of subgrade, which was in consistent with the observation in Figure 4-8.

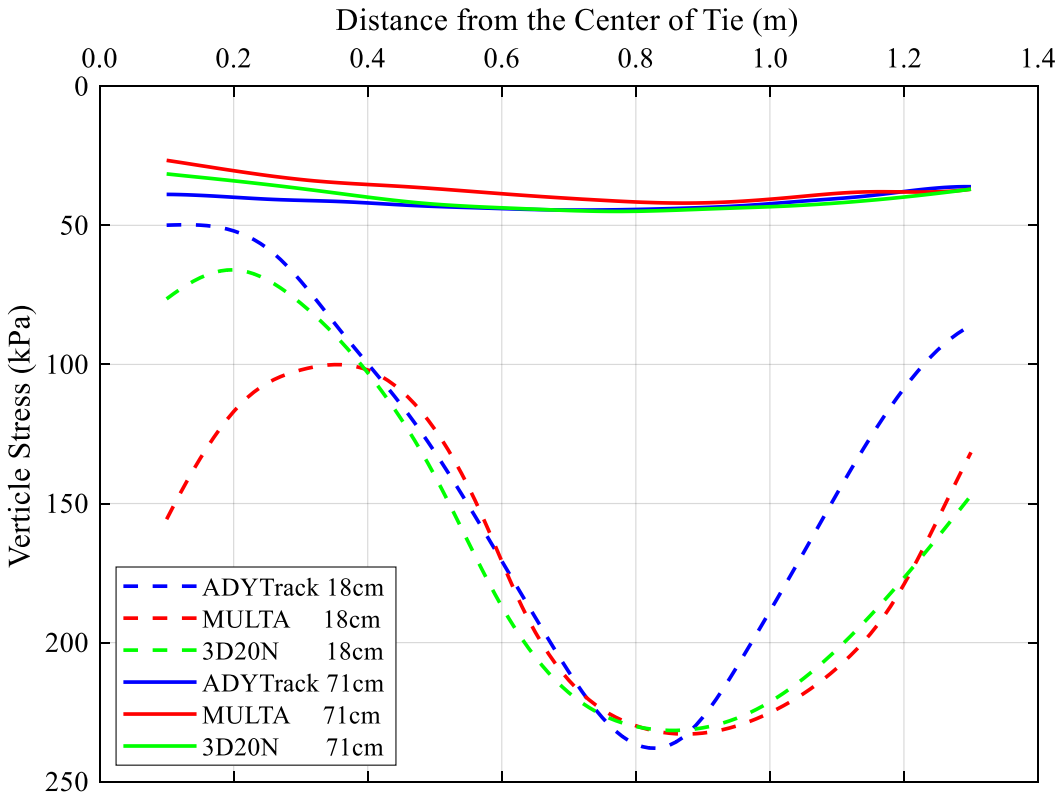

Figure 4-9. Vertical stress along the loaded tie at ballast and subgrade surfaces.

4.3.2. Case Study 2

CW Adegoke [119] studied in detail the response of ballast and subgrade in a trackbed test section at the Department of Transportation Facility for Accelerated Service Testing (FAST) through an intensive instrumentation program. Out of this comprehensive testing program, Chang et al. [23] selected only one test section results for the validation of the results of their model, GEOTRACK. The same field measurements and the predictions made by GEOTRACK model were used here for the purpose of validation of ADYTrack results. These results included the vertical displacements along the depth below the wheel load and the vertical stress distributions along the depth, ties, and rail.

Similar to Model-1, first of all, an identical railroad trackbed geometry and a double axel loading with axle spacing of 72 inch was generated in ADYTrack, as provided by Chang et al.

[23], the numerical values of which are summarized in Table 4.4. GEOTRACK assumed the last layer of infinite depth, however authors modelled the last layer, natural soil layer, with a depth of 8.5m, Young's modulus of 20 MPa and Poison's ratio of 0.4 in ADYTrack. Later, the ADYTrack model calibrated against vertical deformations along the depth below the wheel load as predicted by GEOTRACK model while changing the layer material properties and mesh size. Then the vertical stress analysis was performed (a) below the wheel load along the depth, (b) at the ballast surface along the tie under and adjacent to the wheel load, and (c) at the subgrade surface along the rail in ADYTrack.

Figure 4-10 shows the vertical displacement distribution against depth below the wheel load, as measured in the field and predicted by GEOTRACK and ADYTrack models. This plot was used to calibrate the Model-2 results in ADYTrack. It is clear from the figure that maximum vertical displacement (approximately 2.15 mm) and its variation along the depth as predictions by both the models are matching with less than 10% variations. Furthermore, both the model predictions are falling in the range of field measurements.

After successful calibration of the model, the vertical stress analysis along the depth below the wheel load was performed, the results of which are shown in Figure 4-11. The field measurement and GEOTRACK model predictions are also plotted in the same figure. Again, it is clear from the figure that both the model predictions are matching convincingly with a percentage difference of less than 10% and are falling in the range of field measurements. The vertical stresses at the ballast and subgrade surfaces as predicted by both the models are approximately 190 kPa and 50 kPa respectively.

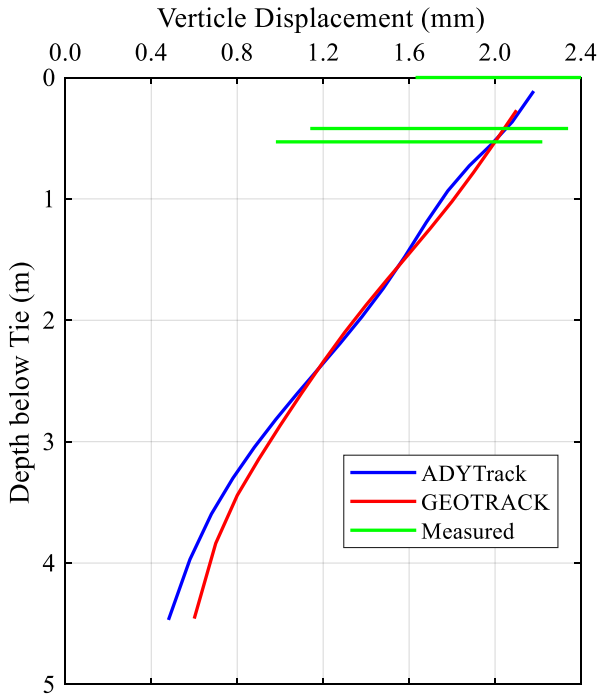

Figure 4-10. Vertical displacement below the wheel load and along the depth of the trackbed.

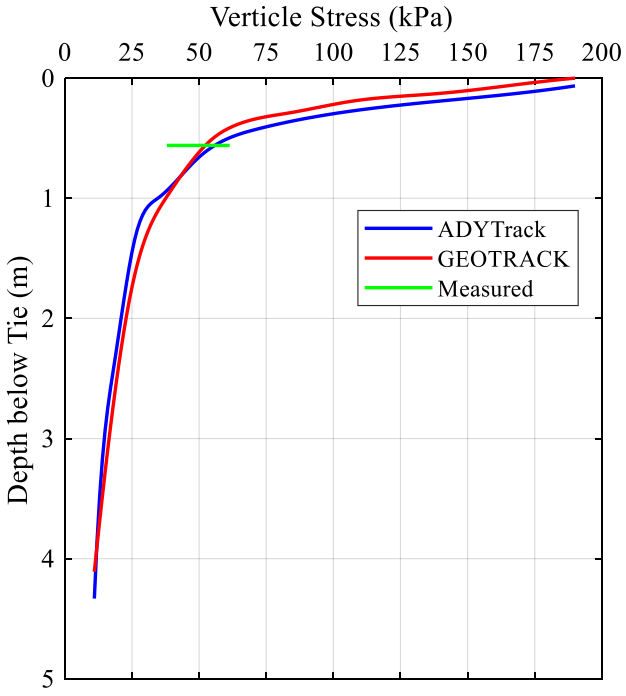

Figure 4-11. Vertical stress below the wheel load and along the depth of the trackbed.

Figure 4-12 shows the variations in vertical stress at the ballast surface along the tie under the wheel load (Tie-1) and two adjacent ties (Tie-2 and Tie-3). The vertical stress predictions by both the models are plotted in this figure. ADYTrack precisely captured the overall response including maximum stress and variation in stress along the ties, as predicted by the GEOTRACK, however it consistently and slightly underestimated the vertical stresses along the ties. For the Tie-1, Tie-2 and Tie-3, on average ADYTrack under predicted the stresses by around 15 %, 13% and 20%. Similarly, ADYTrack underestimated the vertical stress at the ballast surface and over predicted the stress at the subgrade surface in Figure 4-9 as well. These stress underestimations at the top layer can be since the substructure layers are modelled in ADYTrack as rigidly connected brick elements that can cause relatively faster and broader load distribution.

Vertical stress distribution along the rail beam at the top of subgrade due to a double axel railways bogie, as measured in the field and subsequently predicted by GEOTRACK and ADYTrack is shown in Figure 4-13. Overall response of the stress distribution is precisely predicted by both the models with a significant accuracy. The maximum vertical stress right below the wheel locations as calculated by ADYTrack and GEOTRACK are 57 kPa and 56 kPa respectively, as compared to field measurement of 55 kPa, with less than 5% difference.

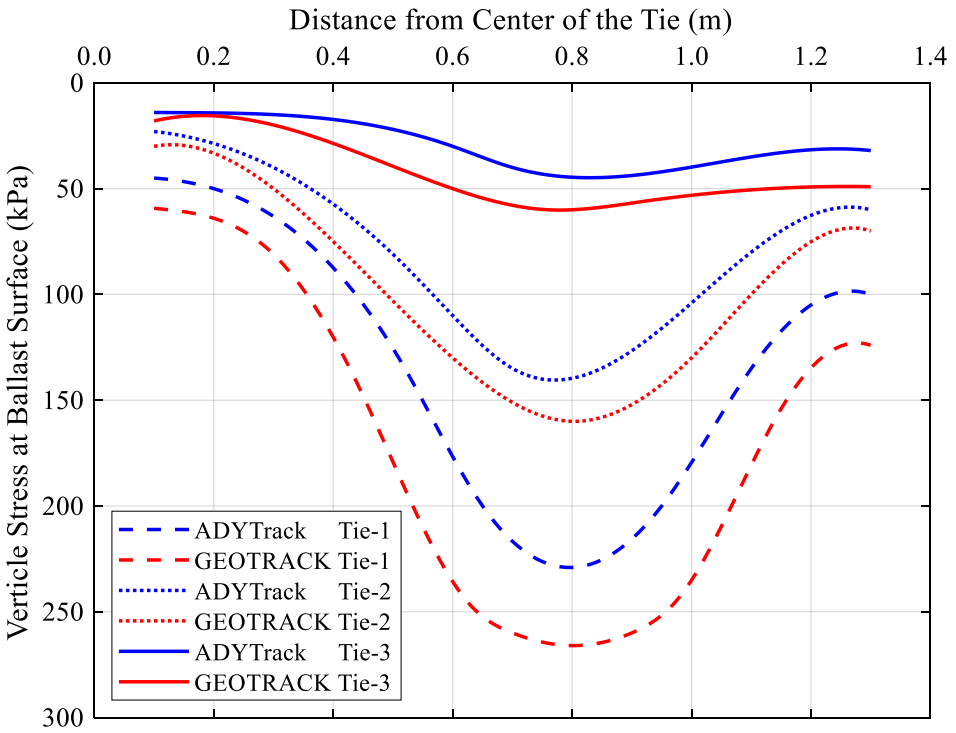

Figure 4-12. Vertical stress at the ballast surface along the tie under and adjacent to wheel load.

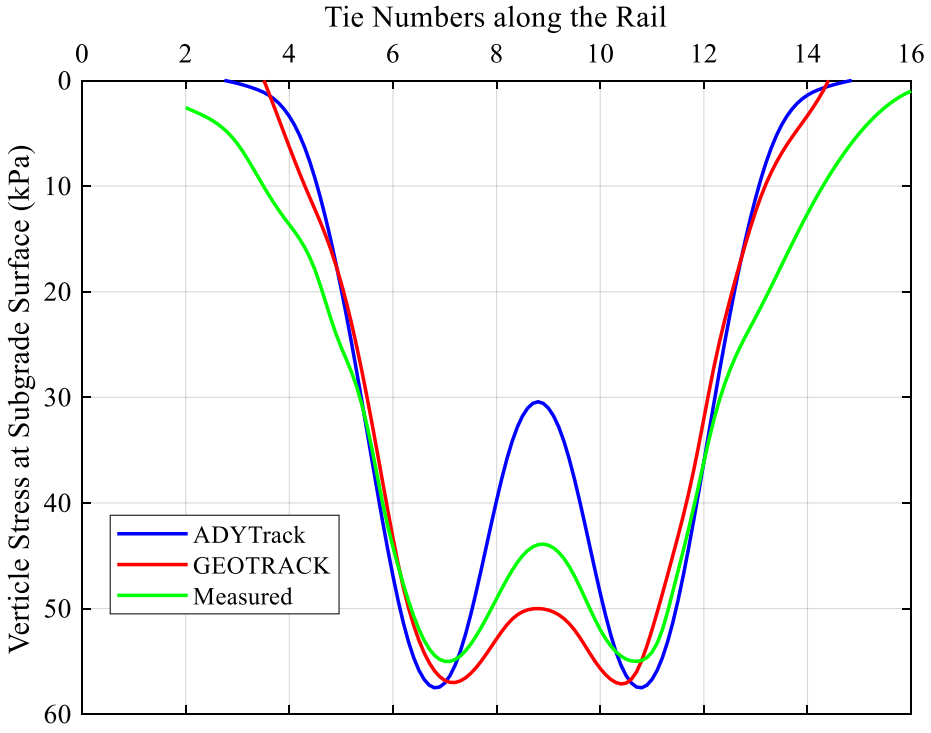

Figure 4-13. Vertical stress at the subgrade surface along the rail.

4.4. Summary

This chapter evaluated the newly built model, ADYTrack, to validate its applicability and accuracy. To verify its accuracy and reliability, it is evaluated against analytical solutions, commercial numerical solution and full scaled field measurements. A straight cantilever beam model with different loading conditions and element shapes was developed in ADYTrack and their response was compared with analytical solutions. Then ADYTrack calculations were examined against ANSYS results for a) a unit cube model and b) a single layered trackbed model. Finally, the capacity of ADYTrack was checked against full scale field test measurements at two different locations with different ground conditions. The results of both these ADYTrack models were further compared with the prediction by other renowned models developed. Comparison of results in all these models revealed that ADYTrack can capture the response of any physical phenomenon with considerable accuracy and reliability.

5. SENSITIVITY STUDY OF A RAILROAD TRACKBED USING ADYTRACK

Following the development of the random finite element based ADYTrack model and its successful validation and evaluation of results, a sensitivity study is conducted to learn the extent of the influence of intrinsic variability in various substructure layers on the performance of the railroad trackbed. The intrinsic variability is modelled such that the coefficient of variation (COV) is changed from 0-80% for each layer, one at a time while keeping all other layers' COVs at 10%. The performance of the trackbed is measured by calculating vertical displacements and stresses right below the wheel at various levels. Each analysis comprised of 5,000 realization (iterations) of Monte Carlo simulations, which was very resource (computing) intensive, therefore, the assistance of Center of Computational Assisted Science and Technology (CCAST) at NDSU was sought. The procedure of using CCAST facility is explained in the following section.

5.1. Usage of CCAST Facility at NDSU

For a single cycle of deterministic analysis of up to 1000 nodes using ADYTrack takes about little less than 3 seconds using a 16 GB random access memory on Intel(R) Core (TM) i7-7700 CPU@3.6GHz processor. This simply means that to perform a probabilistic analysis with 10,000 iterations, it should take more than 8 hours for the same sized FEM model. Therefore, assistance was sought from Center of Computationally Assisted Science and Technology (CCAST) facility at the University (NDSU). Below are the step by step procedures to run the simulations over the CCAST network.

1) Open an account with CCAST to get a user ID and password.
2) Install PuTTY and winSCP on your computer.
3) Log in to PuTTY and winSCP using CCAST ID and password.

4) Change the directory using “cd” command to get to the working directory over the CCAST network.
5) Copy the main code and script file in your working directory over the CCAST network. Script file is a set of user requirements and main instructions to convey to the CCAST network. User requirements will include the number of CPUs, estimated wall time, memory allocation, and main instruction may include loading a specific module (for instance, MATLAB) and name of the input file or main code to run.
6) Submit the job using “qsub” command followed by the name of the script file, for instance “qsub trail_1.pbs”.

5.2. Selection of Trackbed Parameters

To perform the sensitivity study using RFEM based ADYTrack, the same case study 2 (explained in section 4.3.2 and used for the validation and evaluation of ADYTrack) is used. The important deterministic and probabilistic parameters used for the sensitivity analysis are presented in the following sections:

5.2.1. Deterministic Parameters

A quarter (one fourth) model spanning over seven ties is considered for this study as Shahu et. al., [35] Feng Huang [36] and Mishra et.al. [24] have reported that boundary effects and stresses considerably dissipate beyond five ties. The full tie length is taken as 2.75m with shoulder width of 0.3m for ballast with side slope of 1V:1.5H and 6.5m horizontal shoulder at the natural ground surface level, which marks the lateral extent of the model. The substructure is divided into four layers, namely ballast, subballast, subgrade and natural soil of 350mm, 150mm 1000mm and 6500mm, respectively with a total depth of 8.0m. The other important properties used for rail, tie and substructure layers are presented in Table 4.4 (Model-2).

5.2.2. Coefficient of Variation of Intrinsic Substructure Variability

Coefficient of variation is the ratio of mean to the standard deviation and expressed in percentage. Phoon and Kulhawy [120] studied the variability in resilient modulus (E) of soils using correlations with pressuremeter test, dilatometer test and SPT N value test. The inherent variabilities (COV) reported in literature range between 15% and 68% [120], [121]. The range of E-values for substructure layers, adopted in the present study is 0% - 80%, which is slightly higher than reported values. The justification for this higher value is that given the variability of soil depends on the ratio of sampling volume to the sampling domain volume [89], meaning fewer samples over a large domain can significantly increase the COV, which is highly likely in case of railroad trackbed design. Another reason for choosing higher value is to meet the objective of this study (i.e., sensitivity analysis) and to quantify the variability in output fields due to a wide range of input variabilities.

The probabilistic structural performance of the trackbed was evaluated through the variations in vertical displacements, vertical stresses and track modulus against the variations in substructure layers, for 5,000 realizations of Monte Carlo simulations in each case using MATLAB, the details of which are explained previously. A range of 0-80% COVs in resilient modulus was covered, with five discrete COV values, for each substructure layer while keeping COV in resilient modulus of other layers at 10% in each realization, as presented in Table 5.1. Vertica responses (displacement and stress) are calculated at the center of top layered elements in each substructure layer, right below the wheel load.

The results are discussed separately for vertical displacements, vertical stresses and track modulus in the following sections.

Table 5.1. Combinations of COVs for different input variables used in this study.

COV in Ballast Modulus (%)	COV in Subballast Modulus (%)	COV in Subgrade Modulus (%)
0, 10, 20, 40, 80	10	10
10	0, 10, 20, 40, 80	10
10	10	0, 10, 20, 40, 80

5.3. Results and Discussions

5.3.1. Vertical Displacements

Unlike deterministic analysis, a series of calculations were performed using Monte Carlo simulations for every single variation in the input variable parameter, using RFEM based version of ADYTrack. In this study, 5,000 realizations were selected to calculate the structural performance of the railroad. In other words, vertical displacements were calculated at the top of each layer 5,000 times for each set of variations, presented in Table 5.1.

The results of vertical stress at the top of ballast layer for only one set of variations, i.e., in the resilient modulus of ballast layer with COVs of 10%, 20%, 40, and 80% are presented in Figure 5-1. The complete set of reliability plot at the top of substructure layers due to variations in resilient modulus of each layer are presented in Appendix-C. The histograms of vertical stress values are represented in gray color bars, whereas the equivalent normal distribution of these results is shown in red colored bell shape curve on the secondary y-axis (red in color). The same equivalent normal distribution curve can be used to calculate the reliability (inverse of probability of failure) or probability of occurrence or exceedance of any given stress magnitude. Alternatively, for any given probability of occurrence, for instance 1%, 5%, or 10%., the corresponding vertical stress can also be calculated.

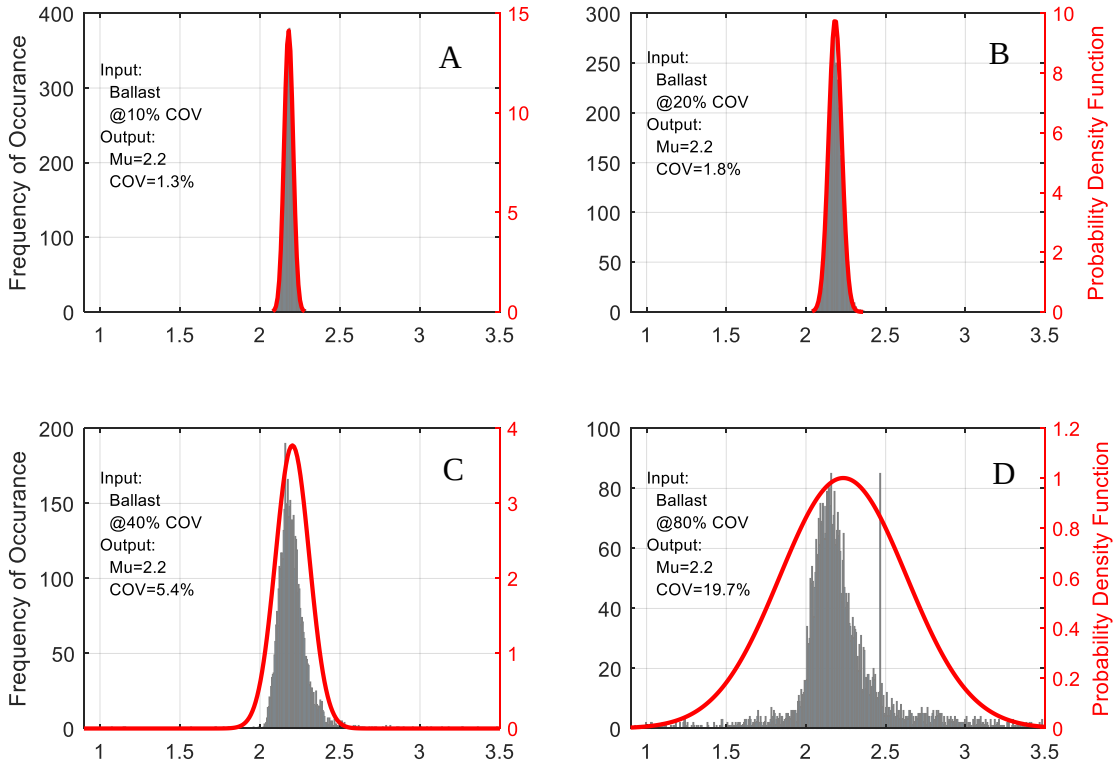

Figure 5-1. Reliability plot for vertical displacement at the top ballast layer due to variations (A=10%, B=20%, C=40%, and D=80%) in resilient modulus of ballast layer using Monte Carlo simulations.

Variations in COV of vertical displacement at the top of each substructure layer including rail beam due to variations (COV) in ballast layer is shown in Figure 5-2. The results show that COVs in vertical displacement increased with increasing COVs in ballast layer's resilient modulus values at the top of each substructure layers including rail. The range of variations in the response variable is observed between 1-19% for the selected range of variations in the ballast layer's resilient modulus values. Similar trend was also observed by Fernandes et. al. [109]. Additionally, at around 30% and beyond of COV in resilient modulus of ballast layer, the increase in vertical response becomes more pronounced and steeper for all the substructure layers including rail. Furthermore, the difference in response of all the substructure layers among themselves, for any given value of COV in resilient modulus of ballast layer, is less than 5%. The output variable

response in ballast layer is found to be most sensitive to variations in its own layer's resilient modulus values.

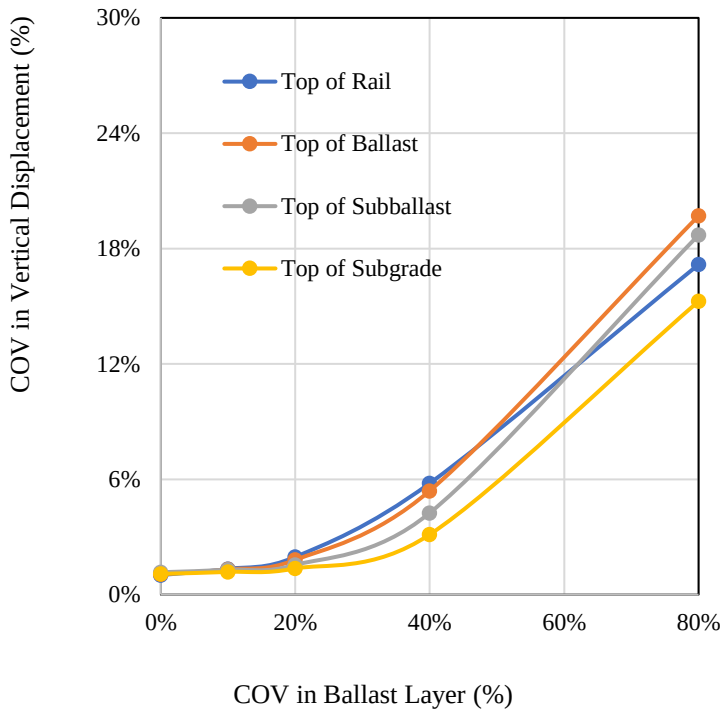

Figure 5-2. COVs (%) in vertical displacement at the top of substructure layers below the wheel load for a range of COVs (%) in the resilient modulus of ballast layer.

Similar trend is observed for the COVs in vertical displacements against COVs in resilient modulus of subballast layer, with relatively shorter range of 1-14% in the COVs of the vertical response, as shown in Figure 5-3. This short range may be attributed to the relatively smaller volume of the layer (150mm thick) compared to total volume of the model.

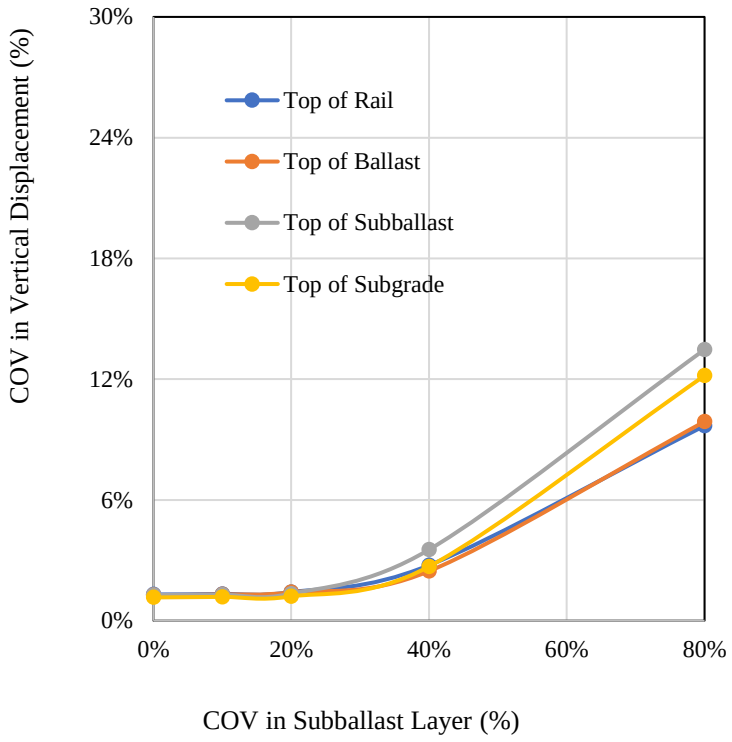

Figure 5-3. COVs (%) in vertical displacement at the top of substructure layers below the wheel load for a range of COVs (%) in the resilient modulus of subballast layer.

Figure 5-4 represents the variations in vertical displacements at the top of substructure layers below the wheel load against the variations in resilient modulus in subgrade layer. The results in this case are like the cases of ballast and subballast layers, except with a higher range of variations in the response variable, i.e., 1-26%. Variations in vertical displacements are found to be most sensitive to the variations in its own layer's resilient modulus values.

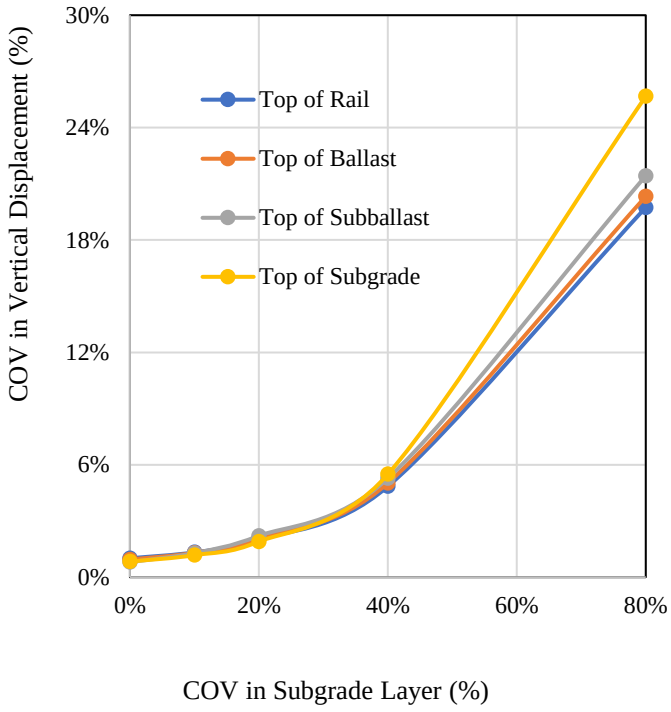

Figure 5-4. COVs (%) in vertical displacement at the top of substructure layers below the wheel load for a range of COVs (%) in the resilient modulus of subgrade layer.

5.3.2. Vertical Stresses

The vertical stresses were calculated using the same procedure as that of vertical displacements at the top of each layer due to variations in resilient modulus of each substructure layer. Figure 5-5 shows the reliability plot for vertical stress calculations at the top of ballast layer due to variations in resilient modulus in the same layer against COVs of 10%, 20%, 40%, and 80% in subplots of A, B, C, and D, respectively. Gray colored histogram shows the frequency of occurrence of a value of vertical stress. And red line represents the equivalent normal distribution of the results, which will be used to calculate the probabilities in various conditions to interpret the results. For more reliability plots, refer to Appendix-C.

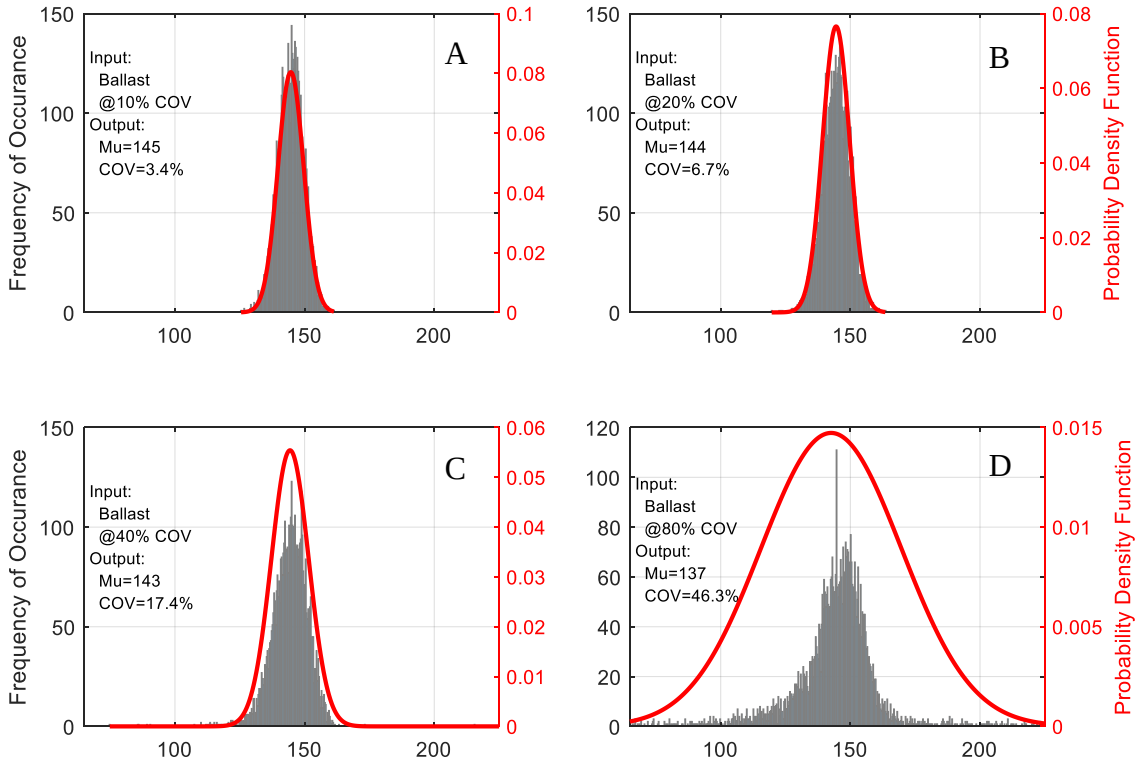

Figure 5-5. Reliability plot for vertical stress at the top ballast layer due to variations (A=10%, B=20%, C=40%, and D=80%) in resilient modulus of ballast layer using Monte Carlo simulations.

Vertical stresses were calculated at the center of elements situated at the top layer of each substructure layer below the wheel load for a range of COVs in the ballast layer's resilient modulus values. The variations in these stresses for each layer were then plotted against variations in the resilient modulus values of substructure layers (Figure 5-6). The variations in the stress increases with the increasing variations in the input variable. This increase switches the slope and takes a steeper slope at around 30% and above COV in ballast layer's resilient modulus. It is also noted that output response is relatively more sensitive to the variations in input variable of its own layer, i.e., ballast layer. Furthermore, the range of output variable observed is between 1-43% for given range of variations in input variable

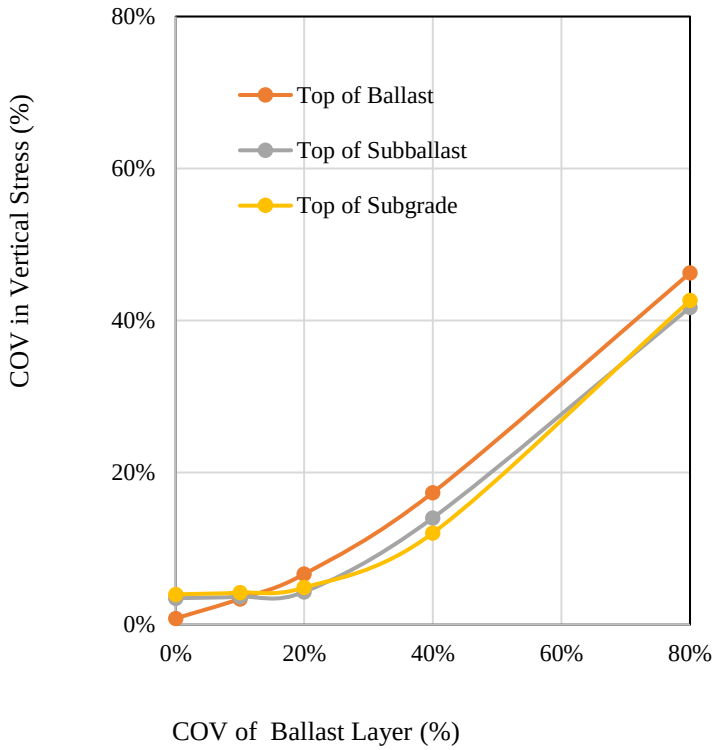

Figure 5-6. COVs (%) in vertical stresses at the top of substructure layers below the wheel load for a range of COVs (%) in the resilient modulus of ballast layer.

Figure 5-7 shows the COV in vertical stresses at the top of various substructure layers due to COVs in the resilient modulus of subballast layer. The results are again like that of ballast layer for the same response. However, the range of COVs in response variable is between 4-59% for the same range of variations in input variable, i.e., resilient modulus of subballast layer. The sensitivity level for variations in vertical stresses in both subballast and subgrade layers found to be identical at higher values of COVs in input variables, whereas ballast layer behavior remained relatively passive for the same higher values.

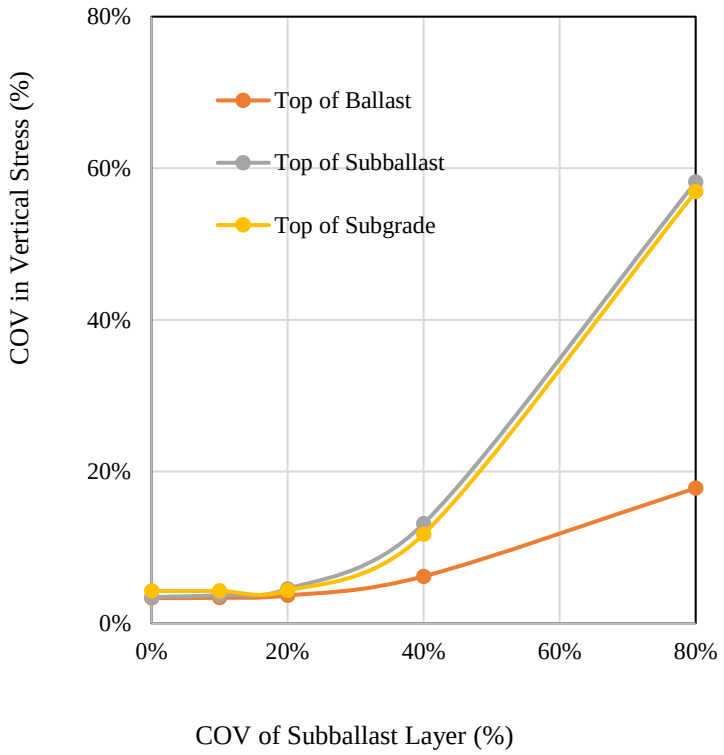

Figure 5-7. COVs (%) in vertical stresses at the top of substructure layers below the wheel load for a range of COVs (%) in the resilient modulus of subballast layer.

Figure 5-8 represents the case of vertical stress response against COVs in the resilient modulus of subballast layer. The range of COVs of response variable is observed between 1-66% for the same range of variations in the resilient modulus of subgrade layer. The variation in output variable is found to be more sensitive for the selected range of variations in input variables of subgrade layer (COVoutput=66%) and this is the highest variation observed in all the cases considered in this study. The response of subballast layer remained relatively less pronounced (COVoutput=47%), followed by even lesser pronounced response from ballast layer (COVoutput=17%), at the 80% COV for input variable considered in this work.

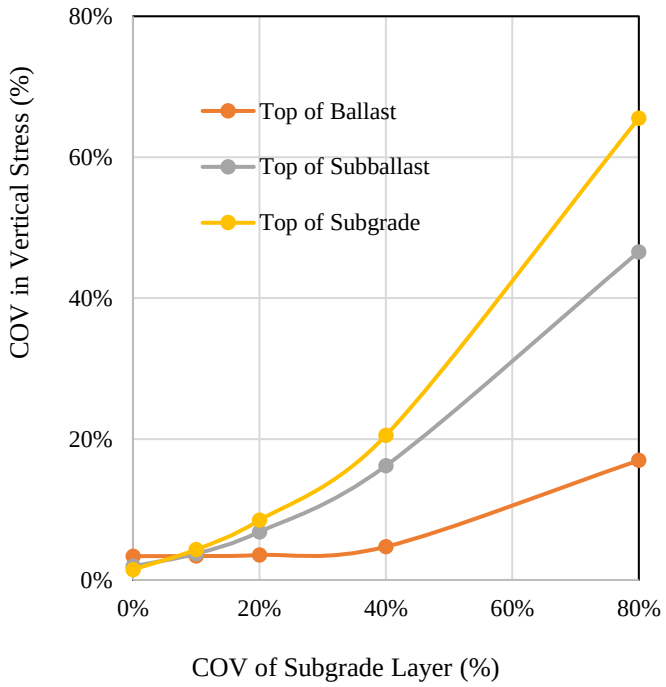

Figure 5-8. COVs (%) in vertical stresses at the top of substructure layers below the wheel load for a range of COVs (%) in the resilient modulus of subgrade layer.

5.3.3. Track Modulus

Track modulus, *u*, is a measure of trackbed performance including the effect of vertical stiffness of the foundation system including rail, ties and substructure and is calculated using Equation (5.1).

$$u = \frac{1}{4}\left(\frac{P}{\delta}\right)^{\frac{4}{3}}\left(\frac{1}{E_r I_r}\right)^{\frac{1}{3}} \tag{5.1}$$

where P is the applied wheel load, u is the vertical displacement at the rail below wheel load, E_r is the Young's modulus of rail and I_r is its second moment of area. Many including Steward and Selig [118], Raymond [122], Cai et. al. [123], Selig and Li [124] and recently Mishra et. al. [24] have used it as a trackbed performance indicator. The major factor influencing track modulus is subgrade resilient modulus, followed by subgrade thickness [124]. Track modulus in the current

study is calculated such that *P* is the wheel load, and *u* is vertical displacement at the node of rail beam below the wheel load.

Figure 5-9 explains probability distribution of track modulus due to variations in subgrade layer's resilient modulus for COV values of 10%, 20%, 40%, and 80% in subplots of A, B, C, and D, respectively. The complete set of reliability plots for variations in all substructure layers are appended in Appendix-C.

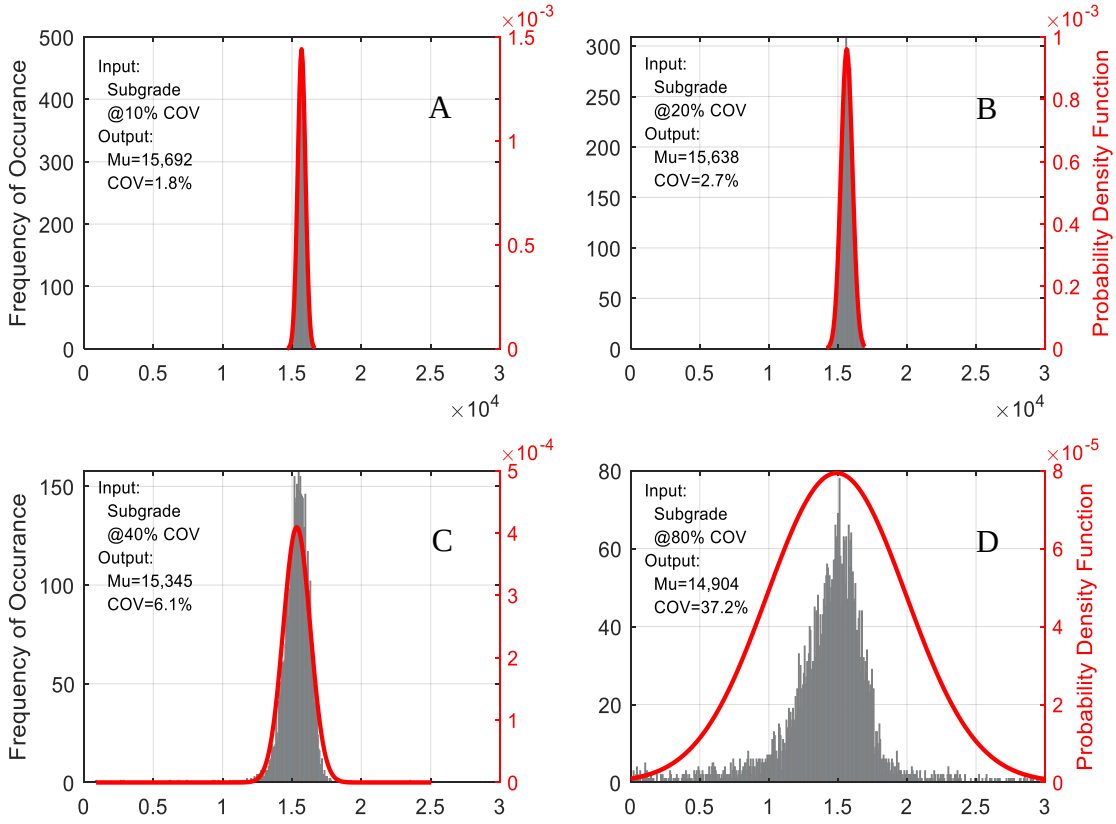

Figure 5-9. Reliability plot for track modulus due to variations (A=10%, B=20%, C=40%, and D=80%) in resilient modulus of Subgrade layer using Monte Carlo simulations.

Track modulus is plotted in Figure 5-10 for all cases against COV in resilient modulus (E) values of various substructure layers. Like the cases of vertical displacements and vertical stresses, variations in track modulus also increases with the increasing variation in substructure layers' E values. It also shows that the variations in subgrade E values are most influential to the variations

of track modulus values, followed by the variations in E values of ballast and subballast layers. At the maximum COV in E value, i.e., 80%, considered in this study, the COVs in the track modulus are calculated as 37.2%, 21.0% and 12.1% for subgrade, ballast and subballast layer, respectively. One reason for this order of influence can be the proportional volumetric contribution of corresponding layers in the trackbed, for instance, the thickness of subgrade, subballast, and ballast layers adopted for this analysis were 1000mm, 350m, and 150mm, respectively.

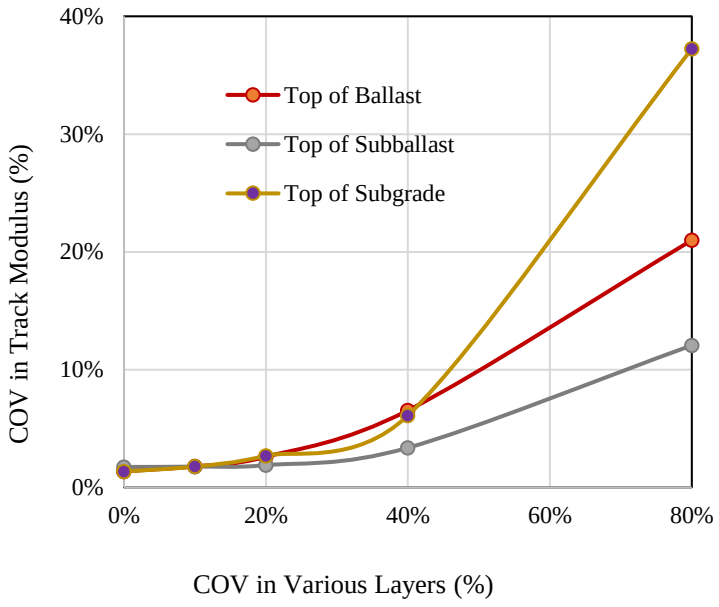

Figure 5-10. COVs (%) in track modulus for a range of COVs (%) in the resilient modulus of different substructure layer.

5.4. Summary

The results of sensitivity study, conducted to understand the influence of intrinsic variability in substructure layers on the performance of railroad trackbed, showed that with the increase in variability in the resilient modulus of substructure layers, there is a bilinear increase in the variability of the trackbed's responses, measured as vertical displacements and stresses, and track modulus. Also, it is observed that this increase in response changes the slope at COV of about 30-35%.

6. CONCLUSIONS AND FUTURE WORK

This chapter summarizes the conclusion and main outcomes of the studies performed under this dissertation, the major contributions of this study, and the potential future work.

6.1. Conclusions

The important conclusions based on this study can be drawn as follows:

1) A new numerical model, named ADYTrack, is developed based on the principles of random finite element (RFEM) to include the intrinsic variations in the material properties including built-in capability of performing reliability analysis using Monte Carlo simulations. This model can also incorporate the variations in geometric properties and loading conditions with some modifications in the base code.
2) The displacements of a straight cantilever beam model using eight nodal isoparametric hexahedral (brick) element in ADYTrack predicted the response in compression with 99.3% of accuracy when compared with analytical solutions, with a percentage difference of less than 10% when compared with the well-known models including GEOTRACK, and with 5-20% error when compared with the field measurements of full-scale experiments of test tracks with different subgrade conditions at the Facility for Accelerated Service Testing (FAST) Pueblo Colorado.
3) The sensitivity study of a trackbed model, using the RFEM based ADYTrack model revealed that the variations in vertical displacements, vertical stresses and track modulus in different layers increased with the increasing variations in resilient modulus (E) values of substructure layers, as expected, in all the cases considered in this study. However, this increases in variations of output variables remained somewhat less than the variations in their corresponding input variables. The reason behind this reduction

can be the mechanism of smooth and systematic load distribution of trackbed to the below layers with smaller E values.

4) An increase in rate of change is observed in output variables at around COV of 30% and above in the input variables, in most of the cases studied in this work.
5) The response of any substructure layer is mostly influenced by the variations in its own layer's E values for the complete range of COVs selected in this study.
6) The variations in subgrade E values are found to be strongly influential for the output variables considered in this study, especially track modulus.

6.2. Key Contributions

The key contribution of this study is the development of the new numerical model, ADYTrack, which is easy enough to be used by the railroad engineering practitioners yet robust enough to include any or all kinds of variations in the trackbed materials, and potentially also variations of the geometries and loadings with future development. These variations may include the materials' inherent variations; variations in the geometries of trackbed due to construction imperfections, rail operations or even weather conditions; or even the variations in loadings that might be caused by different car loads and/or their operating speeds.

The model with its integrated graphical user interface, is very easy to use by a railroad design engineer with minimal knowledge of FEM or probabilistic theories. As this model is specifically designed for trackbeds' structural analysis, it saves a lot of time first in building a finite element model, then in editing any or all parts of the trackbed, if required. This can significantly help design engineers to come up with the most economical and reliable trackbed design in short period of time. Furthermore, this model is validated against some field measurements, it is least prone to human mistakes in building a model.

This model can also significantly contribute in the field of cutting-edge research, for instance the trackbed performance can be evaluated under the anisotropic nature of trackbed construction materials, especially the subgrade or wooden ties etc. Also, design charts can be proposed for various combinations of variations in different substructure layers, for most generic trackbed shapes or geometries. More potential research topics are presented in next section.

Another key contribution of this study is building a numerical code from scratch in MATLAB. This will allow the researchers to modify the base code by adding or modifying a part of the algorithm to study the required aspect of trackbed.

6.3. Future Work

There is huge potential of future works using RFEM based ADYTrack due to its vibrancy and robustness to simulate the variations in different aspects of trackbed using principles of standard FEM as well as theories of probability and reliability. Some of the future work are outlined below:

The study of anisotropic behavior of materials is greatly underworked, primarily due to lack of modeling abilities, as most of the available models assume homogenous, and isotropic materials. However, ADYTrack has built-in capability of generating anisotropic conditions in all three directions, i.e., longitudinal, lateral, and/or traverse direction.

Track transition zone is another hot area of study these days in railways engineering due to chronical issues of riders' comfort and trackbed maintenance in these zones. The transition of track support due to the presence of structures like bridge decks or change of tie types etc., are difficult to model in standard packages of FEM models. However, ADYTrack has the potential to simulate these transitional conditions and analyze the performance of track due various loading types.

The studies of exploring critical spatially correlated length are reported in other applications of geotechnical engineering including retaining wall design, shallow bearing capacity and slope stability problems, using the principles of probability and reliability. A similar study can be performed for railroad trackbed. The identification of critical spatial correlation length can significantly contribute towards the site characterization and trackbed designs.

Another future work may be the development of design charts for various combinations of a range of variations in different components of the trackbed for some most common trackbed geometries and loading. This can be of great value in comparing different design options at preliminary or conceptual stage of trackbed designs. Last, but not the least, it can also help document different case studies with varying levels of variations in substructure layer properties, especially the moisture content which directly and significantly affect the resilient modulus.

REFERENCES

[1] C. C.w., “Track loading fundamentals--1 introduction: track and wheel loading,” *Railw. Gaz.*, vol. 106, pp. 45–48, 1957.

[2] M. Hetényi, “A General Solution for the Bending of Beams on an Elastic Foundation of Arbitrary Continuity,” *J. Appl. Phys.*, vol. 21, no. 1, pp. 55–58, Jan. 1950.

[3] H. C. Meacham, J. E. Voorhees, and J. G. Eggert, “Study of New Track Structure Design. Phase I,” Sep. 1968.

[4] A. P. S. Selvadurai, *Elastic analysis of soil-foundation interaction*. Elsevier Science, 1979.

[5] A. N. Talbot, “Stresses in Railroad Tracks,” American Railway Engineering Association (AREA), New York, N.Y., 1980.

[6] E. Winkler, “Die Lehre von der Elastizist und Festiekeit,” *Praga Dominicus*, vol. 1, no. 367, p. 182, 1865.

[7] D. M. Burmister, “The General Theory of Stresses and Displacements in Layered Soil Systems. III,” *J. Appl. Phys.*, vol. 16, no. 5, pp. 296–302, May 1945.

[8] E. T. Selig, C. S. Chang, C. W. Adegoke, and J. E. Alva-Hurtado, “A Theory for Track Maintenance and Life Prediction,” Buffalo, NY, Aug. 1979.

[9] C. W. Adegoke, C. S. Chang, and E. T. Selig, “Study of analytical models for track support systems,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 733, pp. 12–20, 1979.

[10] K.-J. Bathe and E. L. Wilson, “Numerical methods in finite element analysis.” Prentice-Hall, 1976.

[11] R. Pichumani, “Finite Element Analysis of Pavement Structures Using AFPAV Code (Linear Elastic Analysis).” 1973.

[12] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*. Englewood Cliffs, NJ: Prentice-Hall Inc., 1973.

[13] O. C. Zienkiewicz and Y. K. Cheung, “Finite elements in the solution of field problems.,” *Eng.*, vol. 220, no. 5722, pp. 507–510, 1965.

[14] O. C. Zienkiewicz, *The Finite Element Method*. London: McGraw-Hill, 1977.

[15] I. C. Chang, “Track foundation stresses under horizontal loads,” Illinois Institute of Technology, Illinois, 1975.

[16] K. H. Chu, “Track foundation stresses under vertical loading,” *Rail Int.*, vol. 8, no. 12, Dec. 1977.

[17] J. E. Crawford, “An Analytical Model for Airfield Pavement Analysis.” 1972.

[18] H. L. R., “User’s Manual for PSA: Three-Dimensional Elasticity Analysis of Periodically Loaded Prismatic Solids,” Davis, 1968.

[19] Q. L. Robnett, M. R. Thompson, R. M. Knutson, and S. D. Tayabji, "Development of a Structural Model and Materials Evaluation Procedures," 1976.

[20] S. D. Tayabji and M. R. Thompson, "Program ILLI-TRACK: A Finite Element Analysis of Conventional Railway Support System: User's Manual and Program Listing," Mar. 1976.

[21] C. L. Monismith and F. N. Finn, "Flexible Pavement Design: State-Of-The-Art 1975," *J. Transp. Eng.*, vol. 103, no. 1, Jan. 1977.

[22] B. F. Kallas, "Test procedures for characterizing dynamic stress-strain properties of pavement materials," *Transp. Res. Rec. J. Transp. Res. Board1*, vol. 162, 1975.

[23] C. S. Chang, C. W. Adegoke, and E. T. Selig, "Geotrack Model for Railroad Track Performance," *J. Geotech. Geoenvironmental Eng.*, vol. 106, no. 11, Nov. 1980.

[24] D. Mishra, S. Sharma, A. Shrestha, D. Li, and C. Basye, "GEOTRACK-2015: An Upgraded Software Tool for Railroad Track Analysis," in *2016 Joint Rail Conference*, 2016, p. V001T01A028.

[25] H. E. Stewrt and E. T. Selig, "Predicted and measured reslient response of track," *J. Geotech. Eng.*, vol. 108, no. 11, pp. 1423–1442, 1982.

[26] Stewart H.E., "User's Manual for GEOTRACK Computer Program (," 1988.

[27] J. M. Selig, E. T., and Waters, *Track geotechnology and substructure management*. London: Thomas Telford Services Ltd, 1994.

[28] Y. H. Huang, C. Lin, X. Deng, and J. Rose, *KENTRACK, A Computer Program for Hot-Mix Asphalt and Conventional Ballast Railway Trackbeds*. 1984.

[29] K. V. 2.0.1, "Program, Help File and Example Data Files," 2006.

[30] D. Li, J. Hyslip, T. Sussmann, and S. Chrismer, *Railway Geotechnics*. Boca Raton, FL, USA: CRC Press/Taylor & Francis Group, 2015.

[31] S. Liu, "KENTRACK 4.0: A Railway Trackbed Structural Design Program," University of Kentucky, 2013.

[32] J. G. Rose, E. R. Brown, and M. L. Osborne, "Asphalt Trackbed Technology Development – The First 20 Years," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1713, pp. 1–9, 2000.

[33] J. G. Rose, N. K. Agarwal, J. D. Brown, and N. Ilavala, "KENTRACK, A Performance-Based Layered Elastic Railway Trackbed Structural Design and Analysis Procedure–A Tutorial," in *Joint Rail Conference*, 2010, pp. 73–110.

[34] J. Ose, S. Liu, and R. R. Souleyrette, "Kentrack 4.0: A Railway Trackbed Structural Design Program," in *Joint Rail Conference*, 2014.

[35] J. T. Shahu, N. Kameswara Rao, and Yudhbir, "Parametric study of resilient response of tracks with a sub-ballast layer," *Can. Geotech. J.*, vol. 36, no. 6, pp. 1137–1150, 1999.

[36] H. Feng, "3D-models of Railway Track for Dynamic Analysis," KTH, Stockholm, 2011.

[37] W. Chen and Y. K. Cheung, "Three-dimensional 8-node and 20-node refined hybrid isoparametric elements," *Int. J. Numer. Methods Eng.*, vol. 35, no. 9, pp. 1871–1889, Nov. 1992.

[38] C. S. Jog, "A 27-node hybrid brick and a 21-node hybrid wedge element for structural analysis," *Finite Elem. Anal. Des.*, vol. 41, no. 11–12, pp. 1209–1232, Jun. 2005.

[39] R. H. Macneal and R. L. Harder, "A proposed standard set of problems to test finite element accuracy," *Finite Elem. Anal. Des.*, vol. 1, no. 1, pp. 3–20, Apr. 1985.

[40] K. Y. Sze and H. Fan, "An economical assumed stress brick element and its implementation," *Finite Elem. Anal. Des.*, vol. 21, no. 3, pp. 179–200, Jan. 1996.

[41] J. M. Duncan, "Factors of Safety and Reliability in Geotechnical Engineering," *J. Geotech. Geoenvironmental Eng.*, vol. 126, no. 4, pp. 307–316, Apr. 2000.

[42] S.-H. Dai and M.-O. Wang, *Reliability analysis in engineering applications*. Van Nostrand Reinhold, New York, 1992.

[43] J. T. Christian and G. B. Baecher, "Point-estimate method as numerical quadrature," *J. Geotech. Geoenvironmental Eng.*, vol. 125, no. 9, pp. 779–786, Sep. 1999.

[44] K. C. Foye, R. Salgado, and B. Scott, "Assessment of variable uncertainties for reliability-based design of foundations," *J. Geotech. Geoenvironmental Eng.*, vol. 132, no. 9, pp. 1197–1207, Sep. 2006.

[45] K.-K. Phoon and J. Ching, *Risk and reliability in geotechnical engineering*. Taylor & Francis, 2015.

[46] M. P. Mignolet and P. D. Spanos, "Simulation of homogeneous two-dimensional random fields: Part I-AR and ARMA models," *J. Appl. Mech. Trans. ASME*, vol. 59, no. 2, pp. S260–S269, 1992.

[47] P. D. Spanos and M. P. Mignolet, "Simulation of homogeneous two-dimensional random fields: Part II-MA and ARMA models," *J. Appl. Mech. Trans. ASME*, vol. 59, no. 2, pp. S270–S277, 1992.

[48] A. M. Yaglom, *Introduction to the theory of stationary random functions*. Mineola, New York: Dover Publications Inc., 1962.

[49] G. A. Fenton, "Error Evaluation Of Three Random-Field Generators," *J. Eng. Mech.*, vol. 120, no. 12, pp. 2478–2497, 1994.

[50] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Math. Comput.*, vol. 19, no. 90, p. 297, Apr. 1965.

[51] G. Matheron, "The intrinsic random functions and their applications," *Adv. Appl. Probab.*, vol. 5, no. 3, pp. 439–468, Dec. 1973.

[52] G. A. Fenton and E. H. Vanmarcke, “Simulation of Random Fields Via Local Average Subdivision,” *J. Eng. Mech.*, vol. 116, no. 8, pp. 1733–1749, Aug. 1990.

[53] G. A. Fenton, “Simulation and Analysis of Random Fields,” Princeton University, New Jersey, 1990.

[54] G. B. Baecher and J. T. Christian, *Reliability and statistics in geotechnical engineering*. London and New York: John Wiley and Sons, 2003.

[55] A. H.-S. Ang and W. H. Tang, *Probability concepts in engineering: emphasis on applications in civil & environmental engineering*. New York and London: John Wiley and Sons, 2007.

[56] M. E. Harr, *Reliability based design in civil engineering*. New York: McGraw-hill, 1987.

[57] E. Rosenblueth, “Point estimates for probability moments.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 72, no. 10, pp. 3812–4, Oct. 1975.

[58] T. F. Wolff, “Probabilistic Slope Stability in Theory and Practice,” *ASCE Geotehnical Spec. Publ.*, vol. 58, pp. 419–433, 1996.

[59] A. M. Hasofer and N. C. Lind, “Exact and Invariant Second-Moment Code Format,” *J. Eng. Mech. Div.*, vol. 100, no. 1, pp. 111–121, 1974.

[60] G. M. Filz and M. P. Navin, “Stability of Column-Supported Embankments,” *Virginia Transp. Res. Counc.*, vol. 06-CR13, p. 73, Nov. 2006.

[61] J. Tinsley Oden, T. Belytschko, I. Babuska, and T. J. R. Hughes, “Research directions in computational mechanics,” *Comput. Methods Appl. Mech. Eng.*, vol. 192, no. 7–8, pp. 913–922, Feb. 2003.

[62] G. Stefanou, “The stochastic finite element method: Past, present and future,” *Comput. Methods Appl. Mech. Eng.*, vol. 198, no. 9–12, pp. 1031–1051, Feb. 2009.

[63] V. Papadopoulos, G. Stefanou, and M. Papadrakakis, “Buckling analysis of imperfect shells with stochastic non-Gaussian material and thickness properties,” *Int. J. Solids Struct.*, vol. 46, no. 14–15, pp. 2800–2808, Jul. 2009.

[64] J. Argyris, M. Papadrakakis, and G. Stefanou, “Stochastic finite element analysis of shells,” *Comput. Methods Appl. Mech. Eng.*, vol. 191, no. 41–42, pp. 4781–4804, Sep. 2002.

[65] V. Papadopoulos and M. Papadrakakis, “Finite-element analysis of cylindrical panels with random initial imperfections,” *J. Eng. Mech.*, vol. 130, no. 8, pp. 867–876, Aug. 2004.

[66] G. Stefanou and M. Papadrakakis, “Stochastic finite element analysis of shells with combined random material and geometric properties,” *Comput. Methods Appl. Mech. Eng.*, vol. 193, no. 1–2, pp. 139–160, Jan. 2004.

[67] H. C. Noh, “Stochastic behavior of Mindlin plate with uncertain geometric and material parameters,” *Probabilistic Eng. Mech.*, vol. 20, no. 4, pp. 296–306, Oct. 2005.

[68] H. C. Noh, “Effect of multiple uncertain material properties on the response variability of in-plane and plate structures,” *Comput. Methods Appl. Mech. Eng.*, vol. 195, no. 19–22, pp. 2697–2718, Apr. 2006.

[69] H. C. Noh and H. G. Kwak, “Response variability due to randomness in Poisson’s ratio for plane-strain and plane-stress states,” *Int. J. Solids Struct.*, vol. 43, no. 5, pp. 1093–1116, Mar. 2006.

[70] S. Rahman and B. N. Rao, “An element-free Galerkin method for probabilistic mechanics and reliability,” *Int. J. Solids Struct.*, vol. 38, no. 50–51, pp. 9313–9330, Nov. 2001.

[71] S. Rahman and J. S. Kim, “Probabilistic fracture mechanics for nonlinear structures,” *Int. J. Press. Vessel. Pip.*, vol. 78, no. 4, pp. 261–269, Jun. 2001.

[72] S. Rahman and H. Xu, “A meshless method for computational stochastic mechanics,” *Int. J. Comput. Methods Eng. Sci. Mech.*, vol. 6, no. 1, pp. 41–58, 2005.

[73] C. O. Arun, B. N. Rao, and S. M. Srinivasan, “Stochastic meshfree method for elasto-plastic damage analysis,” *Comput. Methods Appl. Mech. Eng.*, vol. 199, no. 37–40, pp. 2590–2606, Aug. 2010.

[74] Z. Yang and X. F. Xu, “A heterogeneous cohesive model for quasi-brittle materials considering spatially varying random fracture properties,” *Comput. Methods Appl. Mech. Eng.*, vol. 197, no. 45–48, pp. 4027–4039, Aug. 2008.

[75] N. D. Lagaros and V. Papadopoulos, “Optimum design of shell structures with random geometric, material and thickness imperfections,” *Int. J. Solids Struct.*, vol. 43, no. 22–23, pp. 6948–6964, Nov. 2006.

[76] E. Bielewicz and J. Górski, “Shells with random geometric imperfections simulation - Based approach,” *Int. J. Non. Linear. Mech.*, vol. 37, no. 4–5, pp. 777–784, Jun. 2002.

[77] S. Rahman and A. Chakraborty, “A stochastic micromechanical model for elastic properties of functionally graded materials,” *Mech. Mater.*, vol. 39, no. 6, pp. 548–563, Jun. 2007.

[78] X. F. Xu and L. Graham-Brady, “A stochastic computational method for evaluation of global and local behavior of random elastic media,” *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 42–44, pp. 4362–4385, Oct. 2005.

[79] D. B. Chung, M. A. Gutiérrez, L. L. Graham-Brady, and F. J. Lingen, “Efficient numerical strategies for spectral stochastic finite element models,” *Int. J. Numer. Methods Eng.*, vol. 64, no. 10, pp. 1334–1349, Nov. 2005.

[80] M. Steven Greene, Y. Liu, W. Chen, and W. K. Liu, “Computational uncertainty analysis in multiresolution materials via stochastic constitutive theory,” *Comput. Methods Appl. Mech. Eng.*, vol. 200, no. 1–4, pp. 309–325, Jan. 2011.

[81] X. Yin, S. Lee, W. Chen, W. K. Liu, and M. F. Horstemeyer, "Efficient random field uncertainty propagation in design using multiscale analysis," *J. Mech. Des. Trans. ASME*, vol. 131, no. 2, pp. 0210061–02100610, Feb. 2009.

[82] X. Yin, W. Chen, A. To, C. McVeigh, and W. K. Liu, "Statistical volume element method for predicting microstructure-constitutive property relations," *Comput. Methods Appl. Mech. Eng.*, vol. 197, no. 43–44, pp. 3516–3529, Aug. 2008.

[83] Gaurav and S. F. Wojtkiewicz, "Efficient spectral response of locally uncertain linear systems," *Probabilistic Eng. Mech.*, vol. 25, no. 4, pp. 419–424, Oct. 2010.

[84] Gaurav, S. F. Wojtkiewicz, and E. A. Johnson, "Efficient uncertainty quantification of dynamical systems with local nonlinearities and uncertainties," *Probabilistic Eng. Mech.*, vol. 26, no. 4, pp. 561–569, Oct. 2011.

[85] E. A. Johnson, S. F. Wojtkiewicz, L. A. Bergman, and B. F. Spencer, "Observations with regard to massively parallel computation for Monte Carlo simulation of stochastic dynamical systems," *Int. J. Non. Linear. Mech.*, vol. 32, no. 4, pp. 721–734, Jul. 1997.

[86] W. Puła and M. Chwała, "Random bearing capacity evaluation of shallow foundations for asymmetrical failure mechanisms with spatial averaging and inclusion of soil self-weight," *Comput. Geotech.*, vol. 101, pp. 176–195, Sep. 2018.

[87] D. V. Griffiths, G. A. Fenton, and N. Manoharan, "Undrained Bearing Capacity of Two-Strip Footings on Spatially Random Soil," *Int. J. Geomech.*, vol. 6, no. 6, pp. 421–427, Nov. 2006.

[88] D. V. Griffiths and G. A. Fenton, "Bearing capacity of spatially random soil: the undrained clay Prandtl problem revisited," *Géotechnique*, vol. 51, no. 4, pp. 351–359, 2001.

[89] G. A. Fenton and D. V Griffiths, "Bearing-capacity prediction of spatially random $c - \varphi$ soils," *Can. Geotech. J.*, vol. 40, no. 1, pp. 54–65, Feb. 2003.

[90] G. A. Fenton, D. V. Griffiths, and W. Cavers, "Resistance Factors for Settlement Design," *Can. Geotech. J.*, vol. 42, no. 41572279, pp. 2016–2019, 2005.

[91] Y. Lua and R. Sues, "Probabilistic Finite-Element Analysis of Airfield Pavements," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1540, pp. 29–38, Jan. 1996.

[92] J. D. Lea and J. T. Harvey, "Using spatial statistics to characterise pavement properties," *Int. J. Pavement Eng.*, vol. 16, no. 3, pp. 239–255, 2015.

[93] S. Caro, D. Castillo, and M. Sánchez-Silva, "Methodology for Modeling the Uncertainty of Material Properties in Asphalt Pavements," *J. Mater. Civ. Eng.*, vol. 26, no. 3, pp. 440–448, Mar. 2014.

[94] A. W. Ali, A. R. Abbas, M. Nazzal, and K. Sett, "Incorporation of subgrade modulus spatial variability in performance prediction of flexible pavements," 2013.

[95] A. Alhasan, A. Ali, D. Offenbacker, O. Smadi, and C. Lewis-Beck, "Incorporating spatial variability of pavement foundation layers stiffness in reliability-based mechanistic-empirical pavement performance prediction," *Transp. Geotech.*, vol. 17, pp. 1–13, Dec. 2018.

[96] D. Zhu, D. V. Griffiths, and G. A. Fenton, "Worst-case spatial correlation length in probabilistic slope stability analysis," *Géotechnique*, vol. 69, no. 1, pp. 85–88, Jan. 2019.

[97] A. Mouyeaux, C. Carvajal, P. Bressolette, L. Peyras, P. Breul, and C. Bacconnet, "Probabilistic stability analysis of an earth dam by Stochastic Finite Element Method based on field data," *Comput. Geotech.*, vol. 101, pp. 34–47, Sep. 2018.

[98] N. Luo and R. J. Bathurst, "Reliability bearing capacity analysis of footings on cohesive soil slopes using RFEM," *Comput. Geotech.*, vol. 89, pp. 203–212, Sep. 2017.

[99] B. K. Low, J. Zhang, and W. H. Tang, "Efficient system reliability analysis illustrated for a retaining wall and a soil slope," *Comput. Geotech.*, vol. 38, no. 2, pp. 196–204, 2011.

[100] M. K. Lo and Y. F. Leung, "Probabilistic Analyses of Slopes and Footings with Spatially Variable Soils Considering Cross-Correlation and Conditioned Random Field," *J. Geotech. Geoenvironmental Eng.*, vol. 143, no. 9, p. 04017044, Sep. 2017.

[101] D.-Q. Li, X.-H. Qi, L.-M. Zhang, and C.-B. Zhou, "Effect of spatially variable shear strength parameters with linearly increasing mean trend on reliability of infinite slopes," *Struct. Saf.*, vol. 49, pp. 45–55, Jul. 2014.

[102] S.-H. Jiang, D.-Q. Li, Z.-J. Cao, C.-B. Zhou, and K.-K. Phoon, "Efficient System Reliability Analysis of Slope Stability in Spatially Variable Soils Using Monte Carlo Simulation," *J. Geotech. Geoenvironmental Eng.*, vol. 141, no. 2, p. 04014096, 2014.

[103] D. V. Griffiths and G. A. Fenton, "Probabilistic Slope Stability Analysis by Finite Elements," *J. Geotech. Geoenvironmental Eng.*, vol. 130, no. 5, pp. 507–518, May 2004.

[104] D. V Griffiths, J. Huang, and G. A. Fenton, "Probabilistic infinite slope analysis," *Comput. Geotech.*, vol. 38, pp. 577–584, 2011.

[105] J. Oscarsson, "Simulation of Train-Track Interaction with Stochastic Track Properties," *Veh. Syst. Dyn.*, vol. 37, no. 6, pp. 449–469, 2002.

[106] M. Mehrali, S. Mohammadzadeh, M. Esmaeili, and M. Nouri, "Investigating vehicle-slab track interaction considering random track bed stiiness," *Sci. Iran. A*, vol. 21, no. 1, pp. 82–90, 2014.

[107] H. A. El-Ghazaly and M. A. Al-Hajry, "Probabilistic finite element analysis of railway tracks under multiple loads," *Comput. Struct.*, vol. 50, no. 1, pp. 147–158, Mar. 1994.

[108] B. F. Arbabi and C. U. Loh, "Reliability Analysis of Railroad Tracks," *J. Struct. Eng.*, vol. 117, no. 5, pp. 1435–1447, 1991.

[109] V. Alves Fernandes, F. Lopez-Caballero, and S. Costa d'Aguiar, "Probabilistic analysis of numerical simulated railway track global stiffness," *Comput. Geotech.*, vol. 55, pp. 267–276, Jan. 2014.

[110] B. A. Zeldin and P. D. Spanos, “On Random Field Discretization in Stochastic Finite Elements,” *J. Appl. Mech.*, vol. 65, no. 2, pp. 320–327, 1998.

[111] W. Puła, J. M. Pieczyńska-Kozłowska, and M. Chwała, “Search for the Worst-Case Correlation Length in the Bearing Capacity Probability of Failure Analyses,” in *Geotechnical Special Publication*, 2017, no. GSP 283, pp. 534–544.

[112] A. Ali, A. V. Lyamin, J. Huang, S. W. Sloan, and M. J. Cassidy, “Effect of spatial correlation length on the bearing capacity of an eccentrically loaded strip footing.” Tongji University, pp. 312–317, 2016.

[113] D.-Q. Li, T. Xiao, L.-M. Zhang, and Z.-J. Cao, “Stepwise covariance matrix decomposition for efficient simulation of multivariate large-scale three-dimensional random fields,” *Appl. Math. Model.*, vol. 68, pp. 169–181, Apr. 2019.

[114] A. Der Kiureghian and J.-B. Ke, “The stochastic finite element method in structural reliability,” *Probabilistic Eng. Mech.*, vol. 3, no. 2, pp. 83–91, Jun. 1988.

[115] C.-C. Li and A. Der Kiureghian, “Optimal Discretization of Random Fields,” *J. Eng. Mech.*, vol. 119, no. 6, pp. 1136–1154, Jun. 1993.

[116] D. M. Dilip and G. L. Sivakumar Babu, “Influence of Spatial Variability on Pavement Responses Using Latin Hypercube Sampling on Two-Dimensional Random Fields,” *J. Mater. Civ. Eng.*, vol. 26, no. 11, p. 04014083, 2014.

[117] G. I. Schuëller, “Developments in stochastic structural mechanics,” *Arch. Appl. Mech.*, vol. 75, no. 10–12, pp. 755–773, Oct. 2006.

[118] H. E. Stewart and E. T. Selig, “Predicted and Measured Resilient Response of Track,” *J. Geotech. Eng. Div.*, vol. 108, no. 11, pp. 1423–1442, 1982.

[119] C. W. Adegoke, “Elastic and inelastic deformation response of track structures under train loadings,” State University of New York, Buffalo, 1978.

[120] K.-K. Phoon and F. H. Kulhawy, “Evaluation of geotechnical property variability,” *Can. Geotech. J.*, vol. 36, no. 4, pp. 625–639, Nov. 1999.

[121] K. K. Phoon and F. H. Kulhawy, “Characterization of geotechnical variability,” *Can. Geotech. J.*, vol. 36, no. 4, pp. 612–624, 1999.

[122] G. P. Raymond, “Analysis of Track Support and Determination of Track Modulus,” *Transp. Res. Rec. J. Transp. Res. Board*, no. 1022, 1985.

[123] Z. Cai, G. P. Raymond, and R. J. Bathurst, “Estimate of Static Track Modulus Using Elastic Foundation Models,” *Transp. Res. Rec. J. Transp. Res. Board*, no. 1470, pp. 65–72, 1994.

[124] E. T. Selig and D. Li, “Track Modulus: Its Meaning and Factors Influencing It,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1470, 1994.

[125] T. R. Chandrupatla and A. D. Belegundu, *Introduction to Finite Elements in Engineering*, 2nd ed. Prentice-Hall, Inc., 1997.

APPENDIX A. BASE CODE OF ADYTRACK MODEL

A.1. Reading the User Defined Inputs Parameters

RAIL

Magnitude	Description	Validate
1.65	Full Gauge Length (m)	
2.070E+08	Young's Modulus (kPa)	
8.650E-03	Cross Sectional Area (m2)	
2.158E-05	MOI (Iy) m^4	
0.30	Poison'a ratio	
1	# of Elements b/w 2-Tie edges	
1.200E+06	Rail Tie Spring Constant (kN/m)	

TIE

7	No. of Ties to Consider
0.210	Thickness (m) {Z-direction}
0.250	Width (m) {X-direction}
2.750	Full Length (m) {Y-direction}
0.550	Spacing (m)
1.06E+07	Young's Modulus (kPa)
0.37	Poison'a ratio
5	# of Elements in Half Tie
0.2	Incremental rate beyond Tie Length
0.2	Incremental rate in Last Layer

LOAD

145.0	Full Wheel Load (kN)
500	Tolerance in Young Mod(kPa)
4	Model Size (2=Half | 4=Quarter)

General Information

0	Stress analysis (Yes = 1& No = 0)
1	Determnstc(0) | Probabstc(1)
5,000	Total Number of Iterations/Simulations

Layers

Magnitude	Description	Ballast Layer-1	Sub-Bal Layer-2	Sub-Gr Layer-3	Natural Layer-4	Validate
4	Max # of Layers to consider					
1	R-Shoulder width at Top (m)	0.300	0.000	0.000	6.200	OK
2	Side slope (x:1)	1.5	1.5	1.5	0.0	OK
3	Thickness of the Layer (m)	0.350	0.150	1.000	6.500	0
4	No of sub-layers	3	2	3	3	2
5	Initial Young's Modulus (kPa)	400,000	200,000	90,000	15,000	
6	Poison'a ratio	0.37	0.37	0.40	0.40	
7	Linear=0 | Non-Linear=1	0	0	0	0	
8	Non-Linearity Coeff. K3	8.0	1.0	1.0	1.0	
9	Non-Linearity Coeff. K4	0.50	0.50	1.00	1.00	
10	Gibson Soil Coeff.	0	0	0	2,500	
11	Determnstc(0) | Probabstc(1)	1	1	1	1	
12	Coeff. Of variation (COV) of "E"	10%	10%	10%	10%	
13	Spatial Correlation Length (Lx)	1.00	1.00	1.00	1.00	
14	Spatial Correlation Length (Ly)	1.00	1.00	1.00	1.00	
15	Spatial Correlation Length (Lz)	1.00	1.00	1.00	1.00	
16	Random Mesh Scalling Factor(x)	2	4	4	4	
17	Random Mesh Scalling Factor(y)	4	4	4	4	
18	Random Mesh Scalling Factor(z)	4	4	4	4	

Figure A-1. Screen shot of Microsoft Excel sheet to read the inputs for RFEM based analysis using ADYTrack model.

A.2. Main Program 'ADYTrack_4'

```
clear; clc; tic;
%Half scaled Model | Material Non-Linearity | Monte Carlo Simulations

global C NECT LoE GTN L1n nLN nLE NSL GSM Fex AnType MdSz WLd
global ToE GTE StrMat TnoL Nr D3 Er LD NL nnr Nitr NTEG Set1
global Uzrl Uzbl Uzsb Uzsg Szbl Szsb Szsg Tmod

% profile on
Prepare              % Read Inputs, Build Coord and NECT Table
disp(GTN)            % Display Grand Total of Nodes in the system

%%
COVs=[0 10 20 40 80]/100;    % COV Values ??????????????????????
nSet= length(COVs);          % nSet = set of calcs,  COV=0,10,20,40,..
```

```
OutputAll=zeros(32,nSet);
for set= 1:nSet
  LD(12, 2) = COVs(set);            % WHICH LAYER ??????????????????
  OutputPrint(1,1);

 for i=1 : 1+(Nitr-1)*AnType    % Iterations of Monte Carlo
Simulations
  if AnType==1; GenRandE;   end % Generate Random E (Young Modulus)
values

  StifMat;      % Construct Stress-Strain, Loc & Glob Stiff. Matrices
  BCond;        % Apply Boundary Cond.s & Displacement Vectors
  Solve         % Solve simulteneous equations for Nodal Displacements
  if sum(LD(7,:)) >0;   ChkNonLin;    end     % Check Material Non-
Linearity

  OutputPrint(i,2);
%   if mod(i,50)==0;      disp([i toc ]);      end
 end

%%
% Plots               %Plot Displacement, Stresses & Risk-Reliability
Graphs
OutputPrint(0,3);
OutputAll(:,set)= Set1;
set
end
% profreport;
toc
```

A.3. All Main and Nested-Functions

i. Function 'Prepare'

```
function Prepare        %Read Inputs, Build Coord and NECT Table

%% ****************************** INPUTS
*********************************
%*********************************************************************
****
global RTK NoT TTk TWd TLn TSpc RG Nr NTE1 NTEG lre ltec lteo LD NL
AnType
global nnr nnto nnt StrMat TnoL WLd TcD Tol Nitr Pzr GTN GTE MdSz
StrAna
global NECT nLE L1n T LoE ToE Er
A = xlsread('Input_4_SI.xlsx'); % Read the user inputs from Excel

StrMat = zeros(2,6);    % [E A G J Iy Iz].. Row1=Rail.. Tow2=Tie
Pzr = zeros(1,2);       % Poison's Ratio
MdSz = A(26,1);         % Model size (Half or Quarter)
%GENERAL
  StrAna=A(31,1);       % Stress analysis to performed? Yes=1, No=0
```

```
  AnType=A(32,1);       % Analysis Type? Deterministic=0,
Probabilistic=1
  Nitr=A(33,1);         % Number of itrations to perfrom problstic
analysis

%RAIL
  RG= A(1,1)/2;         % Rail Gauge (m)
  StrMat(1,:)= [A(2:3,1)' 1 1 A(4,1) 3.5e-6];   %RAIL Strength Matrix
  Pzr(1)= A(5,1);       % Rail Poison's ratio
  Nr= A(6,1);           % No.of Elem between 2-Ties
  RTK= A(7,1);          % Rail-Tie Spring Stiffness Coefficient (kN/m)

%TIE
  NoT= A(12,1);         % No. of Ties to consider
  TTk= A(13,1);         % Tie Thickness
  TWd= A(14,1);         % Tie Width
  TLn= A(15,1)/2;       % Tie Length
  TSpc=A(16,1);         % Tie C/C Spacing
  StrMat(2,:)= [A(17,1) TTk*TWd  1  1 ...       %Tie Strength Matrix,
                TWd*TTk^3/12  TTk*TWd^3/12];
  Pzr(2)=A(18,1);       % Tie Poison's ratio
  NTE1=A(19,1);         % #of Elem in 1/2 Tie
  NTEG=round(RG/TLn*NTE1);% #of Elem w/in Rail & CL
  TcD= 0.21;            % Tie center depth (m)
  switch MdSz;  case 2; TnoL=fix(NoT/2)+1;      % TnoL= Tie# to put
load on
                case 4; TnoL=1;                 end
  iry=A(20,1)+1;        %Incremental Rate beyond Tie Length (y-coord.
only)
  irz=A(21,1)+1;        %Incremental Rate in Last Layer(z-coord. only)

%LOAD
  WLd= 2*A(24,1)/MdSz;  % Wheal_Load (kN)
  Tol= A(25,1);         % Tolerance in Young's Mod. /NonLinear
analysis

%Soil
  NL= A(1,4);           % No of main Layers
  LD= A(3:20, 6:5+NL);  % LD= Layers Data

%Assign and store Node and Element numbers for the whole model.
lre = TSpc/Nr;                  % Length of Rail Elments
ltec= RG/NTEG;                  % Length of Tie element within Rails
(C/C)
lteo= (TLn-RG)/(NTE1-NTEG);     % Length of Tie elements outside Rails
nnr = (NoT-1)*Nr+1;             % Number of Nodes along each Rail
nnto= (NTE1-NTEG);              % No. of Nodes along Tie outside each
Rail
nnt = NTE1+1;                   % No. of Nodes along each Tie
```

```
%% ***************************** COORDINATES
*****************************
%*********************************************************************
****
% This function creates Coordinates for 1/4 model with in the
following
% sequence; Rails, Ties and Bricks. Node numbering was croocked to
adjust
% Tie nodes at their respecitve locations. This mechanism help all
Lines
% (DoF=6) at the top right of the Global K matrix (GSM) and all Bricks
% (DoF=3) at lower right corner of GSM.
% C(x, y, z) }.............{row# =node#....col= x,y,z coord}
global C nnSL nLN NSL v mGS
    QW=zeros(NoT,nnt);          o=0;
    xV=0:lre:(nnr-1)*lre;       %All X-Coord in single vector

    NSL=sum(LD(4,1:NL));        %No of Sub-Layers
    nnSL=zeros(1,NSL+1);        %No of Nodes in each Sub-Layer
    zTh=zeros(1,NSL+1);
    th=LD(3,:)./LD(4,:);        %Thickness of each Sub-Layer
    zTh(1)=TcD;                 c=1;
    for i=1:NL-1
        zTh(c+1: c+LD(4,i))=th(i);  %Store thickness of each SL but
Last
        c=c+LD(4,i);
    end
    %Incremental Z-coordinates for LAST LAYER (Natural SOIL)
    zL= LD(3,end)/((1-irz^LD(4,end))/(1-irz))/irz;
    for i=1:LD(4,end)           %Run for all sublayers of Last main
Layer
        zTh(c+1)= irz^i *zL;
        c=c+1;
    end
    zV= cumsum(zTh);            %All Z-Coord in single vector
%-----------------------------------------------------------------
%RAILS
    C(1:nnr,:)=[xV; repmat(RG,[1 nnr]); zeros(1,nnr)]';       c=nnr;
%-----------------------------------------------------------------
%TIES
    y22=cat(2, repmat(ltec,[1 ,NTEG]), repmat(lteo,[1 nnto]));
    y2 =cat(2,0,cumsum(y22));
    z2 =repmat(TcD,[1 nnt]);
    for i=1:NoT
        x2= repmat(0+(i-1)*TSpc,[1 nnt]);
        QW(i,:)= c+1: c+nnt;
        C(c+1:c+nnt,:) =[x2; y2; z2]';
        c=c+nnt;
    end
%-----------------------------------------------------------------
%Build a X-section of Nodes in YZ-Plane with full Trapizoidal Geometry
    lc=0;   Ye=0;   Ze=0;   np=nnt;     %Starting values
```

```
    % Y-coordinates (vector form) along the tie
    yp =cat(2,repmat(ltec,[1 NTEG]),repmat(lteo,[1 nnto]));
for i = 1:NL            %Jump for each main-layers

  for j = 1:LD(4,i)     %Run for each sub-layer (TOP OF EACH SUB-
LAYER)!!!!
    lc=lc+1;            %Loop counter,
    % Calculate Additional Length on the (SHOULDER and/or SLOPE)
    if i==1 && j==1;    dl= LD(1,i);                        %Only
Shoulder
    elseif j==1;        dl= LD(2,i-1)*th(i-1) +LD(1,i);
%Slope+Shoulder
    else;               dl= LD(2,i)*th(i);                  %Slope
Only
    end
      %Calculate No. of New Nodes and their distances
    if dl>0
        nn0=log(1-dl*(1-iry)/iry/lteo)/log(iry);
        nn= max(1, floor(nn0));             %Number of NEW NODES
        ln0=dl/iry/((1-iry^nn)/(1-iry));    %Length of 1st NEW ELEMENT
        yn= ln0*iry.^(1:nn);                %Length vector for new
nodes
    else
        nn=0;
        yn=[];
    end
    n =np +nn;          %Total # of nodes @TOP of ith Sub Layer
    if i==1 && j==1
        q=nn;           % q= new nodes at top shoulder
    end

    y2= cat(2,yp,yn);               %Length vector @TOP of ith Sub
Layer
    y = cat(2,0,cumsum(y2));
    Ye= [Ye y];
    Ze= [Ze repmat(zV(lc),[1 n])];  %Cumm Vectors of coord

    %Store data for next loops [nnSL => no@ nodes on top@ each Sub-
Layer]
    nnSL(1,lc) =n;
    yp=y2;
    np=n;
    %REPEAT 2ND-LAST SUB-LAYER of NODES TO ACCOUNT FOR THE LAST SUB-
LAYER
    if i==NL && j==LD(4,i)
        Ye= [Ye y];
        Ze= [Ze repmat(zV(end),[1 n])];
    end
  end
end
    Ye=Ye(2:end);
    Ze=Ze(2:end);                               %Remove 1st zeros
```

```
    nnSL(end)=max(nnSL);
    n1x =length(Ye)+1;                          %Total Nodes/X-section
    L1n= zeros(length(nnSL), nnSL(end), nnr);   %Nodes #s for all X-
sec

%RUN ALONG X-AXIS (for each X-section)
for i=1:nnr                 %Run for each X-section
%1st Layer
    if mod(xV(i),TSpc)==0     %Tie X-section
        x3= repmat(xV(i),[1 n1x-1-nnt]);    o=o+1;
        y3= Ye;     y3(1:nnt)=[];       %First and Last node# of ith
X-sec
        z3= Ze;     z3(1:nnt)=[];       n1=c+1;
        n2=c+n1x-1-nnt;
        C(n1:n2,:)=[x3;y3;z3]';
        n9= n1 :n1+q-1;             %Node# at the 1st layers of each
X-sec
        L1n(1,1:nnSL(1),i)=[QW(o,:) n9];                c=n2;

    else                        %NON-Tie X-section
        x3=repmat(xV(i),[1 n1x-1]);
        n1=c+1;
        n2=c+n1x-1;                 %First and Last node# of ith X-sec
        C(n1:n2,:)=[x3;Ye;Ze]';
        n9=n1+nnt+q-1;
        L1n(1,1:nnSL(1),i)=n1:n9;   c=n2;
    end
%2nd to (End-1) Layers
    for j=2:length(nnSL)-1    %Store node numbers in layered
sequence/X-sec
        L1n(j,1:nnSL(j),i)= max(n9)+1 :max(n9)+nnSL(j);
        n9=max(n9)+nnSL(j);
    end
%Last Layer
    L1n(end,:,i)= n9+1 :n9+nnSL(end);
end
    % GTE=[ Rail  TIe      R/T    Brick]          [Grand Total
Elements]
    GTE=[(nnr-1) NoT*NTE1  NoT (sum(nnSL(1:end-1))-NSL)*(nnr-1)];
    GTN=c;                          %GRAND TOTAL NODES
    nLN= nnr+ (NTE1+1)*NoT;     %Number of LINE NODES
    nLE=sum(GTE(1:3));          %Number of LINE ELEMENTS

%% *************** NODE - ELEMENT CONNECTIVITY (NECT)
******************
%*****************************************************************
****
% 1-Creats Node-Element Connectivity Table
% 2-Store Type of Elements in "ToE" (1=Rail, 2=Tie, 3=Rail-Tie, 4-
n=Brick)
% 3-Store Length of Elements in "LoE"
% 4-Creats k-Node coordinates for Element X-sec orientation
```

```
% NECT(n1,n2,n3, ...,n8) ...{row# =elem#....col=Node #s }
    LoE= zeros(nLE,1);          %Length of Elements
    ToE=zeros(sum(GTE),1);      %NECT(n1 n2 ..n8)
    NECT=zeros(sum(GTE),8);
    Ck=zeros(3);
    T=zeros(12,12,3);   %K-node Coord |Transformation

% Rial Elements
    NECT(1:GTE(1),1:2)=[1:nnr-1; 2:nnr]';
    ToE(1:GTE(1))=1;
    c=GTE(1);

% TIE Elements
for i = 1:NoT       %Jump on each Tie along the Rail
     ni= nnr+(i-1)*nnt +1  : nnr+(i-1)*nnt +NTE1;
    NECT(c+1:c+NTE1,1:2)= [ni;  ni+1]';
    ToE(c+1:c+NTE1)=2;
    c=c+NTE1;
end

% RAIL-TIE ELEMENTS
    NECT(c+(1:NoT),1:2)=[((1:NoT)-1)*Nr+1;  nnr+((1:NoT)-
1)*nnt+NTEG+1]';
    ToE(c+1:c+GTE(3))=3;
    c=c+GTE(3);

% BRICK ELEMENTS
for i= 1:(nnr-1)                %Run for each X-Section but Last
  for z= 1:length(zV)-1         %Run for each sub-layer (SL)on ith X-
sec
    for y= 1:nnSL(z)-1          %Run for each node on jth SL & ith X-
sec
      c=c+1;
      NECT(c,:)= [L1n(z,y,i) L1n(z,y,i+1) L1n(z,y+1,i+1)
L1n(z,y+1,i)...
              L1n(z+1,y,i) L1n(z+1,y,i+1) L1n(z+1,y+1,i+1)
L1n(z+1,y+1,i)];
    end
  end
end

%Type of Element        Material of Elements [BRICKS]
n6=cumsum(nnSL(1:end-1)-1);
n7=[0 n6(cumsum(LD(4,:)))];                 %Elem no.....
for i= 1:nnr-1          %Run along all X-section along the Rail but
Last
  for z=1:NL            %Run for all main layers
    es= nLE +(i-1)*n7(end) +n7(z) +1;       %Starting  Element
    ef= nLE +(i-1)*n7(end) +n7(z+1);        %Finishing Element
    ToE(es:ef) =3+z;    %Typer of Elements = Material of Elements
  end
end
```

```
% LENGTH of Elemetns    (Rail, Tie & Spring Elements)
LoE(1:nLE) =vecnorm( C(NECT(1:nLE,2),:) -C(NECT(1:nLE,1),:), 2, 2)';

% TRANSFORMATION of COORD from LOCAL to GLOBAL
  Ck(1,:)=[C(1,1) C(1,2)+1 C(1,3)];                  %K-Node for Rail
  Ck(2,:)=[C(nnr+1,1)-1 C(nnr+1+1,2) C(nnr+1,3)];    %K-Node for Tie
  Ck(3,:)=[C(1,1) C(1,2)+1 C(1,3)];                  %K-Node for
Spring
  en=[1  GTE(1)+1  GTE(1)+GTE(2)+1];  %First Element# of [Rail  Tie
Spring]
  for i =1:3
    ni=NECT(en(i),1); nj=NECT(en(i),2);               %Node-i #;
Node-j #
    p1=C(ni,:);     p2=C(nj,:);     p3=Ck(i,:);       %Coord of Node-
i,j,k
    V1=p2-p1;       V21= p3-p1;
    V3= cross(V1,V21);V2= cross(V3,V1);
    e1=V1/norm(V1);
    e2= V2/norm(V2);
    e3= V3/norm(V3);
    t =[e1' e2' e3'];
    T(:,:,i)= kron(eye(4),t);
  end

% Run for each material (Rail, Tie, Spring, Each main layer)
                        %( 1     2    3      4 5 6 7 ...)
Ei= [StrMat(:,1)' 0 LD(5,:)];           %Array of E/Moduus /all
Materials
Er(1:sum(GTE),1)= Ei(ToE(1:sum(GTE)));  %Array of E/Moduus /all
Elements
v=[Pzr(:)' 1 LD(6,:)];                  %Array of P/ratio  /all
Materials
mGS=[0 0 0 LD(10,:)];                   %Array of Gibson Soil Coeff.
end
```

ii. Function ‘GenRandE’

```
function GenRandE
% Generate Random E (Young's Modulus) values for all elements of each
% probabilistic layer

global NL LD L1n NECT nLE C nnr Er
  for j=1:NL                        % Run for All Main Layers
    if LD(11,j)==1                  % If this Main Layer is
Hetrogenious ?
      sLt= sum(LD(4,1:j-1)) +1;     % Top layer of j'th Main Layer
      sLb= sum(LD(4,1:j)) +1;       % Bottom layer of j'th Main Layer
      Ldn=find(L1n(sLb,:,1), 1, 'last' );
```

```
      %A=1st Node#s vector in Layer_j including null/dummy and ....
      A= L1n(sLt:sLb-1, 1:Ldn-1, 1:end-1);  % extreme right edge nodes
      A= permute(A, [3, 2, 1]);
      A=A(:);                               % Realign the matrix with
CMD..

      % Real Challenge here is the segretation (finding indices) of
      % nodes in 'A' beyond slopes (Zero-valued)and Loc-Nod-4 of
extreme
      % right elements
      CD=zeros(1,length(A));                % CD= Null Element Indices
      for k=1:length(A)                     % Elements above Slopes
        ind= find(NECT(nLE+1:end,1) == A(k) );  %Match Node-1 (local)
#s
        %comparing 'A' with the 1st column of NECT removes all Node-4
#s

        if isempty(ind) ==1         %If 'ind' represents Zero(dummy)
node-
            CD(k)= 0;               % & extreme right edge nodes
(Node-4#s)
        else                        %If 'ind' represents 1st nodes of
            CD(k) =1;               %all relevant elements
        end
      end
      % CD contains 0,1 values. 0=indices of extreme right nodes and
      % null/dummy nodes. And 1=indices of valid nodes.

      nulIndx= find(~CD);           % Indices of null Node
      A(A==0)=[];

      [Eln, ~]= find(A'==NECT(nLE+1:end,1));    %Elem#s vector in
Layer_j
      Eln=Eln'+nLE;

      xCnod=C(1:nnr, 1);            %x-Coordinates vector along the
Rail
      yCnod=C(L1n(sLb,1:Ldn,1), 2); %y-Coordinates vector along the
Tie
      zCnod=C(L1n(sLt:sLb,1,1), 3); %z-Coordinates vector along the
Tie

      E = GenRandE_1(j,xCnod,yCnod,zCnod);  % Generate Random 'E'
/Layer_j
      E(nulIndx)=[];                % Remove 'E' values for Null
Elemenets
      Er(Eln)=E;
    end
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function E2 = GenRandE_1(mL,xCnod,yCnod,zCnod)
global LD
% INPUTS:   mL = Main Layer number
%           x,y,zCnod = x-,y-,z-Coordinates of FEM Mesh Nodes
%           x,y,zCrnd = x-,y-,z-Coordinates of Random Mesh
Elem_centers
%           x,y,zElemPos = x-,y-,z-Position of Random Mesh Elements

% OUTPUTS:  E = Young Modulus of 'mL' Layer at all Element Centers

Distr="N";          % "N" = Normal      "LN" = Log_Normal
Mu=LD(5,mL);        % MEAN Young Modulus of Considered Layer (mL)
COV=LD(12,mL);      % Coefficient of Variation
theta=LD(13:15,mL); % Spacial correlation length (x,y,z-dir.s)
Rscal=LD(16:18,mL); % Scaling factor for Random field (FEM/Randscal
=RFEM)

%Generate Random Field COORDINATES
[xCrnd, xElemPos] = GenRandCoord(xCnod, theta(1), Rscal(1));
[yCrnd, yElemPos] = GenRandCoord(yCnod, theta(2), Rscal(2));
[zCrnd, zElemPos] = GenRandCoord(zCnod, theta(3), Rscal(3));
%xCrnd   = Coord. of Random points in x-direction
%xElemPos= Positon (index #) of FEM element center in Random Field

N = [length(xCrnd) length(yCrnd) length(zCrnd)];
% Generate Spacially Correlated Random E values at All Random
Coordinates
% using Covariance Matrix Decomposition method Step-Wise procedure in
3D
E = CMD_SW_3D(N, xCrnd, yCrnd, zCrnd, theta, Distr, Mu, COV);
E2= E(xElemPos, yElemPos, zElemPos);    %Rand-E values @ FEM Elem.
centers

% plottings(E, Mu, COV, prod(N))
% make_video(N, E)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GENERATE RANDOM FIELD COORDINATES
function [Crnd, ElemPos] = GenRandCoord(x, tht, Rscal)
% x= FEM nodal coordinates in x-, y- or z-direcction
Ce= zeros(1,length(x)-1);               %Ce= Random Element Center
coord.
  for i=1:length(x)-1                   %Generate Element center
Coord.s
      Ce(i) = ( x(i)+ x(i+1) )/2;
  end

  Crnd=Ce(1);
  ElemPos=1;
  for i=2:length(Ce)                %Generate Random Field Coordinates
    % Number of random points between 2-Element Centers
```

```
    n= ceil( (Ce(i)-Ce(i-1))/ (tht/Rscal) );
    % Distance of random points between 2-Elements
    d= ( Ce(i) -Ce(i-1) )/n;
    Crnd= [Crnd repelem(d,n)];
    ElemPos= [ElemPos n];
  end
Crnd = cumsum(Crnd);
ElemPos=cumsum(ElemPos);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function E = CMD_SW_3D(N,x,y,z,tha,Distr,Mu,COV)
% Random Field Generation using "Stepwise Covariance Matrix
Decomposition"
% Refernce: Li et. al. 2019
% Discretization of Correlation Function to Generate Correlation
Matrix
Rx=zeros(N(1),N(1));
Ry=zeros(N(2),N(2));
Rz=zeros(N(3),N(3));
    for i=1:N(1)            % Markovian Correlation Function in 3D
        for j=1:N(1)
            rx=abs(x(i)-x(j)) /tha(1);          %'tau /theta' ratio
            Rx(i,j) = exp(-2*sqrt(rx^2));
        end
    end

    for i=1:N(2)            % Markovian Correlation Function in 3D
        for j=1:N(2)
            ry=abs(y(i)-y(j)) /tha(2);          %'tau /theta' ratio
            Ry(i,j) = exp(-2*sqrt(ry^2));
        end
    end

    for i=1:N(3)            % Markovian Correlation Function in 3D
        for j=1:N(3)
            rz=abs(z(i)-z(j)) /tha(3);          %'tau /theta' ratio
            Rz(i,j) = exp(-2*sqrt(rz^2));
        end
    end
Lx=chol(Rx,'lower');        % Lower Triangle of Cholskey Factorization
Ly=chol(Ry,'lower');
Lz=chol(Rz,'lower');

% Stepwise Covariance Matrix Decomposition (Referebce: D.Q. Li, 2019)
    U=randn(N(1), N(2), N(3));

    G= Lx* reshape(U, N(1), N(2)*N(3));
    G= reshape(G, N(1), N(2), N(3));
    G= permute(G, [2, 3, 1]);

    G= Ly* reshape(G, N(2), N(3)*N(1));
```

```
    G= reshape(G, N(2), N(3), N(1));
    G= permute(G, [2, 3, 1]);

    G= Lz* reshape(G, N(3), N(1)*N(2));
    G= reshape(G, N(3), N(1), N(2));
    G= permute(G, [2, 3, 1]);

% Transformation of LN_Distribution to underlying Norm_Distribution
if Distr=="LN"
    SigLN= sqrt( log(1+COV^2));
    MuLN = log(Mu)- 0.5*SigLN;
end
    %Calculate the Spatially Correlated Random E values based on
    %their Distribution Type
    if      Distr=="LN";    E= exp(MuLN+ SigLN *G);
    elseif  Distr=="N";     E= Mu + COV*Mu*G;
    end
% RESULTS & PLOTS
% disp("Actual  vs  Given -> Means");     disp([mean(E(:)) Mu]);
% disp("Actual  vs  Given -> COV's");     disp([std(E(:))/mean(E(:))
COV]);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function plottings(E,Mu,COV,n)
    figure(1)
    plot(E(:));         hold on;        ylim([Mu-6*COV*Mu
Mu+6*COV*Mu]);
        set(gca,{'YDir'},{'reverse'});
        set(gca,{'XAxisLocation'},{'top'});
    plot([0 n],[Mu Mu]);   hold off;    grid on;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function make_video(N,E)                % Create the Video
    for k = 1: N(3)
        figure(2)
        imagesc(E(:,:,k));              %colormap gray
        caxis([min(E(:)) max(E(:))]);
        colorbar('Direction','reverse');
        set(gca,{'XAxisLocation'},{'top'});
        F(k) = getframe(gcf) ;          drawnow
    end
    writerObj = VideoWriter('Spatial_Variation_Video.avi');
    writerObj.FrameRate = 10;           % set the frames per second
    open(writerObj);                    % open the video writer
    for k=1:length(F)                   % write the frames to the
video
        frame = F(k) ;                  % convert the image to a frame
        writeVideo(writerObj, frame);
    end
    close(writerObj);                % close the writer object
```

```
end
```

iii. Function 'StifMat'

```
% Creates all types of Stiffness Matrices [Local, Global, Overall
Global]
function StifMat
global GSM GTN GTE C NECT ToE StrMat T nLN RTK LoE Er v mGS
nG= nLN*6+(GTN-nLN)*3;
GSM=zeros(nG,nG);       %Size of Global Stif matrx(GSM)

%% Create Global Stiffness matrix (GSM) & Material Stiffness (E)
matrix
c=0;
for i = 1 : sum(GTE)        %RUN FOR EACH ELEMENT
  if ToE(i) <4              %LINE ELEMENTS ONLY
    Kloc= KLine(i);                             %Local Stiffness
Matrix
    Kglo= T(:,:,ToE(i))* Kloc *T(:,:,ToE(i))';  %Global Stiffness
Matrix

    n1= NECT(i,1);          %1st node of elem i
    n2= NECT(i,2);          %2nd node of elem i
    el= [(n1*6-5):(n1*6) (n2*6-5):(n2*6)];      %Indeces for placing
'K'
    GSM(el,el)= GSM(el,el)+ Kglo;

  else                      %BRICK ELEMENTS ONLY
    c=c+1;
    e=zeros(1,24);
    n8= NECT(i,:);
    Ei = GenEmat(i,n8);                 %With GIBSON SOIL
    C8= [C(n8,1) C(n8,2) C(n8,3)];      %Coord of all 8 nodes
    Kbe = K_Brick(C8,Ei);               %Local Stiffness matrix

    for j=1:8               %Make arrangement for placing K into GSM
      if n8(j)<=nLN
          em= n8(j)*6-5 :n8(j)*6-3;
      else
          em= nLN*6+ ((n8(j)-nLN)*3-2 :(n8(j)- nLN)*3);
      end
      e(1, j*3-2:j*3)= em;              %Seeding positions
    end
    GSM(e,e)= GSM(e,e)+ Kbe;            %Global Stiffness matrix
  end
end

%%
**********************************************************************
% Generate Stress-Strain (E) Matrix incorporating Gibson Soil Coeff.
```

```
  function Ej = GenEmat(j,n8)
     Erj= Er(j) +mGS(ToE(j))*mean(C(n8,3));  %E= Eo +m*z (GIBSON SOIL)
    u = v(ToE(j));
    Ej= Erj/((1+u)*(1-2*u))* [[1-u u u; u 1-u u; u u 1-u] zeros(3);
                                          zeros(3) (1/2-u)*eye(3)];
  end
%%
*********************************************************************
%Create Stiffness Matrix for LINE elements in LOCAL Coordinates
  function Kl = KLine(k)
    if ToE(k) ==3           %Rail-Tie Spring Elements
      AS=RTK;
      [TS, ay, by, cy, dy, az, bz, cz, dz] = deal(0);
    else                    %For both Rail (ToE=1) & Tie (ToE=2)
Elements
      t=ToE(k);             l=LoE(k);
      Ev=StrMat(t,1);       A=StrMat(t,2);
      G=StrMat(t,3);        J=StrMat(t,4);
      Iy=StrMat(t,5);       Iz=StrMat(t,6);
      AS= Ev*A/l;           TS = G*J/l;
      ay=12*Ev*Iy/l^3;      by=6*Ev*Iy/l^2; cy=4*Ev*Iy/l;
dy=2*Ev*Iy/l;
      az=12*Ev*Iz/l^3;      bz=6*Ev*Iz/l^2; cz=4*Ev*Iz/l;
dz=2*Ev*Iz/l;
    end
    m1=[AS 0 0; 0 az 0; 0 0 ay];  m2=[0 0 0;  0 0 bz; 0 -by 0];
m3=m2';
    m4=[TS 0 0; 0 cy 0; 0 0 cz];  m5=[-TS 0 0;0 dy 0; 0 0  dz];
    Kl= [m1 m2 -m1 m2;  m3 m4 -m3 m5;  -m1 -m2 m1 -m2;  m3 m5 -m3 m4];
  end
%%
*********************************************************************
%Create Stiffness Matrix for BRICK elements in
LOCAL(+Global)Coordinates
  function Ke = K_Brick(C8,E)
    ng=2;               % ng=# of Gauss Intergration Points
    w=ones(3,8);        % weight of Gauss Integ. Points
    Ke=zeros(24);       % Stiffness matrix
    Bb=zeros(6,24);     % Strain-Displacement matrix
    Cn=[-1 -1 -1; 1 -1 -1; 1 1 -1; -1 1 -1;...
         -1 -1  1; 1 -1  1; 1 1  1; -1 1  1]'*sqrt(1/3);
    for h= 1:ng^3
                    L=Cn(1,h);      m=Cn(2,h);      n=Cn(3,h);
      dNL=[-(1-m)*(1-n)  (1-m)*(1-n)  (1+m)*(1-n) -(1+m)*(1-n)...
           -(1-m)*(1+n)  (1-m)*(1+n)  (1+m)*(1+n) -(1+m)*(1+n)]/8;

      dNm=[-(1-L)*(1-n) -(1+L)*(1-n)  (1+L)*(1-n)  (1-L)*(1-n)...
           -(1-L)*(1+n) -(1+L)*(1+n)  (1+L)*(1+n)  (1-L)*(1+n)]/8;

      dNn=[-(1-L)*(1-m) -(1+L)*(1-m) -(1+L)*(1+m) -(1-L)*(1+m)...
            (1-L)*(1-m)  (1+L)*(1-m)  (1+L)*(1+m)  (1-L)*(1+m)]/8;
```

```
      J= [dNL; dNm; dNn] *C8;
      B3= J\[dNL; dNm; dNn];        % B3(3x8) = inv(J) *Diff(N,Lmn)

        for t=1:8                   % Construct Bb(6x24) from B3(3x8)
         Bb(:,t*3-2:t*3)=  [B3(1,t)     0          0      ;
                            0           B3(2,t)    0      ;
                            0           0          B3(3,t);
                            B3(2,t)     B3(1,t)    0      ;
                            0           B3(3,t)    B3(2,t);
                            B3(3,t)     0          B3(1,t) ];
        end
      we=w(1,h)*w(2,h)*w(3,h);
      Kb= we* Bb'*(E*Bb) *det(J);
      Ke=Ke+Kb;            %Cumulative over Guass Intg Points
    end
  end
end
```

iv. Function 'BCond'

```
%Create Force & Disp vectors for 6-DOF (Line) & 3-DOF (Brick) Elements
%Force vector = External/Applied Forces
%Displacement vector (Nodal Movements allowed?  Yes=1,  No=0)
function BCond
global C D Fex GTN nLN TnoL Nr WLd
cmax=[max(C(:,1)) max(C(:,2)) max(C(:,3))];     %Coord. of farthest
point
nG=nLN*6+(GTN-nLN)*3;
D= ones(nG,1);          %Disp. vector
Fex=zeros(nG,1);        %Exteral Force vectrs
%% DISPLACEMENT CONSTRAINTS (Ux,Uy,Uz) on Sp. Nodal Planes(GLOBAL
COORD)
  for i= 1:GTN
%FRONT+BACK planes    (YZ-plane) - Nodes at x=0  OR  x=max
    if C(i,1)==0 || C(i,1)==cmax(1) %First and Last Tie Locations
      if i<=nLN;    nx=6*i-5;                       %Line node
      else;         nx=nLN*6+(i-nLN)*3-2;           %Brick Node
      end
      D(nx)=0;      %Ux =0
    end

%CENTER LINE plane  (XZ-plane) - Nodes at y=0
    if C(i,2)==0
      if i<=nLN;    ny=6*i-4;                       %Line node
      else;         ny=nLN*6+(i-nLN)*3-1;           %Brick Node
      end
      D(ny)=0;      % Uy =0
    end

%FAR-LATERAL plane  (XZ-plane) - Nodes at y=max
```

```
    if C(i,2)==cmax(2)
      if i<=nLN;    nxyz= 6*i+(-5:-3);              %Line node
      else;         nxyz= nLN*6 +(i-nLN)*3 +(-2:0); %Brick Node
      end
      D(nxyz)=0;    %[Ux Uy Uz] =0
    end

%BOTTOM Plane       (XY-Plane) - Nodes at z=max
    if C(i,3)==cmax(3)
        nxyz =nLN*6 +(i-nLN)*3 +(-2:0);
        D(nxyz)=0;  %[Ux Uy Uz] =0
    end
  end

%% External Force Vector (Wheel Loads)
    tx1=(TnoL-0-1)*Nr +1;               %Node# below 1st Wheel Load
%     tx2=(TnoL+2-1)*Nr +1;               %Node# below 2nd Wheel Load
    Fex(6*tx1-3)= 1.0*WLd;
%     Fex(6*tx2-3)= 1.0*WLd;
end
```

v. Function 'Solve'

```
%Solve for Nodal Displacmnts & Forces (Reactions)in GLOBAL coordinates
%Apply Global and Local Equilibrium Checks

function Solve
  global D Fex GSM
  d1=find(D);           %Nodes with NON-ZERO Displacements
  d0=find(~D);          %Nodes with ZERO Displacements (B/Cond)
  F1=Fex;
  GSM(d0,:)=[];
  GSM(:,d0)=[];         %Condense GSM for zero displacements
  F1(d0) =[];           %Condense F   for zero displacements

  D(d1) =GSM\F1;        %D =inv(K) *F
  if sum(find(isnan(D)))>0
    'Pinv in progress'
    D(d1)= pinv(GSM)*F1;
  end
  StifMat;              %Recalculate GSM to save space of memory
  F =GSM*D;             %Nodal Force vector
  Fex(d0)=F(d0);        %External Force vector (External + Reactions)

% *************************
  Reshape;              %Reshape 'D' &  'F' vectors for future
processing
%   TrackModulus(max(D))  %Calculate Track Modulus
%   Equilibrium_Checks(1) %Check Global and Local Equilbrum @Specific
Node
```

```
                        % Node# must be a RAIL Node { <= GTE(1)+2 }
end


%%
**********************************************************************
function Reshape                    %Reshape 'D' & 'F' vectors
global nLN D GTN Fex D3 Fex3 D6l
    g1=nLN*6;   g2=numel(D);
    D6l=reshape(D(1:g1),[6,nLN])';
    Db=reshape(D(g1+1:g2),[3,GTN-nLN])';
    D3= vertcat(D6l(:,1:3),Db);

    Fex6=reshape(Fex(1:g1),[6,nLN])';
    Fexb=reshape(Fex(g1+1:g2),[3,GTN-nLN])';
    Fex3 =vertcat(Fex6(:,1:3),Fexb);
end

function TrackModulus(d)            %Calculate Track Modulus
global WLd StrMat
  Kd= WLd/d;
  E= StrMat(1,1);
  I= StrMat(1,5);
  K= (Kd^(4/3))/(64*E*I)^(1/3);
  strcat('Track Modulus = ',num2str(K/1000,'%5.1f'),' MPa',' = ', ...
      num2str(K/6.895,'%6.0f'),' psi')
end

function Equilibrium_Checks(nod)    %Check If Equilibrium holds ???
global C NECT ToE T GTE LoE StrMat RTK Fex3 D6l

%Check GLOBAL EQUILIBRIUM
  Sum_F_Glo= sum(Fex3);                 %Sum(Fx,Fy,Fz)~=0
  Sum_M_Glo= sum(Fex3(:,3).*C(:,1));    %About N1 by Fz

%Check LCOAL EQUILIBRIUM at Node "nod"     nod <=#/Rail
Elements{GTE(1)+1}
    Fz=0;
    My=0;
    [r,~]=find(NECT==nod);      %Find all Element #s connected with
'nod'
    for i=1:numel(r)
        e=r(i);                 %Line Element # in question
        a=NECT(e,1);            %1st Node of Element 'e'
        b=NECT(e,2);            %2nd - - - - - - - - - -
        f= KLine(e)* (T(:,:,ToE(e))'*[D6l(a,:) D6l(b,:)]');    %Calc
F_loc

        if e<=GTE(1) && a==nod;     k=3;                l=5;
        elseif e>GTE(1);            k=1;                l=5;
        else;                       k=9;                l=11;
        end;                        Fz=Fz+f(k);         My=My+f(l);
```

```
    end
    strcat('Equilibrium Check___', ...          %Chk=[Fz My Fz2 My2]
@node2
      num2str([Sum_F_Glo(3)  Sum_M_Glo  Fz  My],'%12.2E'))
%   [Sum_F_Glo(3)  Sum_M_Glo  Fz  My]
%% Create 'K' Matrix for LINE elements in LOCAL Coordinates
    function Kl = KLine(k)
        if ToE(k) ==3           %Rail-Tie Spring Elements
          AS=RTK;
          [TS, ay, by, cy, dy, az, bz, cz, dz] = deal(0);
        else                    %For both Rail (ToE=1) & Tie (ToE=2)
Elements
          t=ToE(k);          y=LoE(k);
          Ev=StrMat(t,1);    A=StrMat(t,2);
          G=StrMat(t,3);     J=StrMat(t,4);
          Iy=StrMat(t,5);    Iz=StrMat(t,6);
          AS= Ev*A/y;        TS = G*J/y;
          ay=12*Ev*Iy/y^3;  by=6*Ev*Iy/y^2; cy=4*Ev*Iy/y;
dy=2*Ev*Iy/y;
          az=12*Ev*Iz/y^3;  bz=6*Ev*Iz/y^2; cz=4*Ev*Iz/y;
dz=2*Ev*Iz/y;
        end
    m1=[AS 0 0; 0 az 0; 0 0 ay];  m2=[0 0 0;  0 0 bz; 0 -by 0];
m3=m2';
    m4=[TS 0 0; 0 cy 0; 0 0 cz];  m5=[-TS 0 0;0 dy 0; 0 0  dz];
    Kl= [m1 m2 -m1 m2;  m3 m4 -m3 m5;  -m1 -m2 m1 -m2;  m3 m5 -m3 m4];
    end
end
```

vi. Function 'ChkNonLin'

```
%1)Check the Non-Linearity (Stress dependent) of Brick Elements
%2)Repeat the calculations until Non-Linearity holds within Tolerance
function ChkNonLin
global ToE Er Tol GTE LD
c=1;
Erp = RevisedYmod;
while max(abs(Erp-Er)) > Tol
%     disp('Repetition cycle #')
    c=c+1;
    StifMat;
    BCond;
    Solve;
    Erp = RevisedYmod;
end

%*********************************************************************
    function Erl = RevisedYmod              %Calc Revised Young's
Modulus
```

```
      Erl = Er;                                  %Store Er for comparison
later
      Pa=101.325;                                %Atmospheric Pressure in
kPa
      for j= sum(GTE(1:3))+1 :GTE(4)             %For Brick elements only
          if LD(7, ToE(j)-3)== 1                 %If jth Element is Non-
Linear ?
              Sij= Stress1(j);
              Soct= abs(sum(sum(Sij(:,1:3)/3)));
              K34= LD(8:9, ToE(j)-3);            %Recall Non-Linear Coeff.
K3,K4
              Er(j)= K34(1)*Pa*(Soct/Pa)^K34(2);
          end
      end
    end

end
```

vii. Function 'Stress1'

```
function Sige = Stress1(b)      %Calc. Stresses for given Brick elem
'b'
    global Er v ToE NECT C D3 mGS
    Sige=zeros(8,6);
    n8= NECT(b,:);                    %All 8 Node #s of Elemenet 'b'
    C8= [C(n8,1) C(n8,2) C(n8,3)];  %Corner Coord. wrt Origin
    Cnm=[-1 -1 -1; 1 -1 -1; 1 1 -1; -1 1 -1;...
         -1 -1  1; 1 -1  1; 1 1  1; -1 1  1]'*sqrt(1/3);
    d=reshape(D3(NECT(b,:),:)', [24,1]);    %Deformation vector (24x1)
    Ei = GenEmat(b);                        %With GIBSON SOIL
  for j= 1:8
                         L= Cnm(1,j);    m= Cnm(2,j);   n= Cnm(3,j);
    dNL=[-(1-m)*(1-n)  (1-m)*(1-n)  (1+m)*(1-n) -(1+m)*(1-n) ...
         -(1-m)*(1+n)  (1-m)*(1+n)  (1+m)*(1+n) -(1+m)*(1+n)]/8;

    dNm=[-(1-L)*(1-n) -(1+L)*(1-n)  (1+L)*(1-n)  (1-L)*(1-n) ...
         -(1-L)*(1+n) -(1+L)*(1+n)  (1+L)*(1+n)  (1-L)*(1+n)]/8;

    dNn=[-(1-L)*(1-m) -(1+L)*(1-m) -(1+L)*(1+m) -(1-L)*(1+m)...
          (1-L)*(1-m)  (1+L)*(1-m)  (1+L)*(1+m)  (1-L)*(1+m)]/8;

    J= [dNL; dNm; dNn] *C8;
    B3=J\[dNL; dNm; dNn];           %B3(3x8)=inv(J)*Diff(N,Lmn)
    for k=1:8
      Be(:,k*3-2:k*3)=  [B3(1,k)     0           0      ;
                         0           B3(2,k)     0      ;
                         0           0           B3(3,k);
                         B3(2,k)     B3(1,k)     0      ;
                         0           B3(3,k)     B3(2,k);
```

```
                          B3(3,k)     0           B3(1,k) ];
%Be(6x24)
    end
    strn= (Be*d);               %e = (exx, eyy, ezz, exy, eyz, ezx)
    Sige(j,:) = Ei *strn;       %sig= E*strn = E*B*d
  end

% Generate Stress-Strain (E) Matrix incorporating Gibson Soil Coeff.
  function Ej = GenEmat(j)
     Erj= Er(j) +mGS(ToE(j))*mean(C(n8,3));  %E= Eo +m*z (GIBSON SOIL)
    u = v(ToE(j));
    Ej= Erj/((1+u)*(1-2*u))* [[1-u u u; u 1-u u; u u 1-u] zeros(3);
                                              zeros(3) (1/2-u)*eye(3)];
  end
end
```

viii. Function 'Plots'

```
function Plots                  % Plot Deformation Graphs
% If its Deterministic Analysis, Plot Displacement Graphs
% Else PROCEED to plot Risk and Reliability Plots
global AnType

% Plot_Disp                       %Plot Displacements along DEPTH, TIE
& RAIL
% Plot_Stress                     %Plot Stresses along DEPTH, TIE &
RAIL
if AnType==1;   Plot_Risk; end  %Plot Risk & Reliability Graph

end


%%
**********************************************************************
function Plot_Disp      % Plot Displacements along DEPTH, TIE & RAIL
global C D3 L1n LD NTE1 NTEG NL nnr TnoL
    xn= TnoL;           %Select Cross Section (default =1 =Loaded Xn)
    Ly= NTEG+1;         %Location of Uz~Depth plot [1=CL,
NTEG+1=Wheel]
    nLyer= LD(4,:);
    nLyer= [1 nLyer(1:end-1)];
     nLyer= cumsum(nLyer);           %Layer #s @top of ALL Main Layers

%% Along the DEPTH
     NNPDaDpth =L1n(:, Ly, xn)      %Node numbers @selected
Xn/Location
    x2= D3(NNPDaDpth,3);           %Uz at selected nodes
    y2= C(NNPDaDpth,3);            %Cz at selected nodes
figure(52);
    plot(x2*1000,y2,'-*b');
```

```
    title('Vertical Deformation Below WHEEL');
    xlabel('Vertical Deformation (U_{z} / mm)');
    ylabel('Depth below WHEEL (m)');        grid 'on';
    set(gca,{'YDir'},{'reverse'});
    set(gca,{'XAxisLocation'},{'top'});

%% Along the TIES
    NNPDaTie =L1n(nLyer, 1:NTE1+1, xn);     %Node #s @selected Xn
    x1= C(NNPDaTie(1,:),2);              %Cy at selected nodes
    y1= D3(NNPDaTie(:,:),3);           %Uz at selected nodes
    y1=reshape(y1, [NL,NTE1+1]);
figure(51);
    plot(x1,y1*1000,'-*');
    %ylim([0 inf]);
    title('U_{z} / mm along Loaded Tie');
    xlabel('Distance from Center of Tie (m)');
    ylabel('Vertical Deformation (U_{z} /mm)');
    set(gca,{'YDir'},{'reverse'});          grid 'on';
    legend('Top of Ballast','Top of Subballast', 'Top of Subgrade'...
        , 'Top of Natural Soil','location','best')

%% Along the RAIL
    nnR= L1n(nLyer, NTEG+1, :);         %Node #s @top of ALL Main
Layers
        x3=C(1:nnr,1)';
        y32=reshape(D3(nnR,3),[NL,nnr]);
        y3=D3(1:nnr,3)';               %Without flipping the data
%       x3= [-fliplr(x3) x3];y32=[fliplr(y32) y32];y3=[fliplr(y3) y3];
figure(53);
    plot(x3,y3*1000,'-*', x3,y32*1000,'-*');
    %ylim([0 inf]);
    title('Deformation along the Rail-Uz (mm)');
    xlabel('Distance along the RAIL(m)');
    ylabel('Vertical Deformation (U_{z} / mm)');
    set(gca,{'YDir'},{'reverse'});          grid 'on';
    legend('Top of Rail','Top of Ballast','Top of Subballast',...
           'Top of Subgrade', 'Top of Natural Soil','location','best')
end

%%
*********************************************************************
function Plot_Stress    % Plot Stresses along DEPTH, TIE & RAIL
global C NSL L1n NTEG TnoL nLE NECT NL NTE1 LD nnr MdSz StrAna
if StrAna==0;   return;    end         % Return if Stresses Are NOT
Required

%% BELOW WHEEL          (VERTICLE STRESS)
    VSW=zeros(1,NSL);
    NnW =L1n(1:end, NTEG+1, TnoL);  %Node#s below Wheel(except last)
    zc=zeros(1,length(NnW)-1);
    for i = 1 : length(NnW)-1
      zc(i)= (C(NnW(i),3)+C(NnW(i+1),3))/2;
```

```
      [r,c]= find(NECT(nLE+1:end,1:4) ==NnW(i));%[Elem#, Loc/Node#]
      Sav=zeros(4,1);
      % Run 2-times for Quarter-sized model & 4-times for Half-sized
Model
      for ii=1:8/MdSz
        Sij= Stress1(r(ii)+nLE);                %Calc Stress matrix
        Sav(ii)= mean(Sij([c(ii) c(ii)+4],3));
%mean(Stress(LocNode#s)
      end
      VSW(i)=mean(Sav);             %Mean/Ver/Stress @middle of
SubLayers
    end
    Plot_Str(-VSW,zc,1,'Vertical'); %[X, Y, Wheel, Verticle/Deviator]

%% ALONG TIE            (VERTICLE STRESS)
    %Due to symmetry, Stress is calculated only one side of the tie
    VST = zeros(NL,NTE1);
    Lns= cumsum([1 LD(4,:)]);
    Lns=Lns(1:end-1);               %SubLayer#s @top of Main Layer
    NnT =L1n(Lns, 1:NTE1+1, TnoL);  %Node#s along Loaded Tie@top
/MainLyr
    yc= (C(NnT(1,1:end-1),2) + C(NnT(1,2:end),2) )/2;
    for i=1:NL
      for j=1:NTE1
        et= find(NECT(nLE+1:end,1)==NnT(i,j))+nLE;  %Elem#
        Sij= Stress1(et);
        VST(i,j)= mean(Sij([2 3], 3));      %Mean(Stress(LocNode#s)
      end
    end
    Plot_Str(yc,-VST,2,'Vertical'); %[X, Y, Wheel, Verticle/Deviator]

%% ALONG RAIL           (VERTICLE STRESS)
    %Stress is calculated on both sides of the Rail and then averaged
    VSR = zeros(NL,nnr-1);
    NnR(:,:) =L1n(Lns, NTEG+1, :);  %Node#s along Rail@top /MainLyr
    xc= (C(NnR(1,1:end-1),1) + C(NnR(1,2:end),1) )/2;
    for i=1:NL
      for j=1:nnr-1
        ei=find(NECT(nLE+1:end,4)==NnR(i,j))+nLE;  %Elem# inside of
Rail
        eo=find(NECT(nLE+1:end,1)==NnR(i,j))+nLE;  %Elem# outside of
Rail
        Si= Stress1(ei);    Savi=mean(Si([3 4], 3));
        So= Stress1(eo);    Savo=mean(So([1 2], 3));
        VSR(i,j)= (Savi+Savo)/2;
%Mean(Stress(LocNode#s)
      end
    end
    Plot_Str(xc,-VSR,3,'Vertical'); %[X, Y, Wheel, Verticle/Deviator]
end
%*********************************************************************
function Plot_Str(X,Y,dir,VD)   %Plot Stresses along DEPTH, TIE & RAIL
```

```
%ALONG THE DEPTH (BELOW THE WHEEL)
    if dir == 1
    figure(62);
      plot(X,Y,'-b*');
      set(gca,{'YDir'},{'reverse'});  grid on
      set(gca,{'XAxisLocation'},{'top'});
      xlabel(strcat(VD,' Stress (kPa)'))
      ylabel('Depth below the Rail (m)');

%ALONG THE TIE
    elseif dir == 2
    figure(64);
      plot(X,Y,'-*');
      xlabel('Distance from Center of Tie (m)');
      ylabel(strcat(VD,' Stress (kPa)'));   grid on;
      legend('Top of Ballast','Top of Subballast', 'Top of
Subgrade'...
        , 'Top of Natural Soil','location','best')

%ALONG THE RAIL
    elseif dir == 3
    figure(66);
      plot(X,Y,'-o');
      set(gca,{'YDir'},{'reverse'});
      xlabel('Distance along the RAIL (m)');
      ylabel(strcat(VD,' Stress (kPa)'));   grid on;
      legend('Top of Ballast','Top of Subballast', 'Top of
Subgrade'...
        , 'Top of Natural Soil','location','best')
    end
end

%%
**********************************************************************
function Plot_Risk      %Plot Risk & Reliability Graph
global Uzbl
D=Uzbl;
Ddt=5.626;

    xs=min(D): 0.001 :max(D);
%   [max(D) min(D) mean(log(D)) std(log(D))]
%   [max(D) min(D) mean(D)      std(D)]

%     yp=normpdf(xs, mean(D), std(D));            %Generate Equiv. N-
distr.
%     yc=normcdf(xs, mean(D), std(D));            %Generate Cummu. N-
distr.
%     uz10= norminv(0.90,mean(D),std(D))          %Uz @ 10% Pf
%      Pfx=1-normcdf(Ddt,mean(D),std(D))          %Pf @ given
settlement

  yp=lognpdf(xs, mean(log(D)), std(log(D)));  %Equivalent  Log_N-dist.
```

```
  yc=logncdf(xs, mean(log(D)), std(log(D)));  %Cummulative Log_N-dist.
  uz10= logninv(0.9, mean(log(D)),std(log(D)));%Uz @ 10% Pf
  Pfx=1-logncdf(Ddt,mean(log(D)),std(log(D)));%Pf @ given settlement

  %   disp([uz10 Pfx mean(log(D)) std(log(D))]');
  disp([uz10 Pfx mean(D) std(D)   std(D)/mean(D)*100]);

  figure(201);               %Compare Actual Results ~ Equ. N-Distr.
  hist(D,50);
  set(gca,'YTick', [])
  hold on;            grid on
  ax1=gca;            ax1p=ax1.Position;
  ax2=axes('Position',ax1p,'YAxisLocation','right','Color','none');
  ax3=axes('Position',ax1p,'YAxisLocation','left','Color','none');
  line(xs, yc,'Parent',ax2,'Color','r','LineWidth',2.5);  grid on
  line(xs, yp,'Parent',ax3,'Color','m','LineWidth',2.5);
  % title('Distribution of Maximum Settlement (mm)', ...
  %                   'FontName','Times New Romen');
  ylabel(ax2,'Cummulative Density Function (CDF)');
  ylabel(ax3,'Probability Density Function (PDF)');
  xlabel('Max. Settlement (mm)');
  % legend('PDF (Lognormal)', 'CDF (Lognormal)')
  set(ax2,'YColor','r');
  set(ax3,'YColor','m');

%  figure(202);                   %Plot Actual Results ~ Cumm. N-
Distr.
%  hist(D,20);
%  hold on;            grid on
%  ax1=gca;            ax1p=ax1.Position;
%  ax2=axes('Position',ax1p,'YAxisLocation','right','Color','none');
%  line(xs, yc,'Parent',ax2,'Color','r');
end
```

ix. Function 'OutputPrint'

```
function OutputPrint(i, id)
global L1n NTEG TnoL D3 LD NSL C NECT nLE MdSz WLd StrMat Nitr
global Uzrl Uzbl Uzsb Uzsg Szbl Szsb Szsg Tmod Set1

if id==1                % Define the variables for Outputs
    Uzrl= zeros(Nitr,1); % Settlement @top of Rail in each iteration
    Uzbl= zeros(Nitr,1); % Settlement @top of Ballast in each
iteration
    Uzsb= zeros(Nitr,1); % Settlement @top of Subballast in each
iteration
    Uzsg= zeros(Nitr,1); % Settlement @top of Subgrade in each
iteration
    Szbl= zeros(Nitr,1); % Ver.Stress @top of Ballast in each
iteration
```

```
    Szsb= zeros(Nitr,1); % Ver.Stress @top of Subballast in each
iteration
    Szsg= zeros(Nitr,1); % Ver.Stress @top of Subgrade in each
iteration
    Tmod= zeros(Nitr,1); % Track Modulus in each iteration
end

if id==2
    NnW =L1n(:, NTEG+1, TnoL);  % Node numbers below Wheel along Depth
    Uzrl(i)= D3(NnW(1), 3)*1000;
    Uzbl(i)= D3(NnW(2), 3)*1000;
    Uzsb(i)= D3(NnW(2+sum(LD(4,1:1))), 3)*1000;
    Uzsg(i)= D3(NnW(2+sum(LD(4,1:2))), 3)*1000;

    VSW=zeros(NSL,1);
    zc=zeros(1,length(NnW)-1);
    for ie = 1 : length(NnW)-1
      zc(ie)= (C(NnW(ie),3)+C(NnW(ie+1),3))/2;
      [r,c]= find(NECT(nLE+1:end,1:4) ==NnW(ie));%[Elem#, Loc/Node#]
      Sav=zeros(4,1);
   % Run 2-times for Quarter-sized model & 4-times for Half-sized
Model
     for ii=1:8/MdSz
         Sij= Stress1(r(ii)+nLE);                %Calc Stress matrix
         Sav(ii)= mean(Sij([c(ii) c(ii)+4],3));
%mean(Stress(LocNode#s)
     end
     VSW(ie)=mean(Sav);          %Mean/Ver/Stress @middle of SubLayers
    end
    Szbl(i)= -VSW(1);
    Szsb(i)= -VSW(1+sum(LD(4,1:1)));
    Szsg(i)= -VSW(1+sum(LD(4,1:2)));

    Kd= WLd/(Uzrl(i)/1000);
    EI= StrMat(1,1) * StrMat(1,5);
    Tmod(i)= ( (Kd^4) /(64*EI ) )^(1/3);
end

if id==3
    %Delete Outliers (-ve displacements and extreme values)(Rarely
happens)
    Uzrl(Uzrl<=0)=[];   Uzrl(abs(Uzrl)> 2*mean(Uzrl) )=mean(Uzrl);
    Uzbl(Uzbl<=0)=[];   Uzbl(abs(Uzbl)> 2*mean(Uzbl) )=mean(Uzbl);
    Uzsb(Uzsb<=0)=[];   Uzsb(abs(Uzsb)> 2*mean(Uzsb) )=mean(Uzsb);
    Uzsg(Uzsg<=0)=[];   Uzsg(abs(Uzsg)> 2*mean(Uzsg) )=mean(Uzsg);

     Szbl(abs(Szbl)> 2*mean(Szbl) )=mean(Szbl);
    Szsb(abs(Szsb)> 2*mean(Szsb) )=mean(Szsb);
    Szsg(abs(Szsg)> 2*mean(Szsg) )=mean(Szsg);
    Tmod(abs(Tmod)> 2*mean(Tmod) )=mean(Tmod);

    Set1=[max(Uzrl) min(Uzrl) mean(Uzrl) std(Uzrl)/mean(Uzrl) ...
```

```
      max(Uzbl) min(Uzbl) mean(Uzbl) std(Uzbl)/mean(Uzbl) ...
      max(Uzsb) min(Uzsb) mean(Uzsb) std(Uzsb)/mean(Uzsb) ...
      max(Uzsg) min(Uzsg) mean(Uzsg) std(Uzsg)/mean(Uzsg) ...
      max(Szbl) min(Szbl) mean(Szbl) std(Szbl)/mean(Szbl) ...
      max(Szsb) min(Szsb) mean(Szsb) std(Szsb)/mean(Szsb) ...
      max(Szsg) min(Szsg) mean(Szsg) std(Szsg)/mean(Szsg) ...
      max(Tmod) min(Tmod) mean(Tmod) std(Tmod)/mean(Tmod) ]';

%     disp(Set1);
end

end
```

APPENDIX B. USER GUIDE OF ADYTRACK MODEL

This appendix describes in detail the variables, procedures and schemes employed in ADYTrack program, which is a computer code to perform the structural analysis of railroad trackbed using Stochastic Finite Element method (SFEM). The program is written in MathWorks's MATLAB language using academic licensed version. The main program is named "ADYTrack_4.m" from where various functions are called. Four (4) in the file names represents the version number of the program.

There are some Nested Functions that are called from within the Parent functions and therefore described under the heads of those Parent functions. The reasons of using Nested functions include the limited use, generally once only, of those functions, ease of calling variables (local and global) across the Parent functions and improving efficiency of the overall program. Also note that length, longitudinal and x-axis mean the same dimension/direction, and traverse, width and y-axis mean the same dimension/direction, and vertical, depth, height and z-axis mean the same dimension/direction.

First, the important variables are described in detail, including their purpose, nature, size and other relevant information. Then a detailed explanation is followed for all the functions used in the program including their purpose, and details of schemes and procedures employed.

B.1. Important Variables

In order to smoothly understand the code of ADYTrack, the description of important variables is presented is presented herein.

AnType: It stores the decision of whether to perform Deterministic or Probabilistic analysis. It is scaler quantity with 0 or 1 value. Zero (0) means perform the Deterministic analysis whereas one (1) means perform the Probabilistic analysis.

C: It stores the coordinates of the nodes. It is a 2-D matrix with three columns, for x-, y- and z-coordinates of the nodes, and number of rows equals the total number of nodes in the model. The coordinates are stored such that the row numbers exactly match with the node numbers.

D: It is a displacement vector for the whole system. Its size is equal to the total degree of freedom of the system [u_{x1}, u_{y1}, u_{z1}, θ_{x1}, θ_{y1}, θ_{z1}, u_{x2}, u_{y2}, u_{z2}, θ_{x2}, θ_{y2}, θ_{z2}, ……]. As such, this variable is difficult to read and process, therefore another variable **D3** and **D6** are extracted using 'reshape' function to read and plot the displacements at critical planes and lines.

D3: It represents the displacements of all the nodes in all three dimensions. It is 2-D matrix, the rows of which represents the node numbers and columns store the nodal displacements in all 3 direction (u_x, u_y, u_z).

D6: It represents the displacements of all the line nodes in all six degrees of freedom. It is 2-D matrix, the rows of which represents the node numbers and columns store the nodal displacements and rotations in all 3 direction (u_x, u_y, u_z, θ_x, θ_y, θ_z).

Er: It is an array of Young's modulus values for all the elements in the system. Its size is same as the sum of GTE vector.

F: It is a force vector for the whole system. Its size is equal to the total degree of freedom of the system [f_{x1}, f_{y1}, f_{z1}, M_{x1}, M_{y1}, M_{z1}, f_{x2}, f_{y2}, f_{z2}, M_{x2}, M_{y2}, M_{z2}, ……]. As such, this variable is difficult to read and process, therefore another variable F3 and F6 are extracted using 'reshape' function to read and plot the displacements at critical planes and lines, the details of which are exactly like variables **D3** and **D6**.

Fex: It is an array of forces externally applied to the system and that includes the wheel loads at their corresponding degree of freedom. the size of this array is same as that **D** or **F** variables. The remaining elements of this array are filled with zeros.

GSM: It represents the global stiffness matrix of the model. It is a 2-D square matrix with a size of total degrees of freedom in the whole system.

GTE: It stores the individual number of elements (of all types) in the model. It is a vector quantity with its size equal to four, storing number of rails, ties, springs and bricks elements respectively.

GTN: It stores the grand total of nodes in the model, which is a scalar quantity.

iry: It is the rate of increase in width (in traverse or y-direction) using a geometric sequence beyond the edge of the ties. It is a scaler quantity. It helps generating a coarser mesh close the boundaries (least stress regions) and finer mesh close to rail and ties (high stress region).

irz: It is the rate of increase in depth (in vertical or z-direction) using a geometric sequence for the last substructure layer only. It is a scaler quantity. It helps generating a coarser mesh close the bottom of the model (least stress regions) and finer mesh close to track structure (high stress region).

L1n: It stores the node numbers of all the nodes in the system but in a systematic manner as they physically situated in space. It is 3-D matrix with its third dimension along x-axis. In other words, this will show all the nodes numbers in YZ-plane at x=0. And all the empty spaces are filled with zeros. This variable is of great importance in tracing the nodes and corresponding elements at specific locations.

LD: It stores layers data, which includes a bundle of information for all the layers in the substructure. It is a 2-D matrix with 18 rows and numbers of columns equal to the maximum number of layers defined by the user. Each row asks for a specific parameter for all the main layers to generate. These parameters include shoulder widths, side slopes, thicknesses, number of sublayers, initial Young's modulus, poison's ratios, linear or non-linear constitutive models

(power function) and related parameters, Gibson soil coefficients, deterministic or probabilistic models and its related parameters.

LoE: It stores the length of all the elements in the model. It is a vector with its row numbers matching with the element numbers.

lre: It represents the length of a single rail elements. The same length will later be used for brick elements length.

ltec: It represents the length of the tie elements between centerline and rail.

lteo: It represents the length of the tie elements between rail and edge of the tie.

MdSz: It represents the scale of the model, i.e., half or quarter scale model. It is a scaler quantity. Half scale model employs only one axis of symmetry along the longitudinal direction at the center of rails, whereas quarter scale model employs additional axis of symmetry along the traverse direction as well right below the wheel load application.

mGS: It is an array of Gibson soil coefficients for all the main layers in the system. First three entries are zero to represent zero coefficient for rail, tie and spring elements. Other entries are user-defined for substructure layers.

NECT: It is Node-Element connectivity table. It is a 2-D matrix with eight columns, for node numbers to be stored in a sequence explained in Section 1.3: Node Numbering, and number of rows equals the total number of elements in the model. The NECT is constructed such that the row numbers exactly match with the element numbers.

Nitr: It represents the number of Monte Carlo simulations or iterations to perform the probabilistic analysis.

NL: It represents the number of main layers in the substructure, for instance ballast, subballast, subgrade, natural soil etc. It is a scaler quantity.

nLE: It represents the number of line elements including rail, ties and spring.

nLN: It represents the number of line nodes, meaning nodes connected to line elements (rail and ties).

nnr: It represents the total number of nodes along the rail.

nnSL: It represents number of nodes in each sublayer. It is a vector with its size equal to NSL+1 to account for bottom most layer of nodes as well.

nnt: It represents the total number of nodes along a single tie.

nnto: It represents the number of nodes along the tie between rail and edge of the tie excluding the common node at the junction of rail and tie.

NoT: It represents the number of ties to consider and is a scaler quantity. It also represents the extend of the model in longitudinal direction (along x-axis).

Nr: It stores the information of number of elements between two ties. This is also a scaler quantity. This information is critical because this will determine the length (in longitudinal or x-axis direction) of all the rail and brick elements in the model.

NSL: It represents total number (sum) of sublayers in all the main layers. It is a scaler quantity.

NTE1: It stores the number of tie elements to generate along a tie (half-tie in this case) from center line to the edge of the tie.

NTEG: It stores the number of tie elements to generate along a tie segment between center line and rail.

StrAna: It stores the decision of whether to perform Stress analysis or not. It is scaler quantity with 0 or 1 value. Zero (0) means skip the stress analysis whereas one (1) means perform the analysis and plot the results.

StrMat: It stores geometric and material properties for rail and ties. It is a 2x6 matrix, first row is for rail properties and second row is for ties properties. Properties for both materials are stored in a sequence of Young's modulus, cross-sectional area, shear modulus, polar moment of inertia, area moment of inertia in y- and z-axis.

T: It is a set of transformational matrices to transform local stiffness matrices to global stiffness matrices for all three elements (rail, tie and spring). It is a 3-D matrix of size 6x6x3.

TcD: It represents the distance between tie mid-height to the rail mid-height. This height is just a number and not affecting anything in the analysis, that's why not included in the user inputs.

TnoL: It represents the tie number upon which the wheel load is applied. This variable help assign the vertical loading at the correct degree of freedom. Also, it helps in changing the scale of the model from half to quarter and vice versa.

ToE: It stores the types of all the elements in the model. It is a vector with its row numbers matching with the element numbers. Different types of elements are assigned different numbers, for instance 1 for rail elements, 2 for tie (sleeper) elements, 3 for spring elements connecting rails with the ties, 4 for ballast, 5 for subballast and 6 for subgrade and so on.

B.2. All Main and Nested-Functions

The purpose and detailed description of all the main and nested functions are presented in the following sections:

i. Main Program 'ADYTrack_4'

Purpose: This is the main program, which call all the Parent functions and serves as main control room of the code.

Details: Being the main code, this first call some built-in function ‘clear’, ‘clc’, ‘tic’ to clear the existing variables (if any), clear the screen and start a new timer, respectively. Then required global variables are called using ‘global’ statement to make them available for other functions and calculations.

All the functions (GenRandE, StifMat, BCond, Slove, Stress) required to perform Random Finite Element Method (RFEM) are called within a for-loop, running for ‘Nitr’ times, where ‘Nitr’ is user-defined number of iterations (Monte Carlo simulation). At the end RRPlot function is called to plot the risk and reliability plot.

ii. Function ‘Prepare’

Purpose: It reads the inputs parameters, store them in variables, generate geometry of the railroad trackbed, generate finite element mesh while assigning numbers to the nodes and elements.

Details: This function can be divided into three components, first reading and storing the input parameters in different variables, second generating the geometry and creating a coordinates matrix, and lastly generating a mesh and creating a node-element connectivity table. All three components are explained in detail in the following passages:

First component of ‘Prepare’ function is straightforward, reading the user defined inputs from Microsoft Excel file using ‘xlsread’ function and then storing the parameters in variables, the details of which is provided earlier in section Important Variables.

Second component of the function is generating geometry using nodes and assigning them x-, y- and z-coordinates. User can either build the half geometric model using the axis of symmetry along the longitudinal direction (along centerline of the rails or x-axis) or a quarter geometric model using both axis of symmetries along longitudinal and traverse

directions (along the ties or in y-axis). Nodes generation starts along the rail (x-axis), then along the ties (y-direction) starting from x=0 and moving towards x=max. Then a whole cross section of nodes is generated at x=0 and moving first along the y-direction followed by an incremental downward movement (z-direction).

Distances between the nodes in traverse direction (y-axis) at all depths right below the ties is kept constract, whereas nodal distrance in traverse direction beyond the edge of ties increases at a user-defined rate by following the geometric sequence. This help gradually change the mesh from fine to coarse towards the edge of the model. Similar scheme is emplyed for the nodal distance along the depth for only the bottom most layer.

The side slope of the track is generated using stepwise brick elements instead of smooth trapozoidal elements. Width (in y-direction) of the element at the farthest edge is decided such that slope passes through mid-height of the element.

Third component is the generation of Node-Element connectivity table (NECT matrix). Rail, ties and spring elements are modelled using 2-nodal line elemnts with 6-degrees of freedom (dof), whereas substructure is modelled using 8-nodal hexahedral brick elements with 3-dof. Element numbering followed similar proedure of numbering as that of nodes, i.e., rail elements are numbered first, followed by ties and springs, whereas bricks are numbered in the last in a similar manner as that of nodes. Alongside generating NECT matrix, two vectors namely type of elements (ToE) and length of elements (LoE) are also generated.

Also a 3-D transformational matrix with three 2-D matrics is generated along rail, tie and spring elements to transform the local stiffness matrix into global stiffness matrix and vice versa using a methodoly described in section 8.8 of [125]. In addition, a vector of Young's

modulus (Er) is generated for all elements by repeating the magnitude of each material/layer. This Er will later be used to generate randomized values of Young's modulus for all the elements using random field theories. Furthermore, an array of Gibson soil coefficient (G) is stored in mGS, which is used to calculate the modified (generally improved) Young's modulus with depth, (E=Eo+G*z), where E and Eo are the modified and basic Young's modulus values, and z is the depth below the ground surface.

iii. Function 'StifMat'

Purpose: This function first calculates the local stiffness matrices for all the elements and then assemble them into a global stiffness matrix.

Details: The function first sorts the line and brick elements to calculate their respective local stiffness matrices using nested Kline and K_Brick functions, respectively. The stiffness matrix of line elements in local coordinates system is then transformed into global coordinates using equation 2.1. For brick elements, already in global coordinates system, another nested function GenEmat is called to generated Stress-Strain (E) matrix, the details of which are provided in the following passages. After creating element stiffness matrices in global coordinates, these matrices are carefully weaved in global stiffness matrix at their respective locations. An important aspect of this global stiffness matrix is that it has mixed degrees of freedoms (dof), for instance, the line nodes have 6-dof (around top left corner of matrix) whereas brick elements composed of 3-dof.

iv. Function 'Kline'

Purpose: This function calculates element stiffness matrix in local coordinates (x-axis along the length of the elements) for all kinds of line elements.

Details: Element stiffness matrices for rail and tie elements as 2-nodal line elements with 6-dof (u_x, u_y, u_z, θ_x, θ_y, θ_z) are calculated using equation-2.3, whereas element stiffness matrix for spring elements connecting rail and ties as 2-nodal line element with 1-dof (u_x) and spring constant k is calculated Equation (3.3).

v. Function 'GenEmat'

Purpose: This function creates stress-strain matrix while incorporating the Gibson soil effect.

Details: First, the magnitude of corrected Young's modulus (E_c) for Gibson soil effect (with coefficient, G) for each element with initial Young's modulus E_o and depth z, is calculated using the expression $E_c = E_o + G*z$. Then the stress-strain matrix is calculated using equation 2.10.

vi. Function 'K_Brick'

Purpose: This function calculates element stiffness matrix directly in global coordinates for the brick element with 3-dof (u_x, u_y, u_z).

Details: The details of this function are presented in section 2.2 of this dissertation with all the relevant equations and references.

vii. Function 'BCond'

Purpose: This function assigns the boundary conditions at respective nodes including the displacement constraints and external (wheel) loadings.

Details: First, the selected displacement constraints are applied at five different plans to ensure local and global stability of the structure. To play a trick here, the displacement variable **D** was initially filled with ones and replace the specific constraint locations

(boundary conditions) with zeros. This helped differentiate nodes that are constraint or free to move.

All the nodes on the front (x=0) and back (x=max) cross sectional plane (YZ-plane) are constraints against longitudinal movement (u_x). Then lateral constraint (u_y) is applied on all the nodes at the centerline plane (XZ-plane) at y=0. In addition, all the nodes at the farthest plane (XZ-plane) at y=max are constraint in all three directions (u_x, u_y, u_z) to represent fix support system. This plane can be of any substructure layer, i.e., ballast, subballast, subgrade or natural soil, depending on the geometry of the trackbed. The model is also constraint using fixed supports by restricting the movement of all the bottom (z=max) most nodes in all three directions (u_x, u_y, u_z). Lastly, the wheel load (f_z) is placed in **Fex** variable at proper degree of freedom.

viii. Function 'Solve'

Purpose: This function solves the system of liner equations in the system to calculate the nodal displacements and nodal forces.

Details: First, the indices of zero displacements are identified to condense the global stiffness and force matrices by eliminating the corresponding rows and columns. Then the system of linear equations is solved using built-in '\' operator in Matlab. In order to save memory, the original **GSM** variable was condensed (deleted some rows and columns) and recalculated by calling StifMat function later to calculate the nodal forces and reactions at supports/constraints (**F** and **Fex**). Finally, some nested functions are called, the purpose and details of which are presented in the following paragraphs.

ix. Reshape

Purpose: As name implies, this function reshapes some arrays into matrices for convenient post processing (plotting).

Details: Two variables **D** and **Fex** are reshaped such that they transformed from arrays into matrices of **D3** and **Fex3** that are relatively easier to read and process (plotting). These matrices consist are three columns storing (u_x, u_y, u_z) or (f_x, f_y, f_z), while rows representing the node numbers.

x. Function 'TrackModulus'

Purpose: This function calculates Track Modulus.

Details: This function calculates the Track Modulus using Equation (5.1)

xi. Function 'Equilibrium_Checks'

Purpose: This function applies local and global equilibrium checks on sum of forces and moments.

Details: This function first check the global equilibrium of forces and moments, using $\Sigma F=0$ in all three directions and taking sum of moments about node-1, $\Sigma M_1=0$. Then it applies local equilibrium check at node-1 by taking sum of forces and moments concurrent at the node. Although one can input any other node as well for local equilibrium check, but that node must be connected to the rail beam. Finally, the results of these checks are displayed in the command window.

xii. Function 'NonLinCheck'

Purpose: This function checks the material nonlinearity (Stress dependent Young's modulus) of all the elements and calculates their correct Young modulus, if required.

Details: This function is only called from the main program /code (ADYTrack_4) if use defined stress dependent nonlinear constitutive for one or more substructure layers with a tolerance limit stored in variable **Tol**. To implement this nonlinear constitutive model, this function calls another nested function 'RevisedYmod' that store the current Young's modulus values (in variable **Erl**) and calculate the stresses in all the brick (substructure) elements and use these stresses in the use defined constitutive model to calculate the revised Young's modulus. Then this revised modulus is compared with **Erl** to find the maximum absolute difference. If the difference falls within tolerance limit, it proceeds to plotting the results else it performs structural analysis again using new modulus values and calling 'StifMat', 'BCond' and 'Solve' functions and then recalculate the revised modulus values using 'RevisedYmod' function. And this cycle repeats until the maximum absolute difference of modulus values fall within tolerance limits.

xiii. Function 'RevisedYmod'

Purpose: This function calculates the revised Young's modulus for all the elements with material nonlinearity.

Details: This function first stores the current modulus values for all the elements into another variable (**Erl**) for comparison purpose at the end of running cycle of calculations. Then this function calculates stresses in all brick elements using another function 'Stress1' to calculate the revised Young's modulus values for these brick elements using nonlinear constitutive model. And finally return vectors of both modulus values to the parent function (NonLinCheck).

xiv. Function 'Plots'

Purpose: This function plots the displacement, stress and risk analysis graphs for deterministic and probabilistic analysis.

Details: This function calls two nested functions, 'Plot_Disp' and 'Plot_Stress' to plot the nodal displacements and element stresses at predefined paths, respectively. For probabilistic analyses only, it also calls 'Plot_Risk' function to plot the probability of failure (or reliability or risk) plots. The details of all these three nested functions is provided in below passages.

xv. Function 'Plot_Disp'

Purpose: This function plots the displacements graphs along predefined paths and lines.

Details: This function plots three displacement graphs, first along the depth (vertical or z-direction) below wheel load, second along the loaded tie (traverse or y-direction) at the top of all main substructure layers (ballast, subballast, subgrade, natural soil etc.), and third along the rail (longitudinal or x-direction) at the top all main substructure layers.

xvi. Function 'Plot_Stress'

Purpose: This function performs two operations, first, check (whether stress calculations required or not?) and calculate the element stresses (if required) and second, plot the vertical stresses (σ_z) at predefined paths.

Details: If user chooses to skip the stress calculations (**StrAna**=0), it will return to the parent function (Plots), otherwise it will proceed to the stresses calculations and plot their respective graphs. It will calculate the vertical stresses (σ_z) along three paths, first along the depth (vertical or z-direction) below wheel load, second along the loaded tie (traverse or y-direction) at the top of all main substructure layers (ballast, subballast, subgrade,

natural soil etc.), and third along the rail (longitudinal or x-direction) at the top all main substructure layers.

For vertical stress calculations below the wheel, the stresses are calculated at the mid-height (depth) of each element. The stresses are first averaged out for two nodes of each element right below the wheel load and then further averaged out for the adjacent elements (2 in case of quarter-model and 4 in case of half-model) at the same depth right below the wheel to get a single stress point at the center of sublayer. Following the same procedure, the vertical stresses are calculated for the all the sublayers and are plotted using Plot_Str function.

For vertical stress calculations along the loaded tie at the top of all main substructure layers, the stresses are calculated at the center (mid-width) of each element. The vertical stresses are averaged out for 1^{st} and 4^{th} node (local numbering of elements) of the elements. Due to symmetrical loading and geometric conditions, there is no need of averaging the stresses from the adjacent elements. Following the same procedure, the vertical stresses are calculated for the all the sublayers and are plotted using Plot_Str function.

For vertical stress calculations along the rail at the top all main substructure layers, the stresses are calculated at the center (mid-length) of each element. The vertical stresses are first averaged out for nodes 3 and 4 (local numbering of elements) of the elements inside the rail and nodes 1 and 2 of the elements outside the rail. Then both these averaged stresses are again averaged to get a single stress point along the rail at each element. Following the same procedure, the vertical stresses are calculated for the all the sublayers and are plotted using Plot_Str function.

xvii. Function 'Plot_Str'

Purpose: This function plots the vertical stress graphs along predefined paths and lines.

Details: This function plots three vertical stress graphs, first along the depth (vertical or z-direction) below wheel load, second along the loaded tie (traverse or y-direction) at the top of all main substructure layers (ballast, subballast, subgrade, natural soil etc.), and third along the rail (longitudinal or x-direction) at the top all main substructure layers.

xviii. Function 'Plot_Risk'

Purpose: This function plots the risk and reliability (probability of failure) plot for the maximum vertical displacement (settlement) that takes place right below the wheel at the rail.

Details: This function plots the probability of failure (or risk and reliability) plot such that the maximum vertical displacement for all the simulations/iterations is plotted in the form of bar charts. These bar charts are then transformed into a continuous function (Probability Density Function or PDF) resembling the trend of bar charts and plotted onto the same graph. Using the same PDF function, its Cumulative Density Function (CDF) is plotted against secondary vertical axis. This plot can be red in two ways, first, find the maximum settlement against a given probability of failure/occurrence/exceedance (CDF value) and second, find the probability of failure (area under PDF curve or value against CDF curve) for a given maximum settlement.

APPENDIX C. RISK AND RELIABILITY PLOTS

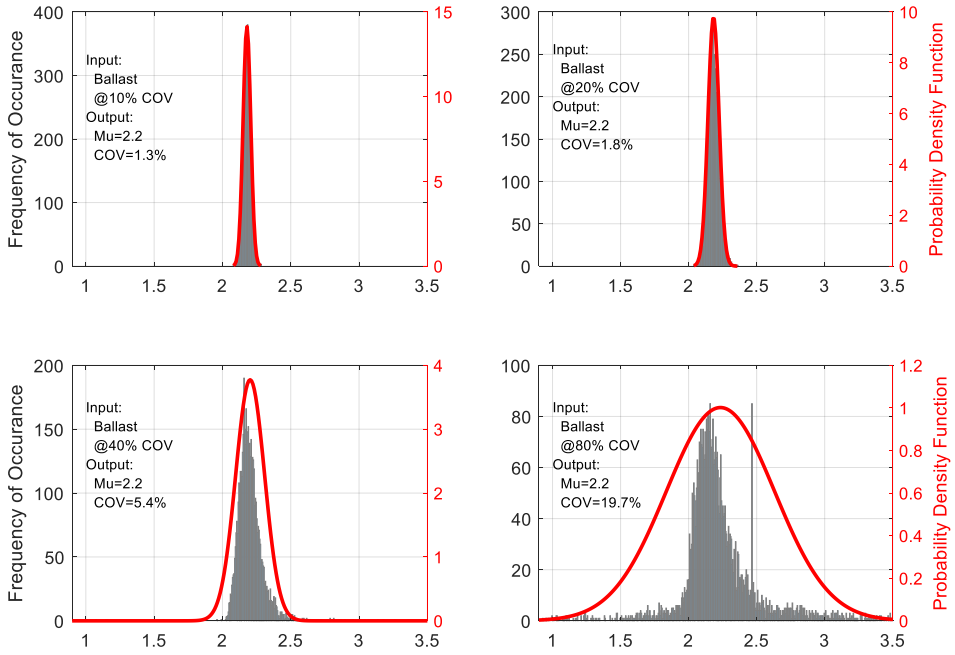

Figure C-1. Probability distribution of vertical displacement at the top of ballast layer due to variations in ballast layer.

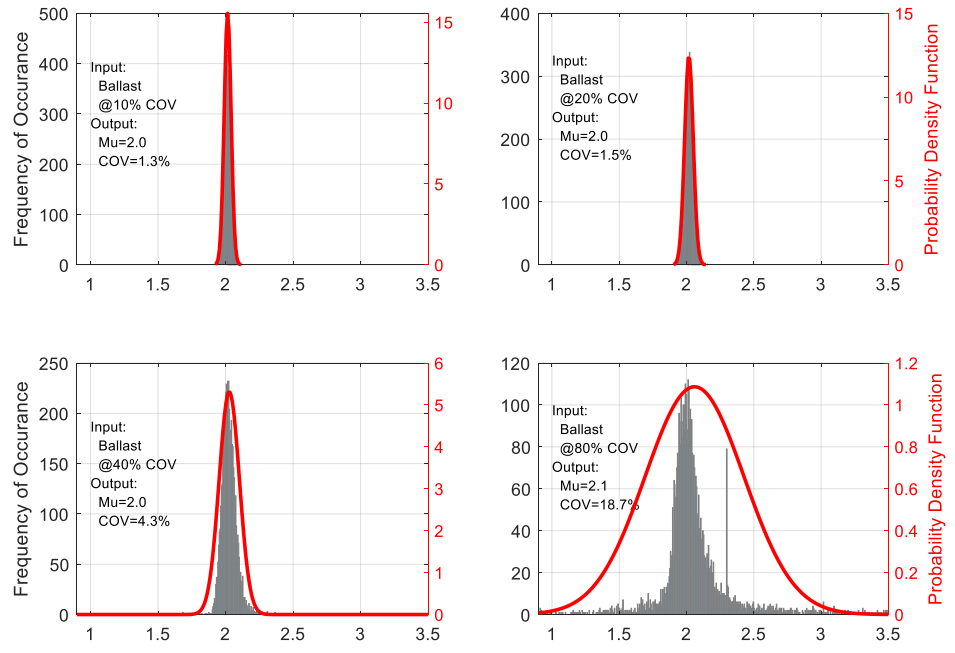

Figure C-2. Probability distribution of vertical displacement at the top of subballast layer due to variations in ballast layer.

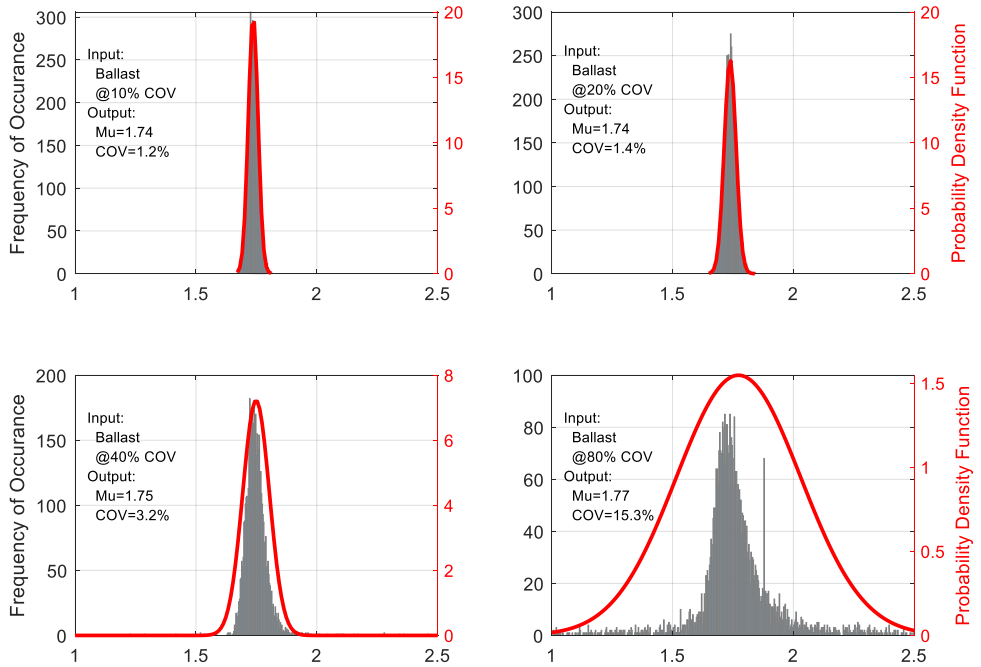

Figure C-3. Probability distribution of vertical displacement at the top of subgrade layer due to variations in ballast layer.

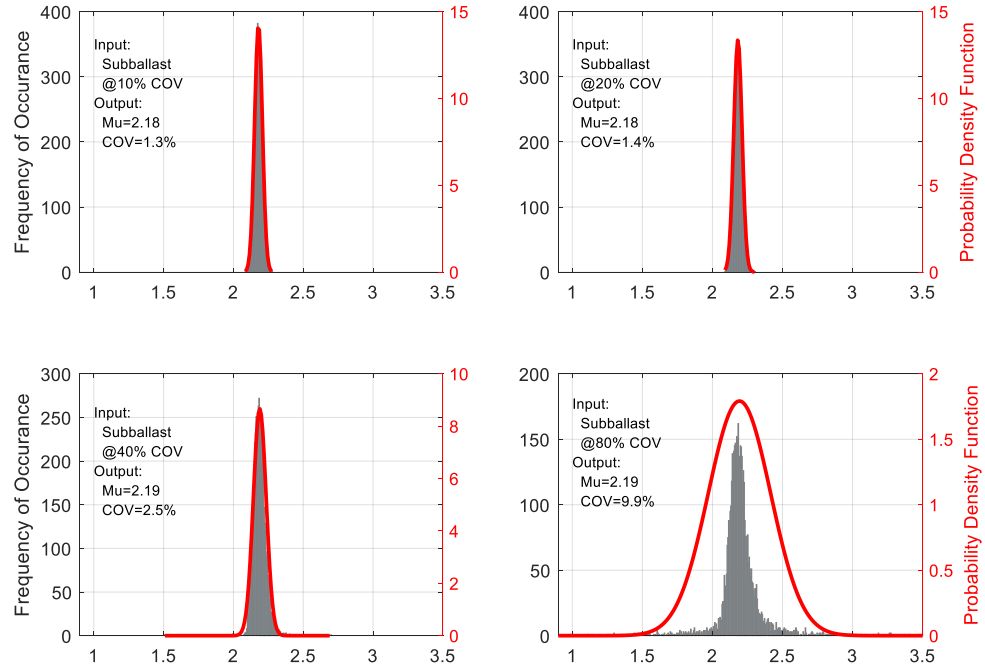

Figure C-4. Probability distribution of vertical displacement at the top of ballast layer due to variations in subballast layer.

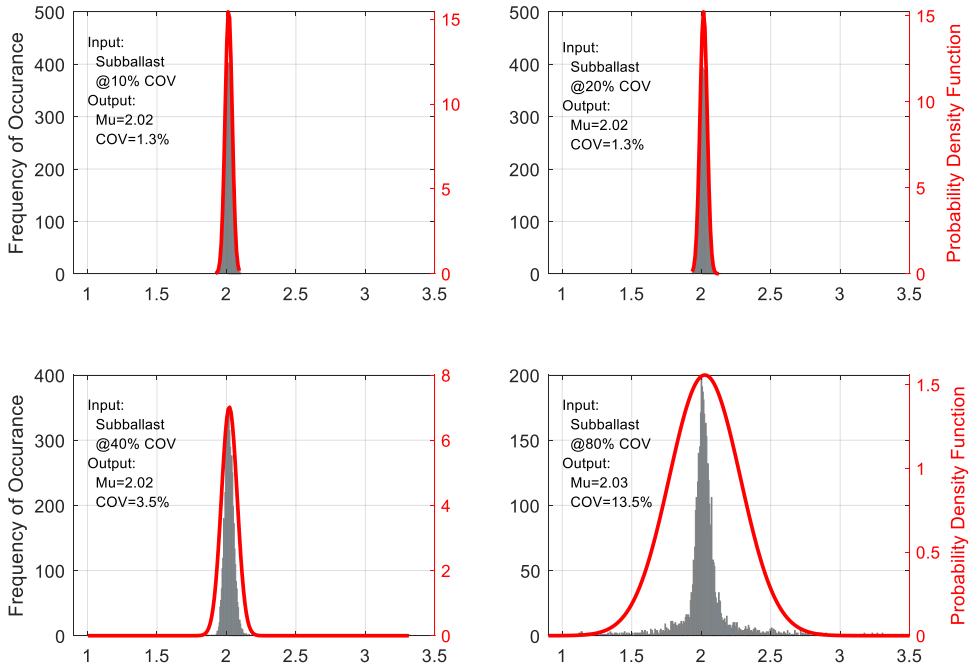

Figure C-5. Probability distribution of vertical displacement at the top of subballast layer due to variations in subballast layer.

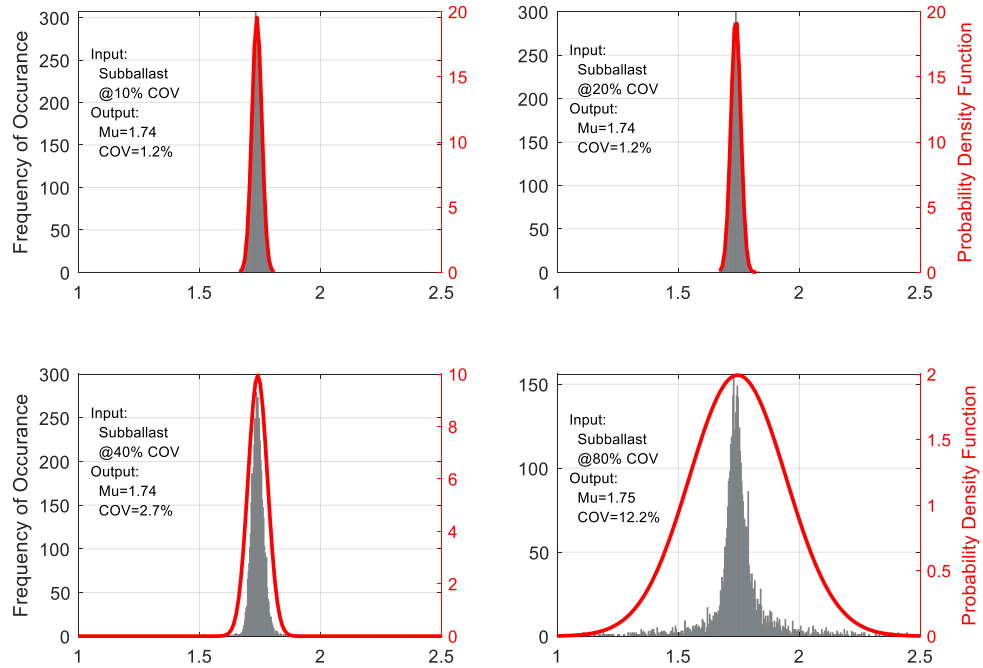

Figure C-6. Probability distribution of vertical displacement at the top of subgrade layer due to variations in subballast layer.

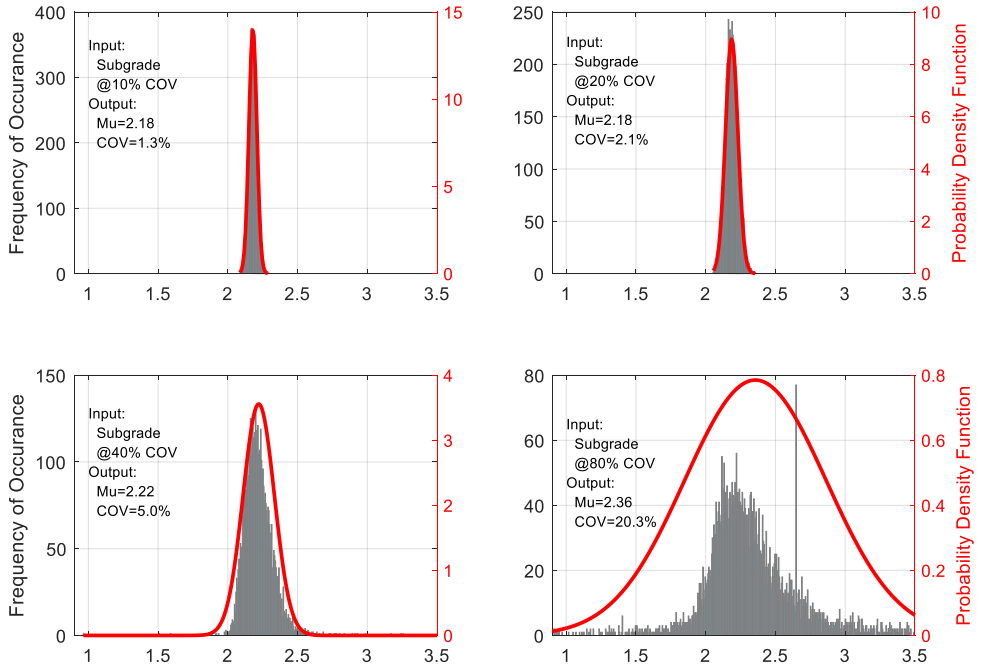

Figure C-7. Probability distribution of vertical displacement at the top of ballast layer due to variations in subgrade layer.

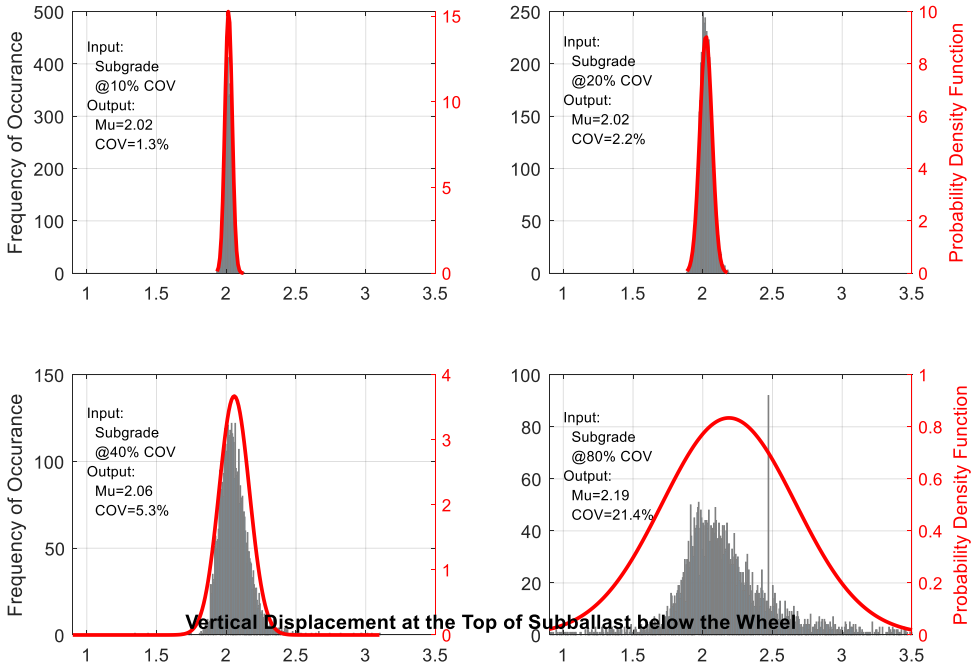

Figure C-8. Probability distribution of vertical displacement at the top of subballast layer due to variations in subgrade layer.

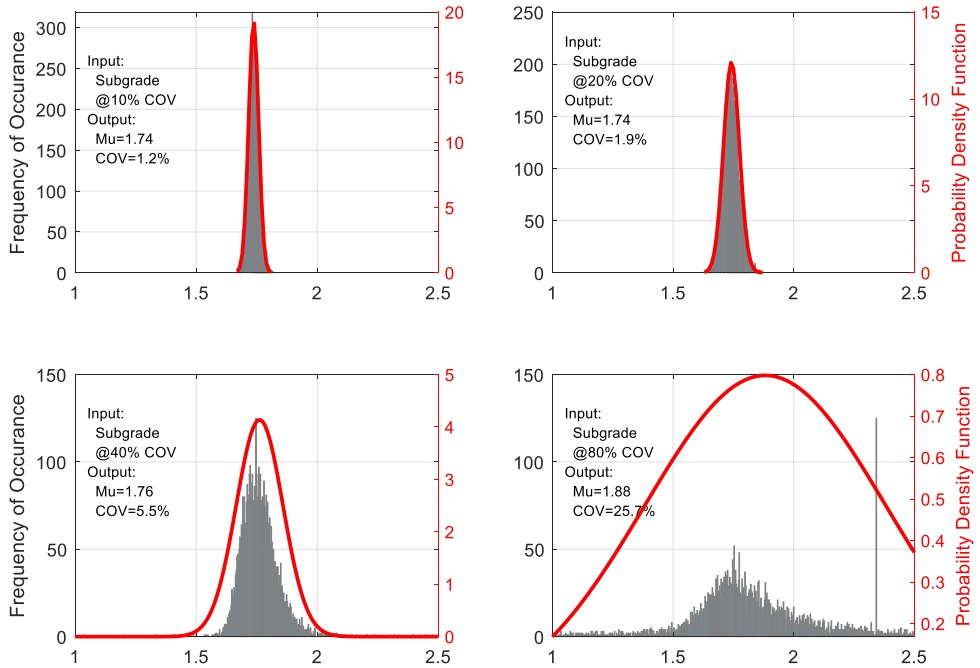

Figure C-9. Probability distribution of vertical displacement at the top of subgrade layer due to variations in subgrade layer.

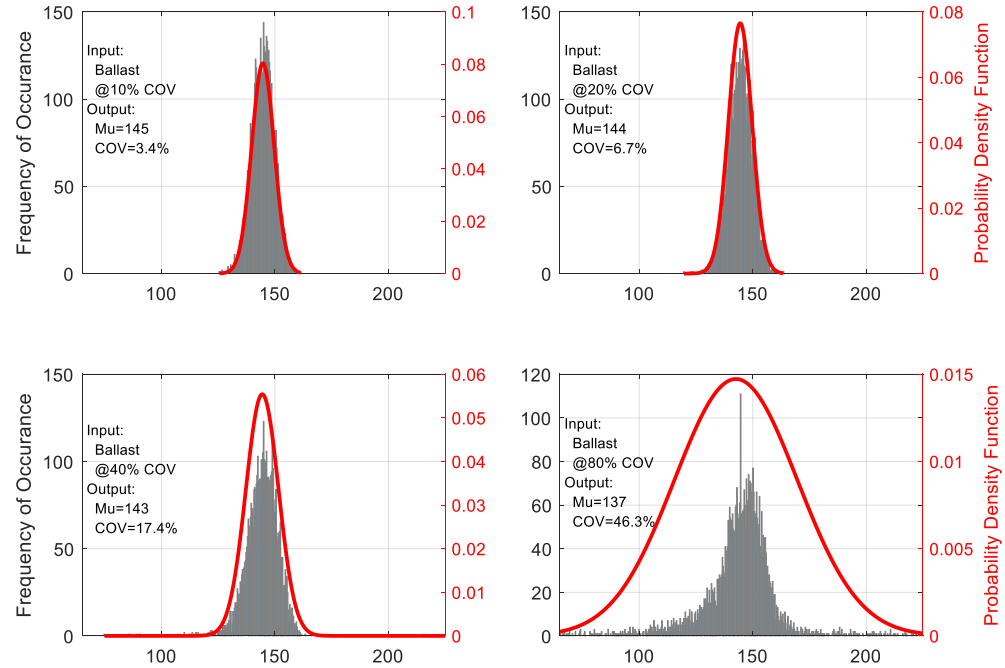

Figure C-10. Probability distribution of vertical stress at the top of ballast layer due to variations in ballast layer.

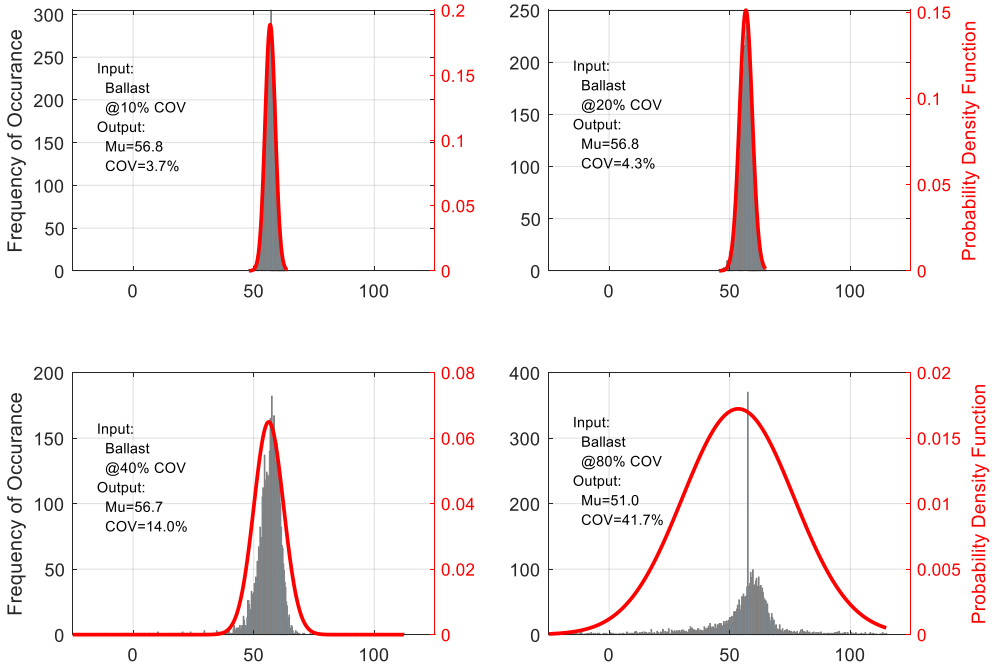

Figure C-11. Probability distribution of vertical stress at the top of subballast layer due to variations in ballast layer.

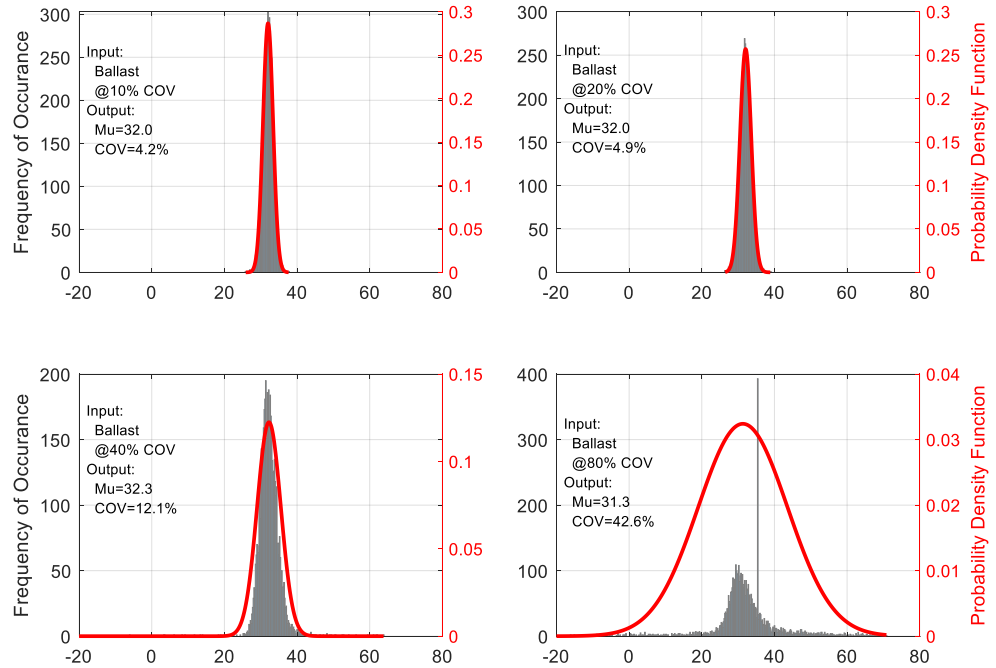

Figure C-12. Probability distribution of vertical stress at the top of subgrade layer due to variations in ballast layer.

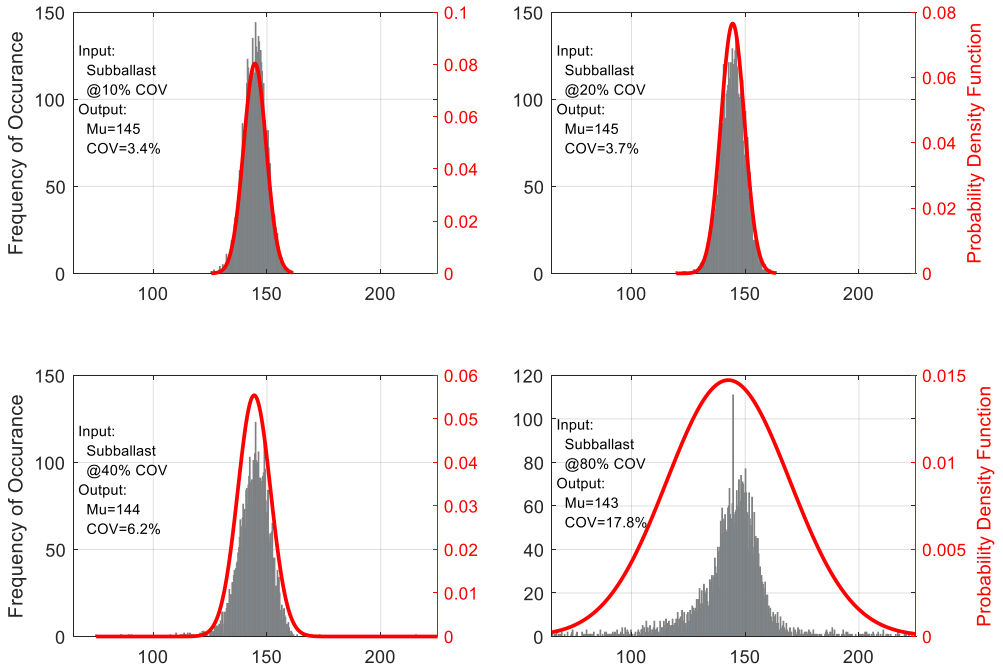

Figure C-13. Probability distribution of vertical stress at the top of ballast layer due to variations in subballast layer.

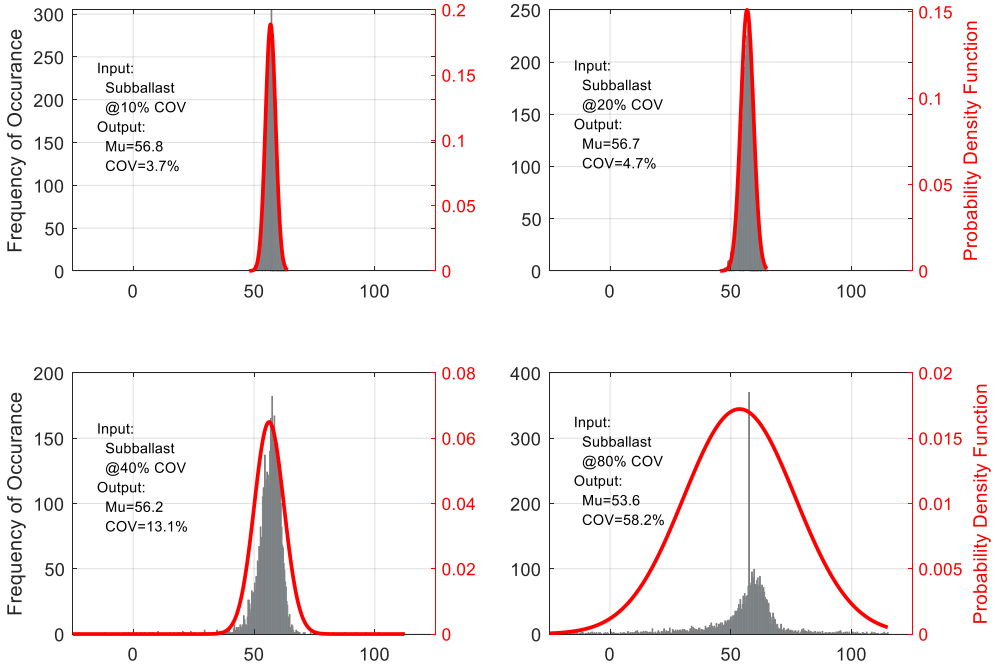

Figure C-14. Probability distribution of vertical stress at the top of subballast layer due to variations in subballast layer.

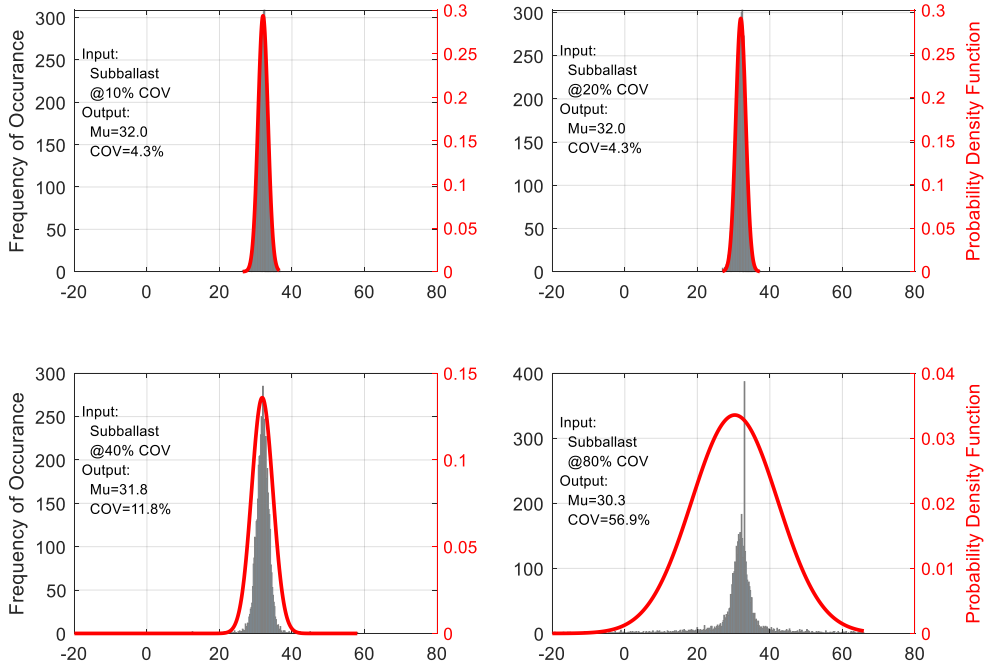

Figure C-15. Probability distribution of vertical stress at the top of subgrade layer due to variations in subballast layer.

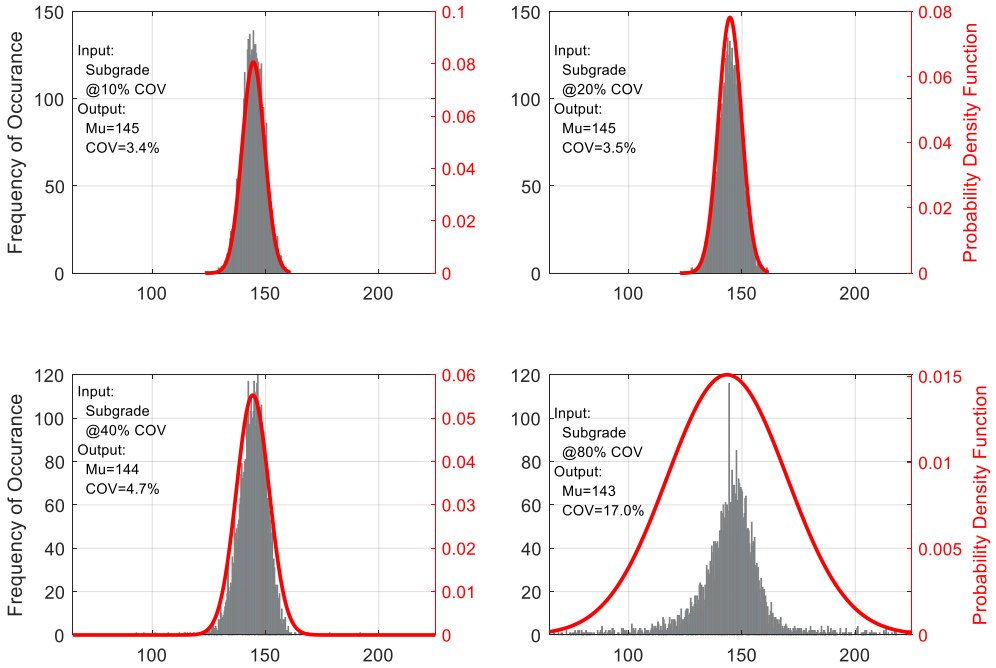

Figure C-16. Probability distribution of vertical stress at the top of ballast layer due to variations in subgrade layer.

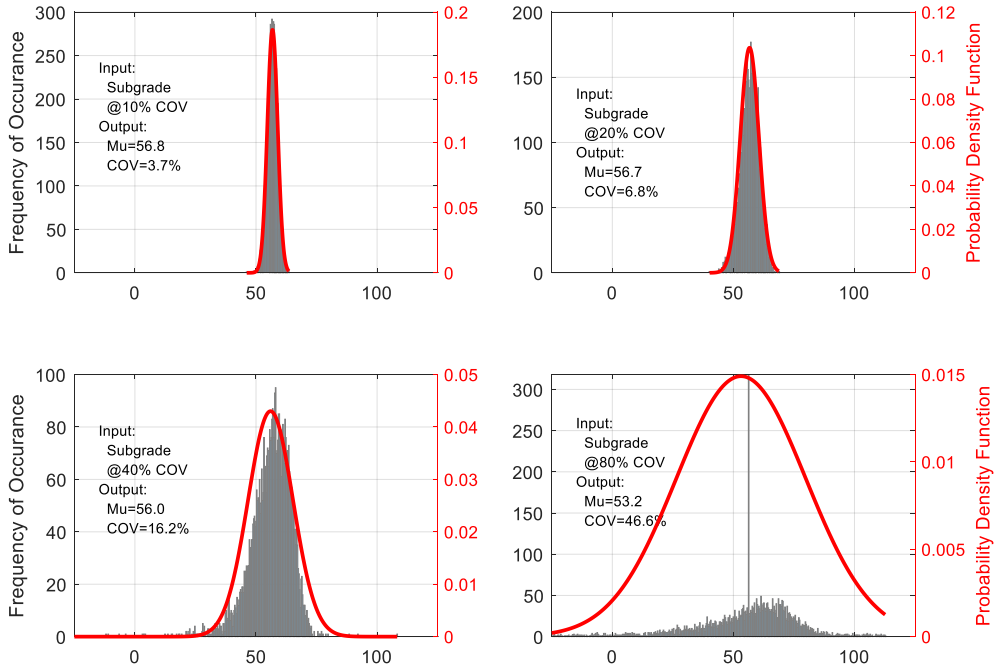

Figure C-17. Probability distribution of vertical stress at the top of subballast layer due to variations in subgrade layer.

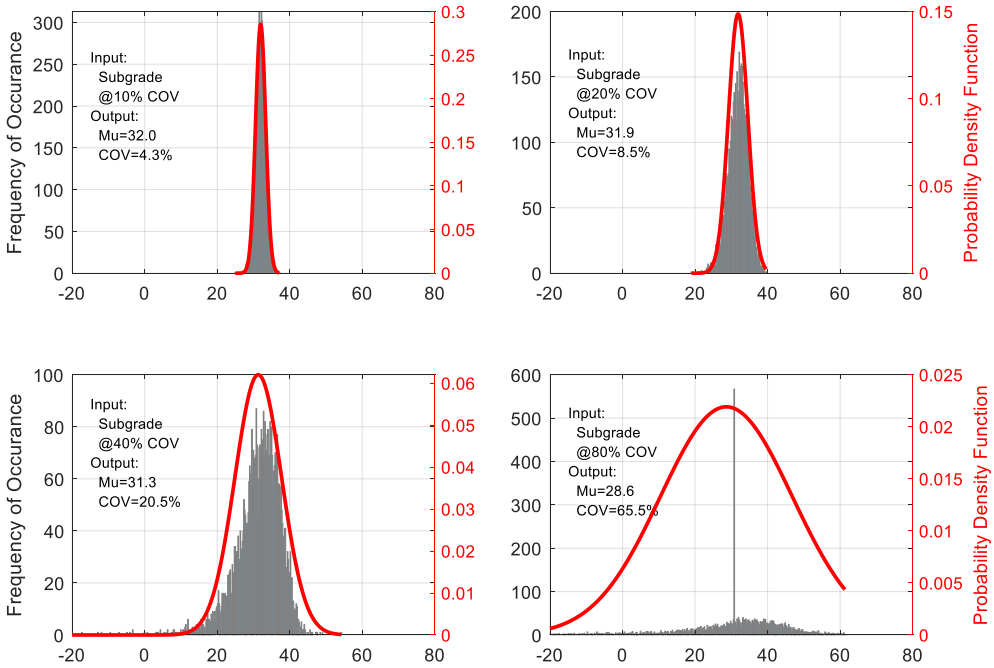

Figure C-18. Probability distribution of vertical stress at the top of subgrade layer due to variations in subgrade layer.

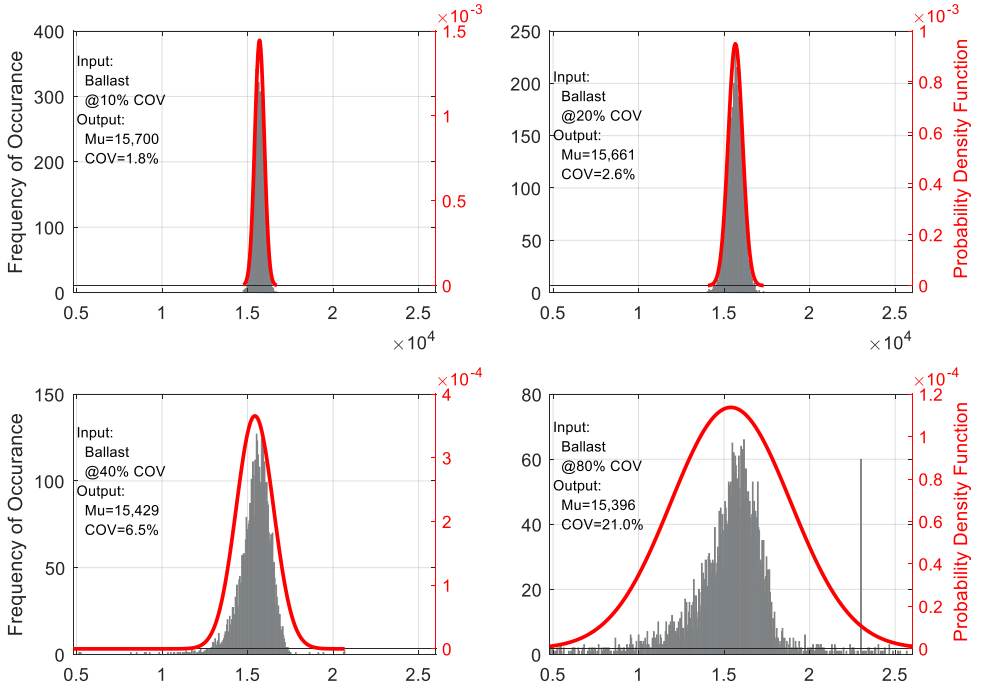

Figure C-19. Probability distribution of track modulus due to variations in ballast layer.

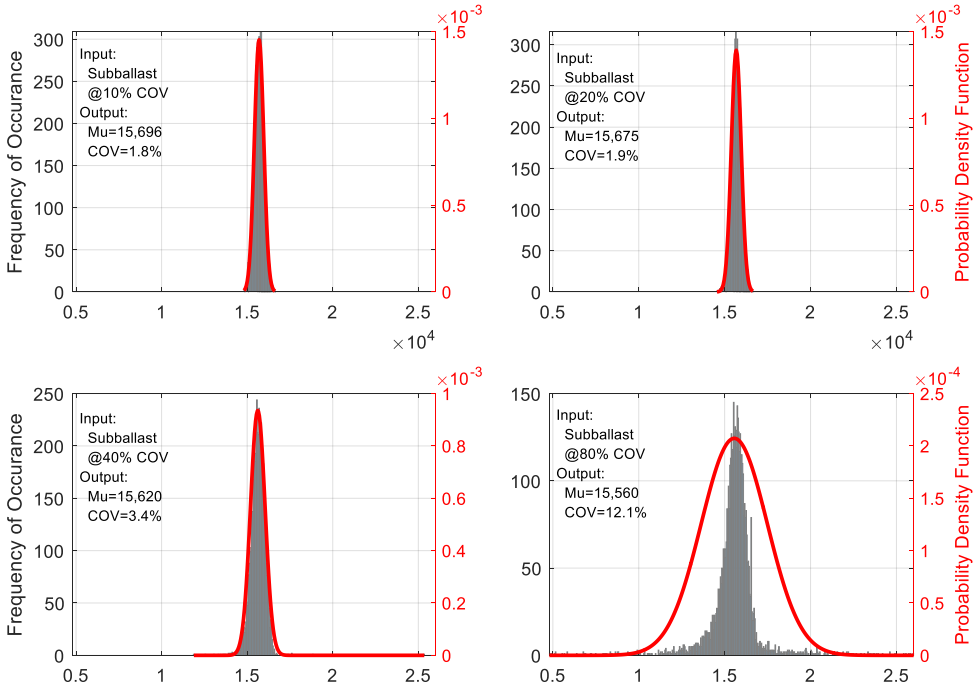

Figure C-20. Probability distribution of track modulus due to variations in subballast layer.

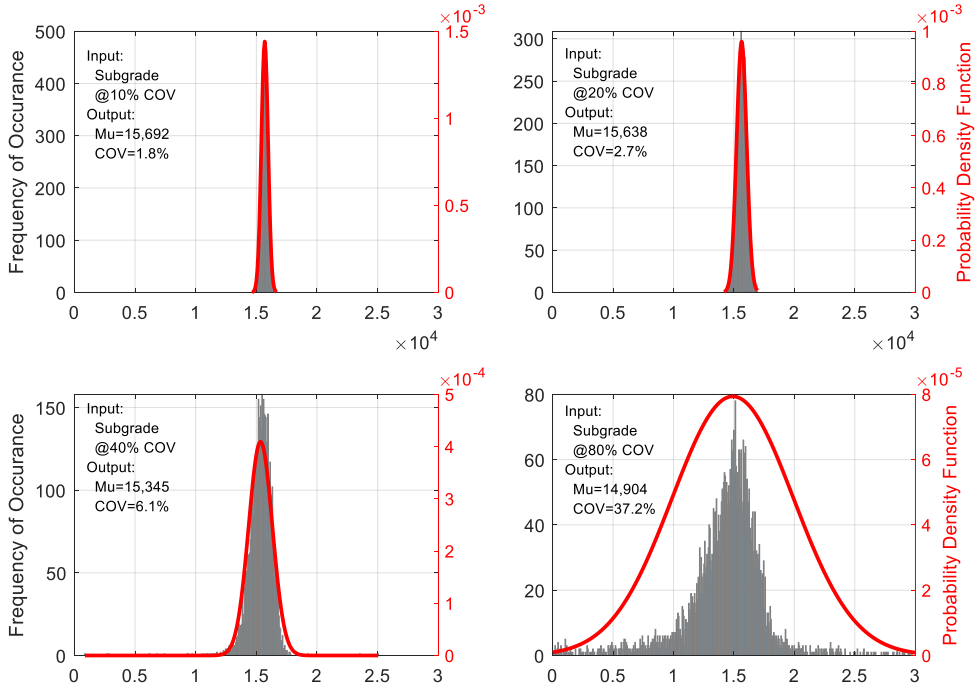

Figure C-21. Probability distribution of track modulus due to variations in subgrade layer.