

**RELIABILITY-BASED DESIGN OPTIMIZATION USING DDM ENABLED FINITE
ELEMENTS**

by

Stevan Gavrilovic

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(CIVIL ENGINEERING)

THE UNIVERSITY OF BRITISH COLUMBIA
(VANCOUVER)

July 2015

© Stevan Gavrilovic, 2015

Abstract

Rts is a risk-based structural optimization, multiplatform computer program that incorporates uncertainty into structural analysis with the utilization of random variable parameters. The major contribution to this thesis is that Rts now has the capability to perform reliability-based design optimization using Finite Element Method (FEM) analytical sensitivities. Analytical gradients are exact, more efficient, and convergence is achieved more rapidly in gradient-based optimization methods when compared to finite difference sensitivity methods.

For this thesis, I have derived and implemented both nodal and material analytical gradients throughout the Rts framework starting at the finite element level up through to the optimization level. The Reliability-Based Design Optimization (RBDO) model stream includes an FEM model, a COST model, a RISK model with built-in First-Order Reliability Model (FORM), and the orchestrating RBDO model. A program wide Direct Differentiation Method (DDM) framework was additionally established that provides efficient analytical gradient calculations throughout the model stream.

The FEM elements implemented consist of the Bilinear-Mindlin four node and nine node plate elements. An academic COST model was created to showcase the multi-model capabilities of Rts and the ability to calculate DDM dependencies of downstream models. Additionally, a RISK model was implemented that incorporated a built-in FORM model with gradient-history capabilities and in-model DDM dependency calculations; the RISK measure used is the mean cost. The RBDO model was also built upon to include DDM capabilities and downstream model integration.

Finally, two reliability-based design optimization examples were implemented using both nodal and material sensitivities. The thickness and width of a timber cantilever beam was optimized with respect to mean cost taking into account deflection damage and construction cost.

Preface

Components of this thesis were implemented in the Rts computer program created at the University of British Columbia. Rts is a merger of St and Rt. Rt was developed during the PhD studies of Dr. Mojtaba Mahsuli at UBC Vancouver while St was developed by Dr. Terje Haukaas. My contributions to the program were implementation of:

- plate finite elements with analytical gradients
- analytical gradients throughout the model framework
- a cost model template with analytical gradients
- a RISK model with integrated FORM analysis; extended original risk framework created by Alfred Larsen to include analytical gradients and integrated FORM optimization algorithms
- reliability-based optimization examples incorporating finite element analysis, cost models, and reliability models; all utilizing analytical derivatives from the downstream models

Table of Contents

Abstract.....	ii
Preface.....	iv
Table of Contents	v
List of Tables	viii
List of Figures.....	ix
List of Symbols and Abbreviations	xi
Acknowledgements	xii
Dedication	xiii
Chapter 1: INTRODUCTION.....	1
Chapter 2: ORCHESTRATING MODELS	5
2.1 RBDO Optimization Model.....	6
2.1.1 Optimization Search Algorithms	8
2.1.1.1 Steepest Descent Method.....	9
2.1.1.2 BFGS Method	10
2.2 RISK Model	11
2.3 FORM Reliability Model.....	13
2.3.1 FORM Limit-State Function.....	15
2.3.2 FORM Design Point Optimization	16
2.3.3 FORM Convergence Criterion.....	17
2.3.4 FORM Probability Transformations.....	18
2.4 COST Model.....	19
2.5 FEM Model.....	19

2.5.1	Energy Formulation	21
2.5.2	Principle of Virtual Displacements	21
2.5.3	Plate Finite Elements	25
2.5.4	FEM Mindlin Plate Shape Functions	28
2.5.5	Parametric Finite Elements	30
Chapter 3: ANALYTICAL GRADIENTS		37
3.1	FEM Gradients	41
3.2	COST Model Gradients	44
3.3	FORM Reliability Model Gradients	45
3.4	RISK Model Gradients	48
3.5	RBDO Gradients	49
3.5.1	Direct Differentiation Method – DDM	50
3.5.2	Adjoint Method – ADM	50
Chapter 4: IMPLEMENTATION AND EXAMPLES IN RTS.....		51
4.1	FEM Implementation	52
4.2	COST Implementation	53
4.3	RISK Implementation	54
4.3.1	FORM and COST Gradients	55
4.3.2	RISK Sensitivity Calculations	55
4.3.3	SAMPLING Gradients	56
4.4	RBDO Implementation	56

Chapter 5: RBDO EXAMPLES IN RTS.....	57
5.1 Optimization of Beam Thickness Using Material DDM Sensitivities.....	60
5.2 Optimization of Beam Width Using Nodal DDM Sensitivities.....	64
5.3 Simultaneous Optimization of a Beam Width and Thickness	68
Chapter 6: CONCLUSION.....	70
Bibliography	72
APPENDICES	73
Appendix A.....	74
A.1 Rts Input File Thickness Optimization	74
A.2 Rts Input File Width Optimization	77
Appendix B.....	80
B.1 Material Optimization Output.....	80
B.2 Nodal Optimization Output.....	89

List of Tables

Table 1. Element material properties matrices	23
Table 2. Shape functions of a plane quadrilateral from 4 to 9 nodes (Cook, Malkus, Plesha, & Witt, 2002)	32
Table 3. Examples of typical FEM parameters used in Rts	52
Table 4. Rts and Excel optimization example results comparison	59
Table 5. Rts model list of parameters and values	57
Table 6. Optimization analysis parameters	59

List of Figures

Figure 1. Graphical outline of orchestrating model stream in Rts	5
Figure 2. Stresses on a plate element	25
Figure 3. Coordinate mapping from x,y to ξ,η (Flaherty E, 2014)	30
Figure 4. Four node and nine node quadrilateral elements.....	33
Figure 5. Four node quadrilateral plate element	36
Figure 6. The gradient flowchart within Rts	37
Figure 7. Direct Differentiation Method (DDM) in Rts (Mahsuli & Haukaas, 2013).....	38
Figure 8. Optimization software architecture in Rts.....	51
Figure 9. Cantilever beam model in Rts	58
Figure 10. Deflected beam Rts model.....	59
Figure 11. Thickness optimization Rts GUI Ouput	61
Figure 12. Optimization plot; iteration number vs. objective function value	61
Figure 13. Thickness optimization CCDF	62
Figure 14. Thickness optimization full GUI	63
Figure 15. Thickness optimization plot	63
Figure 16. Thickness optimization CCDF Evolution	64
Figure 17. Width optimization full Rts GUI output.....	65
Figure 18. Width optimization plot; iteration number vs. objective function value.....	65
Figure 19. Width optimization RISK CCDF evolution plot.....	66
Figure 20. Width optimization Rts GUI Output	66
Figure 21. Width optimization; iteration number vs. objective function value	67
Figure 22. Width optimization CCDF curve.....	67

Figure 23. Iteration number vs. objective function value of simultaneous optimization	68
Figure 24. CCDF curve of of simultaneous optimization.....	68

List of Symbols and Abbreviations

θ_i = the random variables or parameters that influence the probability of a failure event occurring

B = strain-displacement matrix

BFGS = Broyden-Fletcher-Goldfarb-Shanno algorithm

C = cost response

D = material property matrix

DDM = Direct Differentiation Method

d = the vector containing element dofs

dof = Degree of Freedom

F = the vector of global nodal forces

FEM = Finite Element Method

FORM = First-Order Reliability Model

N = Vector or matrix of spatially varying shape functions

NDA = Nondeterministic Approach

N_i = element shape function

PDF = Probability Density Function

RBDO = Reliability-Based Design Optimization

u = the vector of global nodal displacements

u = displacement response

X = original-space vector consisting of all of the random variables or parameters θ_i

Y = standard-normal vector of the normalized random variables **X**

Acknowledgements

I would like to extend my gratitude to Dr. Terje Haukaas for inspiring me, offering his help and invaluable advice whenever I encountered roadblocks along the way. I am also grateful for his guidance and mentoring without which this would not have been possible. I would additionally like to acknowledge the civil faculty, fellow students, and staff at UBC for making it an enjoyable and worthwhile learning experience.

Special thanks are owed to my parents who have compelled me to always give my best and never give up. I am forever grateful for your support and the many sacrifices that paved the way for us to have the life that we have. I also want to thank my brother and sister for their love, understanding, and assistance.

Last, but not least, I want to thank my loving wife for her incessant support. I am lucky to have a person like you in my life. Thank you for putting up with the late nights and the many research hours that were spent on making this a reality.

Dedication

I dedicate this thesis to my wife and family. Without all of your support and care I would not know the meaning of love. Thank you for the encouragement and affection even when I didn't deserve it.

Chapter 1: INTRODUCTION

Traditional engineering problems often reduce and simplify a system so that it could be solved deterministically. Using this approach, engineers often introduce safety factors, or scale the solutions, so that they take into account some degree of uncertainty. Uncertainty is inherent in every methodology and it exists due to material variability, unpredictable loading, imperfect knowledge, and through errors involved in empirical understanding of physical systems to name a few. An example of a system where there is insufficient knowledge is in nuclear energy where the frequency of failures is so low that the probabilities and uncertainties may be difficult to estimate. Though the failures are rare, the consequences of failure are extremely severe.

As modeling processes become more sophisticated and new technologies emerge, the conventional deterministic approach is less efficient at providing solutions for making the best decisions. Complex problems involve varying degrees of uncertainty that need to be considered and quantified so that the best decisions are made with the knowledge that is available. A practical definition of uncertainty is: the knowledge gap between what is known and what needs to be known for making optimal decisions with minimal risk (Singhal, Ghiocel, & Nikolaidis, 2004). Methods that incorporate risk and uncertainty are termed Nondeterministic Approaches or NDA.

Nondeterministic methods quantify and manage uncertainty, reducing and mitigating its effects. The method of NDA that is used in this thesis is the Probabilistic Analysis Approach. In this approach, the system parameters are assumed to be random variables, with

the analysis incorporating their respective joint Probability Density Functions (PDFs). The primary objective of the Probabilistic Analysis Approach is to determine the reliability of the system. The reliability of a system is considered the ability of the system, or subsystem, to function under certain conditions for a specified period of time; in other words the probability of successful operation.

Successful operation consists of the structural system performing its function throughout its lifecycle. The ability of a system to successfully perform a certain function can be quantified using *performance measures*. Some examples of performance measures are cost, displacement, and stress responses. Examples of cost measures include cost of repairs, cost of degradation over time, and future cost of replacement. All of the above measures potentially involve large variations in uncertainty. When combined with inadequate knowledge of the problem, it is apparent why conventional deterministic methods fail to include all components involved in making sound decisions.

Reliability assessment addresses these issues with the selection of a suitable reliability model, analysis of the model, calculation of the reliability performance indices, and evaluation of the results with decisions on improvements (Singhal, Ghiocel, & Nikolaidis, 2004). The end result of a reliability assessment is a *risk measure*. Risk is defined as a measure of uncertainty; the quantitative and qualitative likelihood that a negative outcome may be realized. The final goal is to minimize risk using optimization techniques after the risk is quantified using a probabilistic approach. A typical optimization procedure uses algorithms to find the best possible combination of parameter properties to satisfy a set of

performance requirements and design constraints.

The optimization model requires the input of a risk model that incorporates a reliability model. In this thesis, the First-Order Reliability Method (FORM) is used to calculate the failure probabilities. A reliability model requires the input of a response model. Typically, the response input involves a cost function or structural response; in the examples presented in this thesis, the COST model is implicitly related to the structural response of the system and construction cost. The structural response is evaluated analytically using Finite Element Methods (FEM). Propagation of the uncertainty throughout the system starts at the FEM level where the input parameters can be random variables that carry on through to the final orchestrating Reliability-Based Design Optimization (RBDO) model. The governing models responsible for each subsection of the optimization analysis are termed *orchestrating models*. The order of orchestrating models in Rts from lowest to highest is: FEM, COST, FORM, RISK, and RBDO.

The RBDO model employs algorithms that quantify which parameters have the largest influence on minimizing the risk. This is termed *sensitivity analysis*. Parameter sensitivities involve gradient calculations that start at the FEM level and propagate throughout the model stream. To calculate the various gradients used throughout the model stream, exact analytical Direct Differentiation Methods (DDM) are used. Unlike approximate numerical methods, such as the Finite Difference Method, analytical methods are exact. Having accurate gradients greatly increases the efficiency and accuracy of the RBDO and FORM procedures.

This thesis will go over the orchestrating models implemented throughout Rts used to perform a reliability-based optimization. In addition, it will highlight the theory, derivations, and progression of analytical gradients starting at the FEM level up to the final RBDO model. After the background theory is established, this thesis will provide an overview of the Rts implementation and will conclude with working examples involving the optimization of a timber cantilever beam. The examples will include both nodal and material sensitivities that involve optimizing the beam's width and thickness.

Chapter 2: ORCHESTRATING MODELS

A typical structural optimization problem uses geometric state or response variables such as displacement, stress, or strain as constraints on the objective function. The objective function is a constrained mathematical function that is the target of the RBDO optimization.

Responses are calculated from structural equilibrium formulations such as analytically solving a partial differential equation or numerically from FEM techniques. Since analytical solutions are feasible for only the simplest of cases, FEM is predominantly used to calculate the response.

Rts goes beyond geometric responses by optimizing a risk expectance that is derived from a reliability analysis that incorporates the results from a FEM model which has random variable parameters. A graphical representation of the Rts model structure is given below:



Figure 1. Graphical outline of orchestrating model stream in Rts

The FEM model is the starting point of every optimization iteration i . The output of the FEM model is used by the COST model which assigns a cost based on the response from

FEM. The COST model can assign additional costs based on construction or damage to name a few. The final cost measure is then utilized by the FORM reliability model which is in turn governed by the RISK model; the RISK model specifies thresholds and calculates the mean cost using the probability of exceedance from the reliability model. Finally, the all-encompassing RBDO model minimizes the risk measure, or mean cost, by adjusting the values of decision variables. Sections 2.1 to 2.5 will explain the functions of each orchestrating model starting from governing RBDO model down to the low-level FEM model.

2.1 RBDO Optimization Model

A structure is an assemblage of components and elements that can vary in material, spatial, and cross-sectional properties to name a few. The assemblage must satisfy certain performance and design requirements based on its intended function. A design requirement could be to minimize cost, as in a commercial building, or weight, as in an aerospace component. It can also be a combination of design requirements such as to maximize the stiffness, strength, or stability while minimizing the cost or weight. A performance requirement could be a maximum allowable deformation or stress. Regardless of a structures prescriptive requirements, it must practically satisfy a set of constraints or bounds while being efficient. Finding the optimum structural layout involves employing a suitable optimization methodology. The procedure for solving reliability-based structural optimization problems in Rts is given by the following algorithm:

1. Select an initial design x^0 .
2. The displacement of the system $u(x^i)$ is calculated using FEM for the current design

- iteration x^i .
3. If applicable, using a COST model, estimate the cost of repair or damage $C(u(x^i))$ given the FEM response. Include other cost measures such as construction and degradation over lifecycle if appropriate.
 4. Calculate the probability of failure $P_f(x^i)$ for the current design iteration using FORM. Failure can be exceedance of a cost $C(u(x^i))$ or displacement $u(x^i)$, for example.
 5. Calculate the RISK measure $\mu(x^i)$ using the FORM output; the mean value of the aggregate cost for the current design.
 6. For the current design, formulate the objective function $\hat{g}(x^i)$ and calculate its gradient $\nabla \hat{g}(x^i)$.
 7. Utilizing a numerical optimization technique, calculate a suitable step size and direction to give a new design x^{i+1} .
 8. Repeat steps 2-7 until a convergence criterion is satisfied.

Note: At each iteration i of the optimization, the design state (the current values of decision variables) is represented by x^i .

The objective function (\hat{g}) is a function that is either maximized or minimized while using specified design criterion as constraints. It is composed of decision variables and state variables. Decision variables (θ) include those variables which are adjusted or iterated-over during the optimization procedure. A decision variable can be the thickness of a plate, the length of a beam, or the area of a cross-section. State variables ($u || C$) represent the response of the system; examples include cost, displacement, stress, or strain.

Therefore, to summarize, an optimization problem can be represented as:

$$\min \hat{g}(\theta, u||C) \text{ subject to } \begin{cases} \text{response constraints on } u||C \\ \text{design constraints on } \theta \\ \text{equilibrium constraints} \end{cases} \quad 1$$

Response constraints are typically written as the inequality $g(u||C) \leq 0$ where g is termed the constraint function. In structural optimization, the response $u||C$ is dependent on the design variable θ . Therefore, the two constraints can be combined to define a “nested” formulation in which all response and design constraints are written as the inequality $g(\theta, u||c) \leq 0$. The structural optimization (S \mathbb{O}) problem is repeated mathematically for completeness as:

$$(\text{S}\mathbb{O}) \begin{cases} \min_{\theta} \hat{g}(\theta, u||C) \\ \text{s. t.} \\ g(\theta, u||C) \leq 0 \end{cases} \quad 2$$

The solution of the S \mathbb{O} problem involves gradient-based numerical techniques that employ the derivatives of \hat{g} . At both the FORM and RBDO levels of the procedure numerical algorithms are used that employ gradients. Finding the gradients of the objective and constraint functions is a non-trivial task since $u(\theta)$ is implicitly carried through the model framework starting at the FEM level.

2.1.1 Optimization Search Algorithms

In iterative optimization problems, an important step is selecting a suitable step size and direction when updating decision variables. The sections below describe the two main search algorithms implemented in Rts that are used by the RBDO model; they are the Steepest Descent method and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. Both algorithms employ so called hill-climbing techniques that are used to find a local minimum.

The minimum is called a *stationary point*; a point where the gradient is zero.

2.1.1.1 Steepest Descent Method

The steepest descent method is a simple approach in where the search direction is equal to the gradient of the objective function. The step size is given by the user and remains fixed during the optimization procedure. Using this approach, the function will always make a step towards the design point. It can be described mathematically as:

$$\mathbf{p}_i = \alpha \cdot \nabla f(x_i) \quad 3$$

where

\mathbf{p}_i is the search direction vector,

α is a fixed step size, and

$\nabla f(x_i)$ is the gradient of the objective function evaluated at x_i .

Care must be taken when selecting a step size, for if the step size is too small, the search can be very inefficient. If the step size is too large, then the search may step over the design point thereby oscillating around the design point without converging; especially if the function exhibits large curvature around the design point. Further issues arise when conducting multi-parameter optimization as some parameters require a smaller step size for convergence while others can converge faster with a larger step size. This requirement can prove to be inefficient since multiple optimization iterations are required.

2.1.1.2 BFGS Method

Another method for selecting the step size and direction is known as the BFGS algorithm. BFGS attempts to solve the step size and convergence issues encountered with the steepest descent method by utilizing second-order approximations of the objective function. The BFGS method is considered a quasi-newton method since the Hessian is approximated using first derivative values rather than being evaluated directly. The BFGS analogue to Newton's formulation is given as:

$$\mathbf{B}_k \mathbf{p}_k = -\nabla f(x_k) \quad 4$$

where

\mathbf{B}_k is the approximation to the Hessian matrix,

\mathbf{p}_k is the search direction vector, and

$\nabla f(x_k)$ is the gradient of the objective function evaluated at x_k .

The BFGS algorithm is summarized as follows:

1. Make an initial guess \mathbf{B}_0 and x_0 .
2. Obtain a search direction \mathbf{p}_k where

$$\mathbf{p}_k = -\mathbf{B}_k^{-1} \nabla f(x_k) \quad 5$$

3. Perform a line-search to find the step size α^k (or use a fixed step size).
4. Find $x_{k+1} = x_k + \alpha_k \mathbf{p}_k$.
5. Set $\mathbf{s}_k = \alpha_k \mathbf{p}_k$.
6. Set $\mathbf{y}_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.
7. Find $\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k}$.
8. Repeat steps 2-6 until x_k converges to a solution.

2.2 RISK Model

The RISK model takes the exceedance probability output from the FORM and sampling models to produce an expectance or risk measure. The risk measure, in this instance, is the area under the Complementary Cumulative Distribution Function (CCDF) which represents the mean value of a total cost. The total cost can be the output from a collection of models consisting of any number of random variables or decision variables. Examples can include repair cost, construction cost, or replacement cost models to name a few. Inherently, the total cost is uncertain as it depends on the values of the random and design-decision variables. Hence, the risk measure is given as a mean value that is then used as an input into the RBDO model.

The procedure for calculating the risk measure is summarized as:

1. The sampling model does an initial sampling analysis of a specified number of samples to get a rough estimate of the mean and standard deviation.
2. A specified number of thresholds is then distributed equally on either side of the estimated mean. A larger number of thresholds gives greater resolution in the CCDF curve but it comes with a higher computational cost.
3. An integrated FORM analysis is then run for each threshold value with the corresponding exceedance probability as its output.
4. If the probability exceedance values are not above or below 0.005 or 0.995 at the extreme thresholds, then new thresholds are added in either direction until the exceedance probability falls within the bracket. This ensures that the summation of the risk measure does not exclude large portions of the area in the “tail ends” of the CCDF.

5. The resulting exceedance values are the points on the CCDF curve; with each point being the probability that the cost value will exceed a certain cost threshold.

For the examples used in Chapter 5 of this thesis, 10 initial samples were sufficient to estimate the mean. The number of cost thresholds used was 23. Using a higher number of sampling points or thresholds greatly increases computational cost. At every sample point the FEM model is run while the FORM model is run at every threshold value. FORM typically runs multiple instances of the FEM model. Therefore, if the structure is complicated, using an unnecessarily high number of initial samples or thresholds can be computationally detrimental. Every optimization requires sound judgment when selecting the number of sampling points or thresholds.

The integral, or area under the CCDF, gives the risk measure which in this case is the mean cost. Using the Trapezoidal Rule, the area under the CCDF curve is calculated numerically.

The mean cost μ is given as:

$$\mu = \frac{1}{2} \left\{ C_0(1 + P_{f_0}) + \sum_{i=0}^n (P_{f_{i+1}} + P_{f_i}) \cdot (C_{i+1} - C_i) \right\} \quad 6$$

where P_{f_i} is the exceedance probability from FORM and C_i is the cost threshold. It is worth reiterating that incorporating a higher number of thresholds results in a more precise measure of the mean, but it comes with a high computation cost due to FORM calling repeated COST and FEM models downstream.

2.3 FORM Reliability Model

The main goal of a structural reliability analysis is to evaluate the probability of a failure event occurring within a specified time period. A failure event could be defined as cost overrun, excessive deformation, damage to the environment, or loss of life. Mathematically, failure events can be described by a relation called the limit-state function:

$$F = P_f = \{g(\mathbf{X}) \leq 0\} \quad 7$$

The failure event F is realized when $g(\mathbf{X})$ is equal to or less than zero. The components of the original-space vector \mathbf{X} consist of all of the random variables, or parameters, θ_i that influence the probability of the failure event occurring. \mathbf{X} is given as:

$$\mathbf{X} = \{\theta_1 \ \theta_2 \ \dots \ \theta_n\}^T \quad 8$$

The limit-state function can be either linear or non-linear. In the first case, the limit-state function $g(\mathbf{X})$ is a linear function of continuous random variables \mathbf{X} . The probability of the failure event F can be described by the function:

$$P_f(\mathbf{X}) = P(g(\mathbf{X}) \leq 0) = \varphi(-\beta) \quad 9$$

where β can be described as the structural reliability index, given as:

$$\beta = \frac{\mu_g}{\sigma_g} \quad 10$$

In this context, β represents the number of standard deviations, or the spatial “distance,” from the failure plane. A larger β corresponds with an increase in safety and conversely a smaller value implies a greater failure probability. Taking into account all of the random variables, the total probability of failure can be determined by integrating a joint probability distribution function $f_{\mathbf{X}}(\mathbf{X})$ in original space as:

$$P_f(\mathbf{X}) = P(g(\mathbf{X}) \leq 0) = \int_{g(\mathbf{X}) \leq 0} f_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} \quad 11$$

Analytically solving this integral is non-trivial, therefore numerical approximation techniques are used to evaluate the probability of failure. Numerical integration methods are highly inefficient for increasing dimension of the vector \mathbf{X} . Consequently, a first-order approximation approach is adopted. Reliability analysis using FORM involves a transformation, or normalizing, of the limit-state function from the original space function $g(\mathbf{X})$ to the standard-normal space function $G(\mathbf{y})$. This transformation is accomplished by normalizing the components of the standard-normal vector \mathbf{y} from the original-space vector \mathbf{X} where each component y_i is given by:

$$y_i = \frac{\theta_i - \mu_{\theta i}}{\sigma_{\theta i}} \quad 12$$

By normalizing the limit-state function into standard-normal space, the normalized random variables y_i now have zero means and unit standard deviations. In standard-normal space the structural reliability index β is the shortest distance from the hyper-plane (or line in two dimensions) that forms the boundary between the safe domain and the failure domain to the standard-normal space *origin*. The point on the plane or line that is closest to the origin is denoted the design point, or the point where failure is most likely. This is also the point in the failure domain with the highest probability-density. In standard-normal space, the joint probability distribution function of uncorrelated standard-normal random variables is given by:

$$f_{\mathbf{y}}(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^n}} e^{-\frac{1}{2}\mathbf{y}^T \mathbf{y}} \quad 13$$

where \mathbf{y} is the standard-normal vector of the normalized random variables \mathbf{X} .

2.3.1 FORM Limit-State Function

A cornerstone of FORM involves the process of linearizing the limit-state function described in equation 9 at the design point. The design point \mathbf{y}^* is the point on the failure hyper-plane that is located at the closest distance to the standard-normal space origin. As described above, the shortest distance spatially between the safe and failure domain is called the reliability index and it is defined as:

$$\beta = \|\mathbf{y}^*\| \quad 14$$

In FORM, the limit-state function is linearized by a first-order Taylor approximation to produce:

$$G(\mathbf{y}) \cong G(\mathbf{y}^*) + \nabla G \mathbf{y}^{*T} \cdot (\mathbf{y} - \mathbf{y}^*) \quad 15$$

The term $G(\mathbf{y}^*)$ is equal to zero since we are evaluating the limit state function at its design point. It is common to normalize the gradient and negate it; this negative unit vector is referred to as the alpha-vector and is given as:

$$\boldsymbol{\alpha} = -\frac{\nabla G^T(\mathbf{y}^*)}{\|\nabla G(\mathbf{y}^*)\|} \quad 16$$

where each component is:

$$\alpha_i = -\frac{\frac{\partial G}{\partial \theta_i}(\mathbf{y}^*)}{\sqrt{\nabla G^T(\mathbf{y}^*) \cdot \nabla G(\mathbf{y}^*)}} \quad 17$$

$G(\mathbf{y})$ can now be expressed as:

$$G(\mathbf{y}) \cong -\|\nabla G(\mathbf{y}^*)\| \cdot \boldsymbol{\alpha}^T(\mathbf{y} - \mathbf{y}^*) = \|\nabla G(\mathbf{y}^*)\| \cdot (\boldsymbol{\alpha}^T \mathbf{y}^* - \boldsymbol{\alpha}^T \mathbf{y}) \quad 18$$

Beta is now redefined as:

$$\beta = \boldsymbol{\alpha}^T \mathbf{y}^* = -\frac{\nabla G^T(\mathbf{y}^*)}{\|\nabla G(\mathbf{y}^*)\|} \mathbf{y}^* \quad 19$$

which brings us to the final form of the linearized limit-state expression:

$$G(\mathbf{y}) \cong \|\nabla G(\mathbf{y}^*)\| \cdot (\beta - \boldsymbol{\alpha}^T \mathbf{y}) \quad 20$$

To find the design point \mathbf{y}^* , we must find the minimum β which can be accomplished using an optimization routine.

2.3.2 FORM Design Point Optimization

The design point is found by iteratively solving the following optimization problem:

$$\beta = \min_{\mathbf{y} \in \{G(\mathbf{y})=0\}} \sqrt{\sum_{i=1}^n y_i^2} = \|\mathbf{y}^*\| \quad 21$$

where \mathbf{y}^* is the design point or the distance closest to the failure surface. The above expression infers that the design point is located at the minimum distance from the failure surface linearized on the point \mathbf{y}_i by the line tangent to the surface $G'(\mathbf{y}) = 0$. For FORM to function, $G(\mathbf{y})$ must be continuously differentiable. The search algorithm for finding the FORM design point can be described as the following optimization problem:

$$\mathbf{y}^* = \arg \min \left\{ \frac{1}{2} \|\mathbf{y}\|^2 \mid G(\mathbf{y}) \leq 0 \right\} \quad 22$$

The iterative algorithm to find \mathbf{y}^* is formulated as:

$$\mathbf{y}_{i+1} = \mathbf{y}_i + S_i \cdot \mathbf{d}_i \quad 23$$

where S_i is the step size and \mathbf{d}_i is equal to a search direction. The FORM Model incorporates both fixed and Armijo step sizes. If the fixed step size fails to converge, then Armijo will take over. The search direction is given using the state-of-the-art iHLRF algorithm:

$$\mathbf{d}_i = \mathbf{y}_{i+1} - \mathbf{y}_i = \left(\frac{G(\mathbf{y}_i)}{\|\nabla G(\mathbf{y}_i)\|} + \boldsymbol{\alpha}_i^T \cdot \mathbf{y}_i \right) \boldsymbol{\alpha}_i - \mathbf{y}_i \quad 24$$

The iHLRF algorithm requires the normalized gradient $\frac{G(\mathbf{y}_i)}{\|\nabla G(\mathbf{y}_i)\|}$ which we will solve for analytically.

2.3.3 FORM Convergence Criterion

The design point is attained when the Karush-Kuhn-Tucker (KKT) optimality conditions have been satisfied. The two optimality conditions are:

$$\left| \frac{G(\mathbf{y}_i)}{G_0} \right| \leq e_1 \quad 25$$

and

$$1 - \frac{\boldsymbol{\alpha}_i^T \mathbf{y}_i}{\|\mathbf{y}_i\|} \leq e_2 \quad 26$$

The first condition ensures that the trial point \mathbf{y}_i is close to the limit state surface while the second condition is an angle-based criterion that requires the trial point on the limit-state surface to be the minimum distance to the origin. The values of e_1 and e_2 are in the magnitude of 10^{-3} . To summarize, the FORM algorithm involves the following procedures:

1. Select a starting point in standard normal space \mathbf{y}_i .
2. Transform the parameters from original space \mathbf{X}_i using equation 31.
3. Evaluate the limit-state function $G(\mathbf{y}_i) = g(\mathbf{X}_i)$.
4. Calculate the gradient of the limit-state function $\nabla G(\mathbf{y}_i)$.
5. Set the scaling factor for the first convergence criterion $G_0 = G(\mathbf{y}_i)$.
6. Check convergence using equations 25 and 26.
7. If convergence is not satisfied, iterate to find \mathbf{y}_{i+1} using equation 24 and a suitable step size.

8. Repeat steps 2-7 while skipping 5 until convergence.

The reliability index is only then computed at convergence, followed by the respective failure probability.

2.3.4 FORM Probability Transformations

As previously mentioned, in FORM reliability analysis, we seek to transform each random variable parameter from original to standard-normal space. The goal of probability transformations is to redefine a vector of random variables \mathbf{X} , with known probability distributions, to a vector \mathbf{y} , also with known probability distributions. This process involves determining the functional relation between the random variables; the random variables can be correlated or independent. In the optimization examples, the probability distributions used are normal and log-normal. The examples use independence probability transformations. For independent or uncorrelated random variables, the probability transformation is given as:

$$y_i = \Phi^{-1}(F_i(x_i)) \quad 27$$

from

$$F_i(x_i) = \Phi(y_i) \quad 28$$

and

$$x_i = F_i^{-1}(\Phi(y_i)) \quad 29$$

To find Φ^{-1} , we need the inverse of the Jacobian which we can derive by differentiation of equation 28:

$$\frac{\partial}{\partial x_i}(F_i(x_i) = \Phi(y_i)) = f_i(x_i) = \frac{\partial}{\partial x_i}\Phi(y_i) = \frac{\partial y_i}{\partial x_i} \frac{\partial}{\partial y_i}\Phi(y_i) = \frac{\partial y_i}{\partial x_i} \varphi(y_i) \quad 30$$

Finally, we get the diagonal Jacobian matrix necessary to complete the transformation with the components:

$$\frac{\partial y_i}{\partial x_i} = \frac{f_i(x_i)}{\varphi(y_i)} \quad 31$$

2.4 COST Model

The COST model is where the responses and geometry parameters from the FEM model are converted to a present-value cost. The COST model may include construction cost, future repair cost, degradation cost, financing cost, and environmental cost to name a few. The COST model can be extremely complex and is outside the scope of this thesis, only the model framework for DDM gradients was implemented along with simple examples for academic purposes. For this thesis, the cost model formulation is:

$$C(\theta) = c(u(\theta)) + d(\theta) \quad 32$$

Where $c(u(\theta))$ is a function that assigns a monetary value to a given displacement $u(\theta)$ and $d(\theta)$ is a construction cost that can be considered a function of a component's material and geometric properties. $c(u(\theta))$ can be considered a damage model while $d(\theta)$ could be a construction model that depends on material used and component geometry such as thickness of a plate or volume of a beam.

2.5 FEM Model

There are multiple approaches and derivations for implementing finite element solutions. The main idea behind FEM involves solving a boundary value problem that satisfies all equilibrium, kinematics, and material law requirements. The FEM model, as the name implies, involves discretizing a solid continuum into multiple 'finite' elements. In other

words, it involves meshing a continuous global domain into a collection of sub-domains. In the continuous domain, the responses are described by differential equations on an infinite number of points. In the discrete domain, the responses are described by a finite number of points and with a set of algebraic equations.

In most practical cases, the governing partial differential equations are extremely difficult to solve for point-wise or exact solutions due to the complexity of the solids and domains involved. As a result, the governing equations need to be numerically approximated, or averaged, over each discrete element within a discretized local domain. As more elements are used, and as the mesh spacing becomes infinitesimal, the solution approaches the exact value.

In structural analysis solutions, FEM involves approximating either or both of the displacement and force fields. The equilibrium, or compatibility approach, consists of approximating the displacement unknowns while the kinematics approach subjects the force unknowns to approximation. There are also hybrid approaches that combine both force and displacement approximations to arrive at a solution.

In Rts, the displacements are approximated to satisfy equilibrium. Hence, the remainder of this thesis will focus on the displacement-based FEM approach. Regardless of the governing equation formulations used to derive the force or displacement approximations; they can both be linked with virtual work or energy principles.

2.5.1 Energy Formulation

The energy quantity defined as *work* satisfies both the compatibility and kinematics conditions. More generally, *work* or dW can be defined as the energy required for the differential displacement ds of a point p along the direction of an applied force component F . Mathematically, work can be expressed as:

$$dW = Fds \text{ or as } W = \int Fds \quad 33$$

This integral conveniently represents both the equilibrium and kinematics quantities in a single expression. Since both forces and displacements appear in the definition of work, *virtual work* can be expressed as either the principle of *virtual displacements* or *virtual forces*. The conditions of equilibrium are derived from virtual displacements while compatibility is defined through virtual forces. In the case of a deformable body, such as a structural system, both the external and internal forces can do *virtual work*. This thesis will employ the principle of virtual displacements since it is the method predominantly used in solving structural mechanics problems.

2.5.2 Principle of Virtual Displacements

Using extremum formulations, structural problems can be solved by virtual displacements utilizing a variational calculus approach. Variational calculus involves minimizing an integral, in this case the potential function:

$$\Pi = U - H \quad 34$$

The potential function states that the strain energy of a system U is equal to the total potential of the loads H plus a constant Π . The values of U and H are obtained by summing (integrating) the energy contributions from each of the elements. The internal strain energy

of a deformable system is given by:

$$U = \frac{1}{2} \sigma \varepsilon \quad 35$$

The potential energy due to an external load is given by the area under the force-displacement relationship as:

$$H = \frac{1}{2} \mathbf{F} \cdot \mathbf{u} \quad 36$$

Integrating over the entire volume of a solid gives the following general expression:

$$\int_V \{\boldsymbol{\varepsilon}\}^T \{\boldsymbol{\sigma}\} dV - \int_V \{\mathbf{u}\}^T \{\mathbf{F}\} dV = \Pi \quad 37$$

The core concepts of the virtual displacement principle can be summarized in the following theorems:

- I. If a particle is in equilibrium, the total virtual work done during any arbitrary virtual displacement of the particle is zero.
- II. A system of particles is in equilibrium if the total virtual work done is zero for every independent virtual displacement.
- III. A deformable system is in equilibrium if and only if the total external virtual work is equal to the total internal virtual work for every virtual displacement consistent with the constraints.
- IV. Of all of the possible displacements which satisfy the boundary conditions of a structural system, those corresponding to equilibrium configurations make the total potential energy assume a stationary value.
- V. The displacements corresponding to stable equilibrium configurations make the total potential energy a relative minimum (Eisley, 1989).

The equilibrium equations are therefore obtained by minimizing the potential function:

$$d\Pi = 0 \quad 38$$

To accomplish this, we impose a virtual displacement or strain relative to equilibrium and set the potential function to zero. Integration by parts is then performed on equation 37 and 38 where the derivative is transferred from the minimizing function to the so-called variational δ . The symbol δ in this instance has the same meaning as the differential operator. The final outcome is equation 39 below.

$$\int_V \{\delta \boldsymbol{\varepsilon}\}^T \{\boldsymbol{\sigma}\} dV - \int_V \{\delta \mathbf{u}\}^T \{\mathbf{F}\} dV = 0 \quad 39$$

This FEM formulation is considered a ‘weak’ formulation since we neglect the twice-differentiability condition. Now the kinematic relationship between stress and strain can be formulated as:

$$\{\boldsymbol{\sigma}\} = [\mathbf{D}]\{\boldsymbol{\varepsilon}\} \quad 40$$

where $[\mathbf{D}]$ is a material properties matrix that depends on the element type. For example, various elements and their corresponding material matrices are given in Table 1 below.

Table 1. Element material properties matrices

Element	Material Matrix $[\mathbf{D}]$
Bar	E
Beam	EI
Plane Element	$\frac{E}{1-v^2} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{1-v}{2} \end{bmatrix}$

Substituting equation 40 into 39 yields:

$$\int_V \{\delta \boldsymbol{\varepsilon}\} [\mathbf{D}] \{\boldsymbol{\varepsilon}\} dV - \int_V \{\delta \mathbf{u}\} \{\mathbf{F}\} dV = 0 \quad 41$$

One of the cornerstones of the FEM is the ability to interpolate the displacements using shape functions. Let the displacements \mathbf{u} be interpolated over an element using linear shape functions $[\mathbf{N}]$ such as:

$$\{\mathbf{u}\} = [\mathbf{N}] \{\mathbf{d}\} \quad 42$$

where $\{\mathbf{d}\}$ is the vector of the nodal displacement at each Degree of Freedom (dof) of each element. We also define $[\mathbf{B}]$ as the strain displacement matrix:

$$[\mathbf{B}] = \nabla [\mathbf{N}] \quad 43$$

By definition of the strain-displacement relationship, strains are determined from displacements as:

$$\{\boldsymbol{\varepsilon}\} = \nabla \{\mathbf{u}\} \quad 44$$

Hence, combining equations 42, 43, and 44 we define the strain as:

$$\{\boldsymbol{\varepsilon}\} = \nabla [\mathbf{N}] \{\mathbf{d}\} = [\mathbf{B}] \{\mathbf{d}\} \quad 45$$

Substituting equations 42 and 45 into equation 41, we arrive at the final form of the FEM integral:

$$\{\delta \mathbf{d}\}^T \int_V \underbrace{[\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}]}_{[\mathbf{k}]} dV \{\mathbf{d}\} - \{\delta \mathbf{d}\}^T \int_V [\mathbf{N}]^T \{\mathbf{F}\} dV = 0 \quad 46$$

Where the element stiffness matrix is:

$$[\mathbf{k}] = \int [\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}] dV \quad 47$$

Equation 47 can be extended to any element, in any dimension. Rts uses this approach for line elements, bar elements, plane elements, and as of this thesis, plate elements. The

following sections will focus on the numerical solution techniques employed on plate elements since that was the element implemented in Rts for this thesis.

2.5.3 Plate Finite Elements

In general, plate elements are based on either Kirchhoff or Mindlin plate theories. Kirchhoff plate theory excludes transverse shear deformation while Mindlin plate theory accounts for it. This is analogous to beam bending where Euler-Bernoulli beam theory neglects shear deformation while Timoshenko theory includes it. Also, much like beam theory, as a beam gets deeper (or a plate becomes thicker) transverse shear plays a greater role in the accuracy and reliability of the computed results. The stresses and stress resultants are highlighted on a plate element in Figure 2 below.

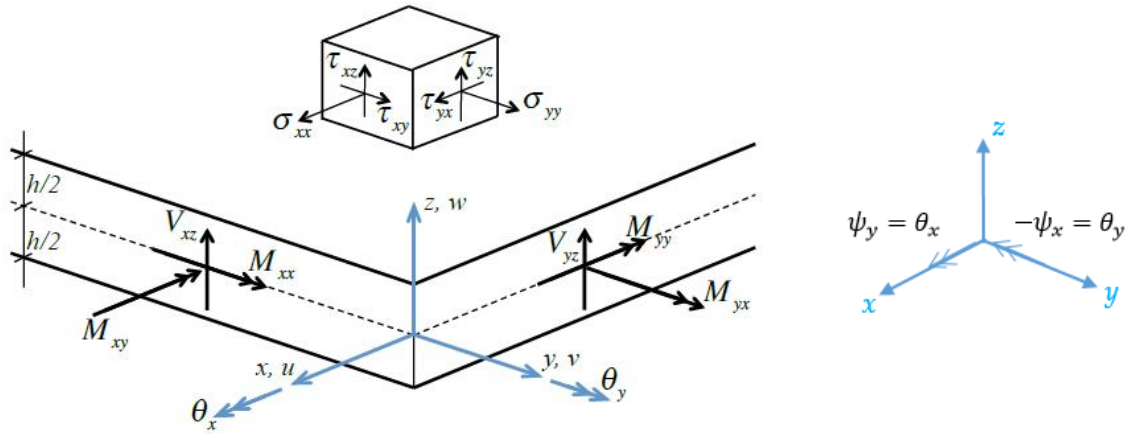


Figure 2. Stresses on a plate element

Plates develop bending moments in the two principle directions along with a twisting moment. Positioned with the x - y plane at its midsurface, both rotations at the midsurface and surface slopes are used to describe plate elements. The midsurface rotations θ_x and θ_y are positive when pointing in the x and y directions. The rotation or slope on the surface of the plate element is given as $w_{,x}$ and $w_{,y}$; where the comma represents differentiation with

respect to the following subscript. In Kirchhoff plate theory, a straight line normal to the midsurface is assumed to remain straight and normal to the deformed midsurface; transverse shear deformation is zero. This requirement is relaxed in Mindlin theory. In Rts, Mindlin plates were implemented because they provide more options such as the ability to analyze layered plates where transverse shear deformation is often important. Additionally, they can be easily extended to become shell elements. The drawback of using Mindlin plates is that they are susceptible to shear locking and require more expertise to implement (Cook, Malkus, Plesha, & Witt, 2002).

The plane-stress expression for plate elements is as follows:

$$[\sigma_b] = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} \quad 48$$

and for shear as:

$$[\tau_s] = \begin{Bmatrix} \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \quad 49$$

where E is the Young's modulus and the shear modulus is:

$$G = \frac{E}{2 + 2\nu^2} \quad 50$$

The b and s subscripts in equations 48 and 49 denote bending and shear respectively.

Kirchhoff plates utilize only σ_b while Mindlin plates incorporate both σ_b and τ_s . The full plane-stress expression for Mindlin plates is:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = \begin{bmatrix} & & & 0 & 0 \\ & [\sigma_b] & & 0 & 0 \\ & & & 0 & 0 \\ 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & G \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \quad 51$$

Since we are dealing with plane stress, ε_z and σ_z are considered negligible.

Kirchhoff plates need only be represented by a single field $w(x,y)$; the lateral deflection of the midsurface. The stress and deformation throughout a Mindlin plate is expressed as three fields; $w(x,y)$, ψ_x , and ψ_y . The variables ψ_x and ψ_y being the rotations of the midsurface normal along x and y respectively. With Kirchhoff elements, $w_{,x} = \psi_x$ and $w_{,y} = \psi_y$ since the slope at the midsurface is assumed to equal the slope at the surface; hence only the field $w(x,y)$ is sufficient to represent the total state of deformation.

The moment-curvature relationship for a homogenous, isotropic, and linearly elastic Mindlin plate is:

$$\begin{Bmatrix} M_{xx} \\ M_{yy} \\ M_{xy} \\ Q_x \\ Q_y \end{Bmatrix} = \begin{bmatrix} D & D\nu & 0 & 0 & 0 \\ D\nu & D & 0 & 0 & 0 \\ 0 & 0 & \frac{D(1-\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & khG & 0 \\ 0 & 0 & 0 & 0 & khG \end{bmatrix} \begin{Bmatrix} \psi_{x,x} \\ \psi_{y,y} \\ \psi_{x,y} + \psi_{y,x} \\ \psi_x - w_{,x} \\ \psi_y - w_{,y} \end{Bmatrix} \quad \text{or} \quad \{\mathbf{M}\} = [\mathbf{D}]\{\mathbf{k}_c\} \quad 52$$

where

D = the material property matrix $\frac{Eh^3}{12(1-\nu^2)}$,

G = is the shear modulus $\frac{E}{2+2\nu^2}$,

k = effective thickness $5/6$,

h = plate thickness, and

ν = material Poisson's Ratio

D is analogous to flexural rigidity EI in beam theory while G denotes the shear modulus. kh can be regarded as the “effective thickness” for transverse shear deformation. Note that the

lateral deflection field in Mindlin theory is coupled to the rotation fields only by transverse shear deformation. The curvature deformations $\{\mathbf{k}_c\}$, the far-right vector of equation 52, are approximated through shape functions. The following sections will provide an overview of the shape functions used for the Mindlin plate element implementation.

2.5.4 FEM Mindlin Plate Shape Functions

In mathematics, any continuous function can be represented as a linear combination of discrete *basis* functions. To discretize the continuum domain, an appropriate ‘basis’ function must be selected. A shape function is considered a restriction of a basis function to an individual element. These unique basis functions are related to the geometry of the meshed element; either through a vertex, edge, face or the whole element in simple geometries. Constructing a basis function in terms of a shape function eliminates the need to know specific geometric information about overall element layout and connectivity. Basis functions are constructed by combining the shape functions of the neighboring elements; this ensures connectivity and continuity through the system. 2D elements preserve edge continuity while 3D elements require face continuity as well.

Placing a node at the element’s vertices ensures a continuous basis. The neighboring edges share the same nodal values; the edges are not collinear and by definition the shape functions from both edges are determined by the same two nodal values at their shared vertex; hence, the basis is continuous. When constructing shape functions, the objective is to construct a p -*th* degree polynomial approximation. The resulting polynomial must be linear, unique, and complete. In two-dimensions, Pascal’s triangle is used to construct shape functions. These

piecewise, linear polynomials are referred to as Langrangian Shape Functions. Including all of the terms in the polynomial ensures isotropy in that the degree of the shape function is retained under coordinate translation and rotation. The shape functions described above take the form:

$$N_j(x, y) = \sum_{i=1}^{n_p} a_i q_i(x, y) = a^T q(x, y) \quad 53$$

where a^T are unique coefficients and where

$$q^T(x, y) = [1, x, y, x^2, xy, y^2, \dots, y^p] \quad 54$$

Quadratic shape functions have the form:

$$N_j = a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 \quad 55$$

where the six coefficients $a_j \dots j = 1 \dots 6$ are determined uniquely by the values of the shape functions of the three nodes on a given edge. The values of the shape functions on shared edges are determined by the same nodal values in neighboring elements.

Constructing Langrangian approximations, though straight forward to implement, can be algebraically complicated. To reduce the complexity, an element can be transformed into a canonical or parametric element. This coordinate transformation is achieved by mapping an arbitrary element from the (x, y) plane to a canonical element in the (ξ, η) plane. The ability to solve any problem using arbitrary coordinate transformations facilitates the straightforward formulation and solution of differential equations using these parametric elements.

2.5.5 Parametric Finite Elements

Isoparametric mapping permits quadrilateral elements to have non-rectangular shapes. This technique uses reference coordinates to map an arbitrary physical element into a “reference element.” The mapping of a 2-dimensional four node quadrilateral element to a canonical element is shown below in Figure 3.

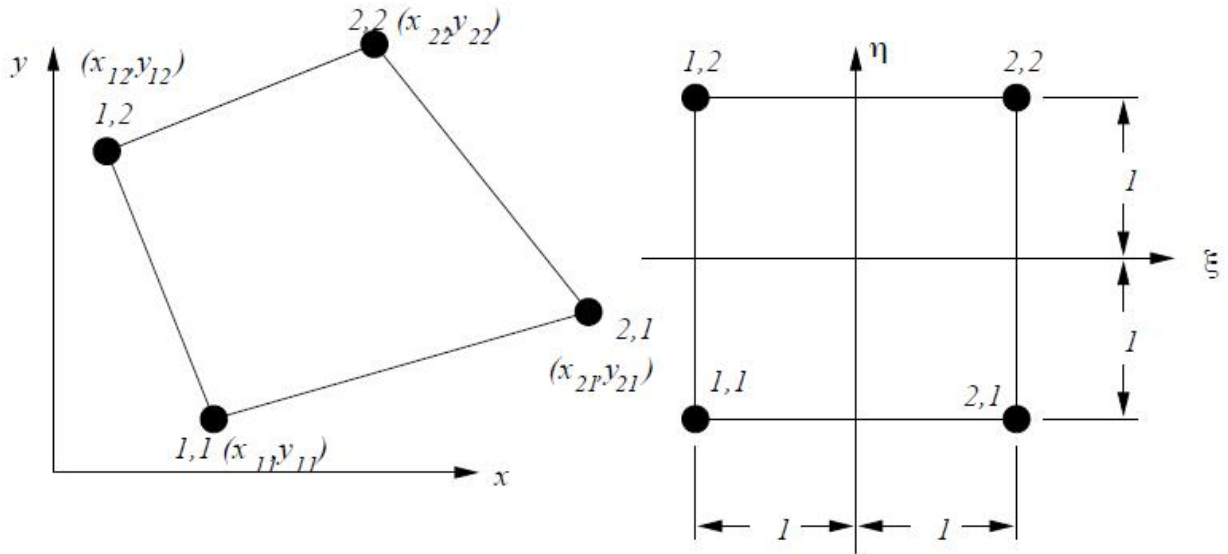


Figure 3. Coordinate mapping from x,y to ξ,η (Flaherty E, 2014)

This is accomplished with coordinate transformation using shape functions that interpolate both the displacement field and the element’s geometry. The displacement of a point within the element can be expressed in terms of the nodal dof and the shape functions $[\mathbf{N}]$ which are also functions of the reference coordinates. The global coordinates of a point within the element can be expressed as shape functions $[\tilde{\mathbf{N}}]$; these are also functions of the reference coordinates.

An element is called isoparametric if $[\tilde{\mathbf{N}}]$ is identical to $[\mathbf{N}]$. If $[\tilde{\mathbf{N}}]$ is of lower degree than $[\mathbf{N}]$, then the element is called subparametric. If $[\tilde{\mathbf{N}}]$ is a higher degree than $[\mathbf{N}]$, then the

element is superparametric. Isoparametric elements were incorporated in Rts for this thesis.

When using any form of parametric elements, the strain-displacement matrix $[\mathbf{B}]$ is not straightforward to assemble. In this case, it involves gradients and $\frac{\partial}{\partial x}$ is not simply a constant multiplied with a differential. To accomplish this, a function $\varphi = \varphi(\xi, \eta)$ is defined and we use the chain rule to differentiate this with respect to x and y as follows:

$$\frac{d\varphi}{d\xi} = \frac{d\varphi}{dx} \frac{dx}{d\xi} + \frac{d\varphi}{dy} \frac{dy}{d\xi} \quad 56$$

$$\frac{d\varphi}{d\eta} = \frac{d\varphi}{dx} \frac{dx}{d\eta} + \frac{d\varphi}{dy} \frac{dy}{d\eta} \quad 57$$

or as:

$$\begin{Bmatrix} \varphi_{,\xi} \\ \varphi_{,\eta} \end{Bmatrix} = [\mathbf{J}] \begin{Bmatrix} \varphi_{,x} \\ \varphi_{,y} \end{Bmatrix} \quad 58$$

where $[\mathbf{J}]$ is defined as the Jacobian such that:

$$[\mathbf{J}] = \begin{bmatrix} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n \frac{dN_i}{d\xi} x_i & \sum_{i=0}^n \frac{dN_i}{d\xi} y_i \\ \sum_{i=0}^n \frac{dN_i}{d\eta} x_i & \sum_{i=0}^n \frac{dN_i}{d\eta} y_i \end{bmatrix} \quad 59$$

and where n is equal to the number of nodes on the element. The Jacobian contains all first-order partial derivatives of the shape functions of an element. Equation 58 is reformulated to:

$$\begin{Bmatrix} \varphi_{,x} \\ \varphi_{,y} \end{Bmatrix} = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix} \begin{Bmatrix} \varphi_{,\xi} \\ \varphi_{,\eta} \end{Bmatrix} \quad 60$$

where $[\Gamma]$ equals the Jacobian inverse:

$$[\Gamma] = [\mathbf{J}]^{-1} = \frac{1}{J_{11} \cdot J_{22} - J_{12} \cdot J_{21}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \quad 61$$

The above transformation equations apply to plane elements that have any number of nodes.

By definition, we define the displacement over a Mindlin element by shape function

interpolation. As previously noted, there are three fields necessary to completely describe

the deformation. The lateral displacements and rotation fields are given by:

$$\begin{Bmatrix} w \\ \psi_x \\ \psi_y \end{Bmatrix} = \sum \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix} \begin{Bmatrix} w_i \\ \psi_{xi} \\ \psi_{yi} \end{Bmatrix} \quad \text{or} \quad \{\mathbf{u}\} = [\mathbf{N}]\{\mathbf{d}\} \quad 62$$

The shape functions $N_i = N_i(\xi, \eta)$ depend on the number of nodes in the element, they are given in Table 2 below for a quadrilateral element of up to nine nodes.

Table 2. Shape functions of a plane quadrilateral from 4 to 9 nodes (Cook, Malkus, Plesha, & Witt, 2002)

$N_i = N_i(\xi, \eta)$	Include only if node i is present in the element				
	$i=5$	$i=6$	$i=7$	$i=8$	$i=9$
$N_1 = \frac{1}{4}(1 - \xi)(1 - \eta)$	$-\frac{1}{2}N_5$			$-\frac{1}{2}N_8$	$-\frac{1}{4}N_9$
$N_2 = \frac{1}{4}(1 + \xi)(1 - \eta)$	$-\frac{1}{2}N_5$	$-\frac{1}{2}N_6$			$-\frac{1}{4}N_9$
$N_3 = \frac{1}{4}(1 + \xi)(1 + \eta)$		$-\frac{1}{2}N_6$	$-\frac{1}{2}N_7$		$-\frac{1}{4}N_9$
$N_4 = \frac{1}{4}(1 - \xi)(1 + \eta)$			$-\frac{1}{2}N_7$	$-\frac{1}{2}N_8$	$-\frac{1}{4}N_9$
$N_5 = \frac{1}{2}(1 - \xi^2)(1 - \eta)$					$-\frac{1}{4}N_9$
$N_6 = \frac{1}{2}(1 + \xi)(1 - \eta^2)$					$-\frac{1}{2}N_9$
$N_7 = \frac{1}{2}(1 - \xi^2)(1 + \eta)$					$-\frac{1}{2}N_9$
$N_8 = \frac{1}{2}(1 - \xi)(1 - \eta^2)$					$-\frac{1}{2}N_9$
$N_9 = (1 - \xi^2)(1 - \eta^2)$					

Nodes are placed at an elements vertices and along the midpoints. If present, node 9 is located at $\xi = \eta = 0$. The two plate quadrilateral elements implemented in Rts are shown in Figure 4 Below.

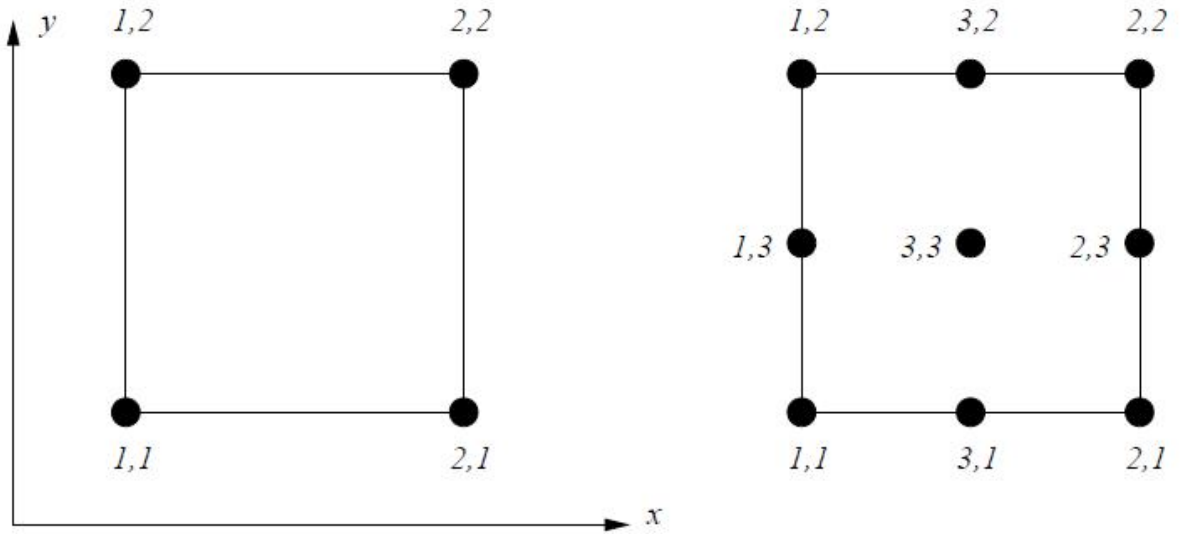


Figure 4. Four node and nine node quadrilateral elements.

The bilinear 4 node Mindlin (Mind4) is shown on the left while the 9 node (Mind9) Lagrange element is shown on the right. As introduced above in equation 43, the stiffness matrix is assembled using the strain-displacement interpolation matrix $[\mathbf{B}]$. Every element type has a unique strain-displacement interpolation matrix. To assemble the strain-displacement matrix for a Mindlin plate element, we must account for all three displacement fields: $w(x,y)$, ψ_x , and ψ_y . Again, these are the out-of-plane displacements and the rotations at the plate's midsurface and surface respectively.

Here we define the displacement $\{\mathbf{u}\} = [w \ \psi_x \ \psi_y]^T$ and curvature $\{\mathbf{k}_M\}$ as:

$$\{\mathbf{k}_M\} = \begin{Bmatrix} \psi_{x,x} \\ \psi_{y,y} \\ \psi_{x,y} + \psi_{y,x} \\ \psi_x - w_{,x} \\ \psi_y - w_{,y} \end{Bmatrix} = [\mathbf{d}]\{\mathbf{u}\} = [\mathbf{B}] \quad 63$$

In equation 63, the displacement vector $\{\mathbf{u}\}$ is approximated by the Mindlin shape functions $[\mathbf{N}]$ to give:

$$[\mathbf{B}] = [\mathbf{d}][\mathbf{N}] = \nabla[N] \quad 64$$

where \mathbf{d} is the derivative operator matrix known as the “nabla” operator ∇ .

$$[\mathbf{d}] = \begin{bmatrix} 0 & \partial/\partial x & 0 \\ 0 & 0 & \partial/\partial y \\ 0 & \partial/\partial y & \partial/\partial x \\ -\partial/\partial x & 1 & 0 \\ -\partial/\partial y & 0 & 1 \end{bmatrix} \quad 65$$

For an element having n nodes, we obtain $[\mathbf{B}]$ as:

$$[\mathbf{B}]_{5 \times 3n} = \begin{bmatrix} 0 & N_{1,x} & 0 & \cdots & 0 & N_{n,x} & 0 \\ 0 & 0 & N_{1,y} & \cdots & 0 & 0 & N_{n,y} \\ 0 & N_{1,y} & N_{1,x} & \cdots & 0 & N_{n,y} & N_{n,x} \\ -N_{1,x} & N_1 & 0 & \cdots & -N_{n,x} & N_n & 0 \\ -N_{1,y} & 0 & N_1 & \cdots & -N_{n,y} & 0 & N_n \end{bmatrix} \quad 66$$

We can now rewrite the element stiffness matrix in terms of the isoparametric coordinates as:

$$[\mathbf{K}] = \int_{-1}^{+1} \int_{-1}^{+1} [\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}] dV \quad 67$$

Using the definition of the Jacobian $[\mathbf{J}]$ from equation 59, its determinant $|\mathbf{J}|$ is used to formulate the change in volume of an element:

$$dV = |\mathbf{J}|d\xi d\eta \quad 68$$

Now the integral in equation 67 can be restated as:

$$[\mathbf{K}] = \iint_{-1}^{+1} [\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}] |\mathbf{J}| d\xi d\eta \quad 69$$

Using gauss quadrature numerical integration techniques, we calculate the local element stiffness matrix by summing the values at each integration point i, j :

$$[\mathbf{K}] = \sum_i \sum_j \alpha_{ij} [\mathbf{B}_{ij}]^T [\mathbf{D}] [\mathbf{B}_{ij}] |\mathbf{J}|_{ij} \quad 70$$

where

α_{ij} is the gauss integration weight coefficient,

$[\mathbf{B}]$ is the strain-displacement matrix,

$[\mathbf{D}]$ is the material properties matrix, and

$|\mathbf{J}|$ is the Jacobian determinant

Using the direct-stiffness method, the local element stiffness matrices are compiled into the global stiffness matrix. Incorporating the forces acting on the system, we are left with the final form of the global element stiffness relation:

$$\{\mathbf{F}\} = [\mathbf{K}]\{\mathbf{u}\} \quad 71$$

Finally, the system of equations is solved for the displacements:

$$\{\mathbf{u}\} = [\mathbf{K}]^{-1}\{\mathbf{F}\} \quad 72$$

The quadrilateral four node plate element used in Rts is shown below in Figure 5 with its 12 dof.

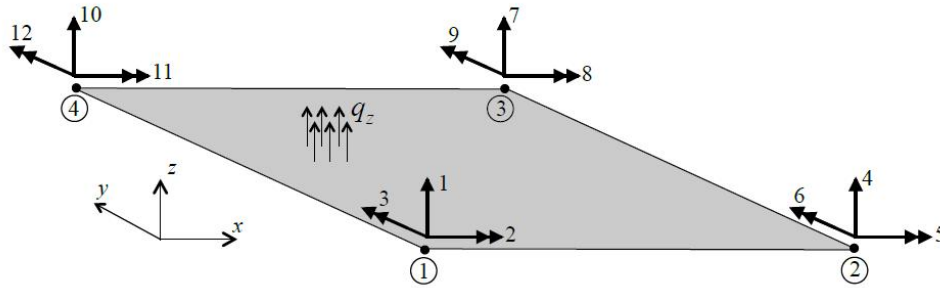


Figure 5. Four node quadrilateral plate element

In Rts, every node has 6 dof though plate elements only make use of 3 dofs in their stiffness assembly as shown above.

Chapter 3: ANALYTICAL GRADIENTS

Gradients are important throughout the model stream since almost all optimization algorithms throughout the orchestrating models involve the gradients in one form or another. In addition, the gradients provide us with sensitivity measures of the design parameters. This information is helpful in deciding which design variables to vary to achieve the lowest overall cost with all things considered. The gradient flow chart throughout the models is given as:

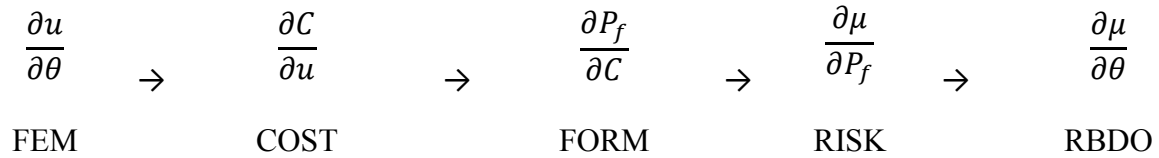


Figure 6. The gradient flowchart within Rts

The overall intent is to calculate the sensitivity of the mean cost in relation to a change in the design parameters θ . In other words, how much will the mean cost change when we vary the values of the design parameters. This information is directly used by the optimization algorithm since the minimums occur at stationary points where the gradient of the objective function is equal to zero ($\nabla(S\mathbb{O}) = 0$). Note that for constrained problems, the local minimum may not be located at a stationary point if it does not fall within the constraint boundaries; the local minimums could potentially be located on the boundary itself. For this thesis I will focus on unconstrained optimization.

Before this thesis, Rts had an initial framework in place for calculating the sensitivities as shown in the graphic below.

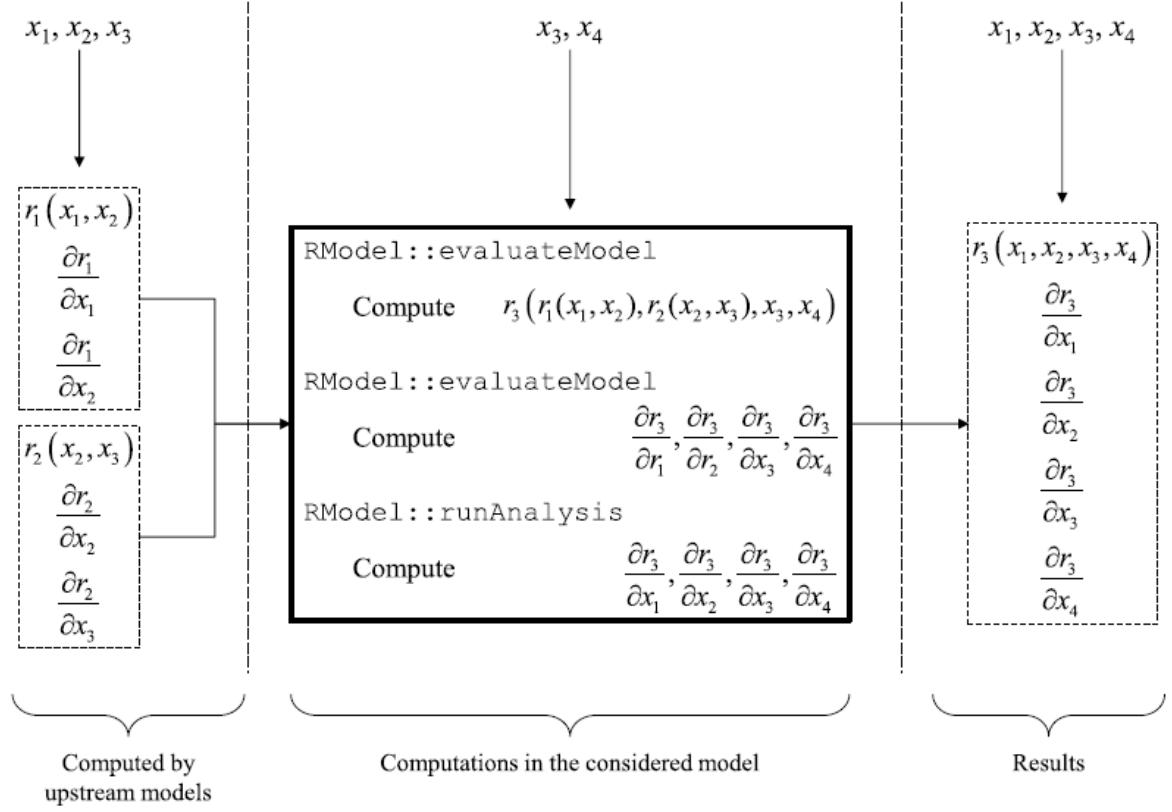


Figure 7. Direct Differentiation Method (DDM) in Rts (Mahsuli & Haukaas, 2013)

Figure 7 outlines the gradient computational stream for a response r_3 with respect to parameters $x_1 \cdots x_4$. In this case, the response depends explicitly on parameters x_3 and x_4 and implicitly on parameters x_1 and x_2 . The sensitivities for upstream models are calculated previously and stored in each model's `RResponse` class. In the current orchestrating model, the analytical expressions for the derivatives are required for all parameters. For this example, analytical expressions for $\frac{\partial r_3}{\partial r_1}$, $\frac{\partial r_3}{\partial r_2}$, $\frac{\partial r_3}{\partial x_3}$, $\frac{\partial r_3}{\partial x_4}$ are computed within the orchestrating model's class whose input responses are r_1 and r_2 and whose output response is r_3 . The result is then inserted into the `RModel` DDM Map; `RModel` is the parent

class for all orchestrating models and the variable DDMMMap is one of its members. After the orchestrating model is evaluated, the gradient ‘dependency’ calculations are performed within the RModel class. Here the implicit and explicit dependencies between DDMMMap of the current orchestrating model $\left(\frac{\partial r_3}{\partial r_1}, \frac{\partial r_3}{\partial r_2}, \frac{\partial r_3}{\partial x_3}, \frac{\partial r_3}{\partial x_4}\right)$ and the gradient maps of the input models $\left(\frac{\partial r_1}{\partial x_1}, \frac{\partial r_1}{\partial x_2}\right)$ and $\left(\frac{\partial r_2}{\partial x_2}, \frac{\partial r_2}{\partial x_3}\right)$ are evaluated. The calculation performed is:

$$\frac{\partial r_3}{\partial x_1} = \frac{\partial r_3}{\partial r_1} \cdot \frac{\partial r_1}{\partial x_1} \quad 73$$

$$\frac{\partial r_3}{\partial x_2} = \frac{\partial r_3}{\partial r_1} \cdot \frac{\partial r_1}{\partial x_2} + \frac{\partial r_3}{\partial r_2} \cdot \frac{\partial r_2}{\partial x_2} \quad 74$$

$$\frac{\partial r_3}{\partial x_3} = \frac{\partial r_3}{\partial r_2} \cdot \frac{\partial r_2}{\partial x_3} + \frac{\partial r_3}{\partial x_3} \quad 75$$

$$\frac{\partial r_3}{\partial x_4} = \frac{\partial r_3}{\partial x_4} \quad 76$$

The final results are then saved as a gradient map to the RResponse class of the orchestrating model, for use in models downstream.

Gradients can be calculated both numerically and analytically. The finite difference method is a numerical method used for approximating the solutions to differential equations by employing finite difference equations. Finite difference equations are used to replace, or more generally approximate, the derivatives of a differential equation by central, forward or backward difference(s). By definition, the finite difference method is not exact; it is only an approximation. Therefore, there will always be an error which is the difference between the

approximation and the exact analytical solution. The two main sources of error are round-off and discretization error. Round-off errors are the loss of precision due to computer rounding of decimals while discretization errors, or truncation errors, inherently come from the discretization of the problem's domain and the continuous applications of the finite difference method.

A drawback of the finite difference method is that it requires additional simulations and computations each time a design parameter is perturbed. Also, selecting the optimal perturbation size of a set of parameters with the sensitivity of the numerical solution in context can be very difficult and important (Anderson, 2000). Additionally, when performing nodal optimization, every perturbation requires re-meshing of the FEM domain. Therefore, the disadvantage of the finite difference method is that it can be computationally expensive while still incorporating errors.

Conversely, analytical gradients are exact. They are calculated from the direct differentiation of the functions used throughout the various models. Exact gradients are useful because they ensure consistent convergence; especially when the analysis is near the stationary point of the optimization where accurate gradients are required. Implementation of DDM comes with the initial cost of calculating the analytical derivatives and implementing them in the code alongside the orchestrating models. This is not a trivial task due to the complexity of the models and their interconnectivity. The sections below will go over the DDM gradient theory, calculations, and their logistics throughout each model.

3.1 FEM Gradients

The FEM analysis is the starting point of the model analysis stream. The inputs into the FEM models are the design parameters θ . The design parameters could be decision variables, random variables, responses for nodal coordinates, and/or constants. The outputs of the FEM are the displacement responses $\{\mathbf{u}\}$. Recall from equation 72 that the final formulation of the FEM system of equations is:

$$\{\mathbf{u}\} = [\mathbf{K}]^{-1}\{\mathbf{F}\} \quad 77$$

We start with the derivation of $\frac{\partial \mathbf{u}}{\partial \theta}$ where it is required to find the gradient of the stiffness matrix $\frac{d\mathbf{K}}{d\theta}$, and $\frac{\partial \mathbf{F}}{\partial \theta}$ as shown in the following equation:

$$\frac{\partial \mathbf{u}}{\partial \theta} = \mathbf{K}^{-1} \underbrace{\left(\frac{\partial \mathbf{F}}{\partial \theta} - \frac{\partial \mathbf{K}}{\partial \theta} \cdot \mathbf{u} \right)}_{\text{pseudo load}} \quad 78$$

where $\frac{\partial u}{\partial \theta}$ is the gradient of the displacements u with respect to the parameters θ . The expression within the brackets is termed the “pseudo-load.” Recall that the stiffness matrix from equation 70 is:

$$\mathbf{K} = \sum \sum \alpha \mathbf{B}^T \mathbf{D} \mathbf{B} \quad 79$$

where $\alpha = dvol = |\mathbf{J}|w_iw_j$. α can be considered a constant for material sensitivity (E,v,h etc.) though it is NOT a constant for nodal or shape sensitivity; this is because α contains the Jacobian which is a function of the nodal coordinates. Therefore, without shape sensitivity (when not considering nodal coordinate variations) only material sensitivity is considered.

This simplifies the sensitivity expression to:

$$\frac{d\mathbf{K}}{d\theta} = \alpha \frac{d\{\mathbf{B}^T \mathbf{D} \mathbf{B}\}}{d\theta} \quad 80$$

Let

$$\frac{d}{d\theta}\{\mathbf{DB}\} = \frac{d\mathbf{D}}{d\theta}\mathbf{B} + \mathbf{D}\frac{d\mathbf{B}}{d\theta} \quad 81$$

and let

$$\frac{d}{d\theta}\{\mathbf{B}^T\} = \left\{\frac{d\mathbf{B}}{d\theta}\right\}^T \quad 82$$

now

$$\frac{d\mathbf{K}}{d\theta} = \alpha \left[\left(\frac{d\mathbf{B}}{d\theta}\right)^T \mathbf{DB} + \mathbf{B}^T \left\{ \frac{d\mathbf{D}}{d\theta} \mathbf{B} + \mathbf{D} \frac{d\mathbf{B}}{d\theta} \right\} \right] \quad 83$$

which gives

$$\frac{d\mathbf{K}}{d\theta} = \alpha \left[\left(\frac{d\mathbf{B}}{d\theta}\right)^T \mathbf{DB} + \mathbf{B}^T \frac{d\mathbf{D}}{d\theta} \mathbf{B} + \mathbf{B}^T \mathbf{D} \frac{d\mathbf{B}}{d\theta} \right] \quad 84$$

For material sensitivities, the derivative of $\frac{d\mathbf{B}}{d\theta}$ equals zero since the strain displacement matrix is not a function of any material properties. This simplifies to the final expression for material sensitivities:

$$\frac{d\mathbf{K}}{d\theta} = \alpha \left[\mathbf{B}^T \frac{d\mathbf{D}}{d\theta} \mathbf{B} \right] \quad 85$$

Therefore, for material sensitivity, the gradient of the stiffness matrix is derived solely from the strain displacement and material matrices for material sensitivity. For shape sensitivity, equation 84 is extended to:

$$\frac{d\mathbf{K}}{d\theta} = w_i w_j |\mathbf{J}| \left[\text{Tr} \left\{ \mathbf{J}^{-1} \frac{\partial \mathbf{J}}{\partial \theta} \right\} \mathbf{B}^T \mathbf{DB} + \left(\frac{d\mathbf{B}}{d\theta}\right)^T \mathbf{DB} + \mathbf{B}^T \mathbf{D} \frac{d\mathbf{B}}{d\theta} \right] \quad 86$$

where

$$\frac{\partial |\mathbf{J}|}{\partial \theta} = |\mathbf{J}| \text{Tr} \left\{ \mathbf{J}^{-1} \frac{\partial \mathbf{J}}{\partial \theta} \right\} \quad 87$$

The derivative of the Jacobian is incorporated as well as the derivative of the strain-

displacement matrix **B**. Recall that the strain-displacement matrix from equation 66 is:

$$\mathbf{B}_{5 \times 3n} = \begin{bmatrix} 0 & N_{1,x} & 0 & \cdots & 0 & N_{n,x} & 0 \\ 0 & 0 & N_{1,y} & \cdots & 0 & 0 & N_{n,y} \\ 0 & N_{1,y} & N_{1,x} & \cdots & 0 & N_{n,y} & N_{n,x} \\ -N_{1,x} & N_1 & 0 & \cdots & -N_{n,x} & N_n & 0 \\ -N_{1,y} & 0 & N_1 & \cdots & -N_{n,y} & 0 & N_n \end{bmatrix}$$

The **B** matrix already involves the derivatives of the shape functions. For nodal sensitivities, we require the second derivatives, which depend on the parameter that we are taking the derivative with respect to. The formula for the derivative of the strain-displacement matrix is:

$$\frac{d\mathbf{B}}{d\theta} = \frac{\partial \nabla \mathbf{N}}{\partial \theta} \quad 88$$

In the case of nodal sensitivities, for the 2-D plane element, the parameters θ can either be x or y . That is, when nodal coordinates depend on a geometric parameter that we need sensitivities for, such as plate width or length, for example. Recall that the Jacobian is given as:

$$[\mathbf{J}] = \begin{bmatrix} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n \frac{dN_i}{d\xi} x_i & \sum_{i=0}^n \frac{dN_i}{d\xi} y_i \\ \sum_{i=0}^n \frac{dN_i}{d\eta} x_i & \sum_{i=0}^n \frac{dN_i}{d\eta} y_i \end{bmatrix} \quad 89$$

The derivative of the Jacobian is therefore:

$$\frac{\partial [\mathbf{J}]}{\partial \theta} = \begin{bmatrix} \sum_{i=0}^n \frac{dN_i}{d\xi} \frac{\partial x_i}{\partial \theta} & \sum_{i=0}^n \frac{dN_i}{d\xi} \frac{\partial y_i}{\partial \theta} \\ \sum_{i=0}^n \frac{dN_i}{d\eta} \frac{\partial x_i}{\partial \theta} & \sum_{i=0}^n \frac{dN_i}{d\eta} \frac{\partial y_i}{\partial \theta} \end{bmatrix} \quad 90$$

From theorems of matrix calculus, it can be shown that the inverse is:

$$\left[\frac{\partial [\mathbf{J}]}{\partial \boldsymbol{\theta}} \right]^{-1} = [\mathbf{J}]^{-1} \cdot \frac{\partial [\mathbf{J}]}{\partial \boldsymbol{\theta}} \cdot [\mathbf{J}]^{-1} \quad 91$$

The derivative of \mathbf{B} is made up of the components of the differentiated inverse Jacobian to yield:

$$\frac{\partial \mathbf{B}_{5 \times 3n}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} 0 & N_{1,x,\boldsymbol{\theta}} & 0 & \cdots & 0 & N_{n,x,\boldsymbol{\theta}} & 0 \\ 0 & 0 & N_{1,y,\boldsymbol{\theta}} & \cdots & 0 & 0 & N_{n,y,\boldsymbol{\theta}} \\ 0 & N_{1,y,\boldsymbol{\theta}} & N_{1,x,\boldsymbol{\theta}} & \cdots & 0 & N_{n,y,\boldsymbol{\theta}} & N_{n,x,\boldsymbol{\theta}} \\ -N_{1,x,\boldsymbol{\theta}} & 0 & 0 & \cdots & -N_{n,x,\boldsymbol{\theta}} & 0 & 0 \\ -N_{1,y,\boldsymbol{\theta}} & 0 & 0 & \cdots & -N_{n,y,\boldsymbol{\theta}} & 0 & 0 \end{bmatrix} \quad 92$$

where

$$\begin{Bmatrix} N_{x,\boldsymbol{\theta}} \\ N_{y,\boldsymbol{\theta}} \end{Bmatrix} = \left[\frac{\partial [\mathbf{J}]}{\partial \boldsymbol{\theta}} \right]^{-1} \cdot \begin{Bmatrix} N_{\xi} \\ N_{\eta} \end{Bmatrix} \quad 93$$

3.2 COST Model Gradients

The next step in the model stream is the calculation of the cost model gradients. The cost models may consist of construction, damage, and degradation models. The input to the cost model consists of the displacement response output from FEM along with other random variables that represent cost measures. The cost model can be a simple algebraic expression or a complex relationship that is a function of the displacement, design variables, and/or time. In any case, the output of the cost model is related to the displacement output from the FEM model, which is in turn related to the design variables. In this instance, the cost is given by:

$$C = c(u(\boldsymbol{\theta})) + d(\boldsymbol{\theta}) \quad 94$$

where $c(u(\boldsymbol{\theta}))$ is a function that assigns a monetary value to a given displacement $u(\boldsymbol{\theta})$ and

$d(\theta)$ is a function of a design variable (density or beam geometry, for example). The gradient of the cost model is hence given by:

$$\frac{\partial C}{\partial \theta} = \frac{\partial C}{\partial u} \cdot \frac{\partial u}{\partial \theta} + \frac{\partial C}{\partial d} \cdot \frac{\partial d}{\partial \theta} \quad 95$$

The gradients for $\frac{\partial C}{\partial \theta}$ must be analytically solved within the cost model. Cost and damage models can be highly complex; as such, care must be taken so that the gradients are smooth-continuous e.g. there are no large jumps in the costs that cause “kinks” or discontinuities within the gradient curve.

3.3 FORM Reliability Model Gradients

The gradient of the limit state function is necessary for the gradient-based optimization methods used in FORM. The gradient of the limit-state function is given by:

$$\nabla G(\mathbf{y}) = \frac{\partial G(\mathbf{y})}{\partial \mathbf{y}} = \frac{\partial G}{\partial u} \frac{\partial u}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \mathbf{y}} \quad 96$$

where

$\frac{\partial G}{\partial u}$ is the derivative of the limit state algebraic expression which is in terms of the response $u(\mathbf{X})$,

$\frac{\partial u}{\partial \mathbf{X}}$ is the response gradient vector calculated from the FEM model, and

$\frac{\partial \mathbf{X}}{\partial \mathbf{y}}$ is the inverse of the Jacobian matrix from the probability transformation composed of the components

$$\frac{\partial y_i}{\partial x_i} = \frac{f_i(x_i)}{\varphi(y_i)} \quad 97$$

where f and φ are the CDF and PDF corresponding to F and Φ . \mathbf{X} is the vector of random

variables in original space while \mathbf{y} is its transformation in standard-normal space. The input into the reliability model is the cost response. The output is the probability of exceedance of a failure, or in this case the total cost threshold C_0 . The derivatives of P_f and β with respect to the parameters θ are valuable because they are used in the gradient-based optimization algorithms; analytical derivatives are particularly useful since they are exact and offer faster and more reliable convergence. In addition, the gradients are useful in themselves as “importance measures.” In FORM, the alpha vector serves as an indicator or importance vector in where the absolute value of each individual component serves as a relative indicator of its contribution to the total variance. Recall that the limit-state function in original space is given by:

$$g(\mathbf{X}) = C_0 - C(\mathbf{X}) \quad 98$$

and in standard-normal space by:

$$G(\mathbf{y}) = C_0 - C(\mathbf{y}) \quad 99$$

In FORM, the limit-state function is linearized by a first-order Taylor approximation to produce:

$$G(\mathbf{y}) \approx G(\mathbf{y}^*) + \nabla G \mathbf{y}^{*T} \cdot (\mathbf{y} - \mathbf{y}^*) \quad 100$$

where \mathbf{y} is the random variable vector in standard-normal space and \mathbf{y}^* contains the coordinates of the design point. At the design point, the gradient of the limit-state function is found by using the chain rule:

$$\nabla G(\mathbf{y}^*) = \frac{\partial G}{\partial \mathbf{y}^*} = \frac{\partial g}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{X}^*} \cdot \frac{\partial \mathbf{X}^*}{\partial \mathbf{y}^*} \quad 101$$

Also, recall from equation 9 that the probability P_f is:

$$P_f(\mathbf{X}) = P(g(\mathbf{X}) \leq 0) = \Phi(-\beta) \quad 102$$

In other words, P_f is the probability that the cost $C(\mathbf{X})$ equals or exceeds a threshold C_0 ; where the limit-state function is less than or equal to zero. The vector \mathbf{X} consists of the random variables from the design parameters θ in original space. It follows that the derivative of the exceedance probability with respect to the cost threshold is:

$$\frac{\partial P_f}{\partial C_0} = \frac{\partial \Phi(-\beta)}{\partial C_0} = \frac{\partial \beta}{\partial C_0} \frac{\partial \Phi(-\beta)}{\partial \beta} = -\frac{\partial \beta}{\partial C_0} \cdot \varphi(\beta) \quad 103$$

$\frac{\partial \beta}{\partial C_0}$ can be calculated by realizing that the reliability index β is directly related to the design point values at \mathbf{y}^* , such as:

$$\frac{\partial \beta}{\partial C_0} = \frac{\partial \beta^T}{\partial \mathbf{y}^*} \frac{\partial \mathbf{y}^*}{\partial C_0} \quad 104$$

The reliability index can be stated as $\beta = \boldsymbol{\alpha}^T \mathbf{y}^*$, the derivative of which is:

$$\frac{\partial \beta}{\partial \mathbf{y}^*} = \boldsymbol{\alpha} = -\frac{\nabla G^T(\mathbf{y}^*)}{\|\nabla G(\mathbf{y}^*)\|} \quad 105$$

Hence, $\frac{\partial \beta}{\partial C_0}$ is given with the following expression:

$$\frac{\partial \beta}{\partial C_0} = -\frac{\nabla G^T(\mathbf{y}^*)}{\|\nabla G(\mathbf{y}^*)\|} \frac{\partial \mathbf{y}^*}{\partial C_0} \quad 106$$

We must now differentiate the limit-state function at the design point \mathbf{y}^* , where $\mathbf{G} = \mathbf{0}$.

$$\frac{\partial G}{\partial C_0} = \frac{\partial G}{\partial \mathbf{y}^*} \frac{\partial \mathbf{y}^*}{\partial C_0} + \frac{\partial G}{\partial C_0} \Big|_{\mathbf{y}^*} = \nabla G^T(\mathbf{y}^*) \frac{\partial \mathbf{y}^*}{\partial C_0} + \frac{\partial G}{\partial C_0} \Big|_{\mathbf{y}^*} = 0 \quad 107$$

Solving for $\frac{\partial \mathbf{y}^*}{\partial C_0}$ at the design point gives the conditional derivative:

$$\frac{\partial \mathbf{y}^*}{\partial C_0} \Big|_{\mathbf{y}^*} = -\nabla G^T(\mathbf{y}^*) \frac{\partial \mathbf{y}^*}{\partial C_0} \quad 108$$

Substituting the conditional derivative into equation 106 gives:

$$\frac{\partial \beta}{\partial C_o} = \|\nabla G(\mathbf{y}^*)\|^{-1} \frac{\partial G}{\partial C_o} \Big|_{\mathbf{y}^*} \quad 109$$

It can be proven that the derivatives of the reliability index at the design point in standard-normal and in original space are related as follows:

$$\frac{\partial \beta}{\partial C_o} = \|\nabla G(\mathbf{y}^*)\|^{-1} \frac{\partial G}{\partial C_o} \Big|_{\mathbf{y}^*} = \|\nabla G(\mathbf{y}^*)\|^{-1} \frac{\partial g}{\partial C_o} \Big|_{\mathbf{x}^*} \quad 110$$

$\frac{\partial g}{\partial C_o}$ is the derivative of the original limit state function ($g(\mathbf{X}) = C_o - C(\mathbf{X})$), which is equal

to 1. Substituting equation 110 into 103 gives the reliability gradient $\frac{\partial P_f}{\partial C_o}$ as:

$$\frac{\partial P_f}{\partial C_o} = \frac{-\varphi(\beta)}{\|\nabla G(\mathbf{y}^*)\|} \quad 111$$

If we want the derivative of the reliability index with respect to parameters θ , the following expression is derived:

$$\frac{\partial \beta}{\partial \theta} = - \frac{\nabla G^T(\mathbf{y}^*)}{\|\nabla G(\mathbf{y}^*)\|} \frac{\partial \mathbf{y}^*}{\partial \theta} \quad 112$$

In other words, this relates to how much the distance from failure plane to origin changes in relation to changes in the design variables.

3.4 RISK Model Gradients

The risk model takes the FORM model probability of exceedance P_f as an input and outputs the RISK measure μ to the optimization model. The FORM output $P_f(g(C(\mathbf{X}) \geq C_o))$ yields a single point on the Complementary Cumulative Density Function (CCDF) curve for the response $C(\mathbf{X})$. A method of acquiring the mean of cost $C(\mathbf{X})$ is by calculating the area under the curve. In this case, the mean is calculated by numerically integrating the

CCDF values using the trapezoidal rule, formulated as:

$$\mu = \frac{1}{2} \left\{ c_0 (1 + P_{f_0}) + \sum_{i=0}^n \left\{ (c_{i+1} - c_i) \cdot (P_{f_{i+1}} + P_{f_i}) \right\} \right\} \quad 113$$

The gradient required for optimization is the sensitivity of the mean with regards to any changes in the parameters. Since μ is a function of both C_o and P_f , the chain rule is used to derive the expression for the risk sensitivities:

$$\frac{\partial \mu}{\partial \theta} = \frac{1}{2} \left\{ \frac{\partial c_0}{\partial \theta} (1 + P_{f_0}) + c_0 \cdot \frac{\partial P_{f_0}}{\partial \theta} + \sum_{i=0}^n \left\{ \left(\frac{\partial c_{i+1}}{\partial \theta} - \frac{\partial c_i}{\partial \theta} \right) \cdot (P_{f_{i+1}} + P_{f_i}) + (c_{i+1} - c_i) \cdot \left(\frac{\partial P_{f_{i+1}}}{\partial \theta} + \frac{\partial P_{f_i}}{\partial \theta} \right) \right\} \right\} \quad 114$$

where

$\frac{\partial c_o}{\partial \theta}$ equals the gradients from the COST model evaluated at the design point, and

$\frac{\partial P_f}{\partial \theta} = \frac{\partial P_f}{\partial c_o} \cdot \frac{\partial c_o}{\partial \theta}$ where $\frac{\partial P_f}{\partial c_o}$ is equal to the FORM gradient at the design point derived above in

equation 111. The final expression for the RISK gradient is equal to:

$$\begin{aligned} \frac{\partial \mu}{\partial \theta} = & \frac{1}{2} \left\{ \frac{\partial c_0}{\partial \theta} (1 + P_{f_0}) + c_0 \cdot \frac{\partial P_f}{\partial c_o} \frac{\partial c_o}{\partial \theta} \right. \\ & \left. + \sum_{i=0}^n \left\{ \left(\frac{\partial c_{i+1}}{\partial \theta} - \frac{\partial c_i}{\partial \theta} \right) \cdot (P_{f_{i+1}} + P_{f_i}) + (c_{i+1} - c_i) \cdot \left(\frac{\partial P_{f_{i+1}}}{\partial c_{c_{i+1}}} \frac{\partial c_{i+1}}{\partial \theta} + \frac{\partial P_{f_i}}{\partial c_i} \frac{\partial c_i}{\partial \theta} \right) \right\} \right\} \end{aligned} \quad 115$$

3.5 RBDO Gradients

The analytical gradients used in the optimization analysis can be calculated using the Direct and Adjoint Differentiation Methods; DDM and ADM respectively. Currently, only the DDM method is employed in Rts but the related ADM is given for completeness. The DDM and ADM methods are explained in the sections below.

3.5.1 Direct Differentiation Method – DDM

The sensitivity of the objective function $\hat{g} = (x^i, u)$ is given as:

$$\frac{\partial \hat{g}(x^i)}{\partial \theta_j} = \frac{\partial F(x^i)^T}{\partial \theta_j} u(x^i) + F(x^i) \cdot K(x^i)^{-1} \cdot \underbrace{\left(\frac{\partial F(x^i)}{\partial \theta_j} - \frac{\partial K(x^i)}{\partial \theta_j} \cdot u(x^i) \right)}_{\text{pseudo-load}} \quad 116$$

Therefore at each design iteration point, to find the gradient of the objective function, we need to solve the “pseudo-load” for each parameter j , which is then inserted into each constraint function g_i ($i = 1 \dots l$). Thus, the total number of calculations is then $j \times l$.

3.5.2 Adjoint Method – ADM

Using the adjoint method, we begin with:

$$K(x^k) \cdot \lambda = \left(\frac{\partial g_i}{\partial u} \right)^T \quad 117$$

Solving for λ gives:

$$\lambda = K(x^k)^{-1} \left(\frac{\partial g_i}{\partial u} \right)^T \quad 118$$

The sensitivity is then given as:

$$\frac{\partial \hat{g}(x^k)}{\partial \theta_j} = \frac{\partial g_i}{\partial \theta_j} + \lambda^T \left(\frac{\partial F(x^k)}{\partial \theta_j} - \frac{\partial K(x^k)}{\partial \theta_j} \cdot u(x^k) \right) \quad 119$$

Using this approach, λ is solved for each constraint function g_i ($i = 1 \dots l$), and then it is inserted into the final sensitivity equation l times. In this case, the total number of calculations performed is $l \times l$. To conclude, the adjoint method is more efficient if the number of constraints is more than the number of design variables. Conversely, if the design variables outnumber the constraints, the direct differentiation method is more effective (Christensen & Klarbring, 2009).

Chapter 4: IMPLEMENTATION AND EXAMPLES IN RTS

This chapter outlines the implementation of the orchestrating models and their respective DDM gradient calculations in Rts for performing reliability-based optimization. The model architecture for a typical analysis is given in Figure 8:

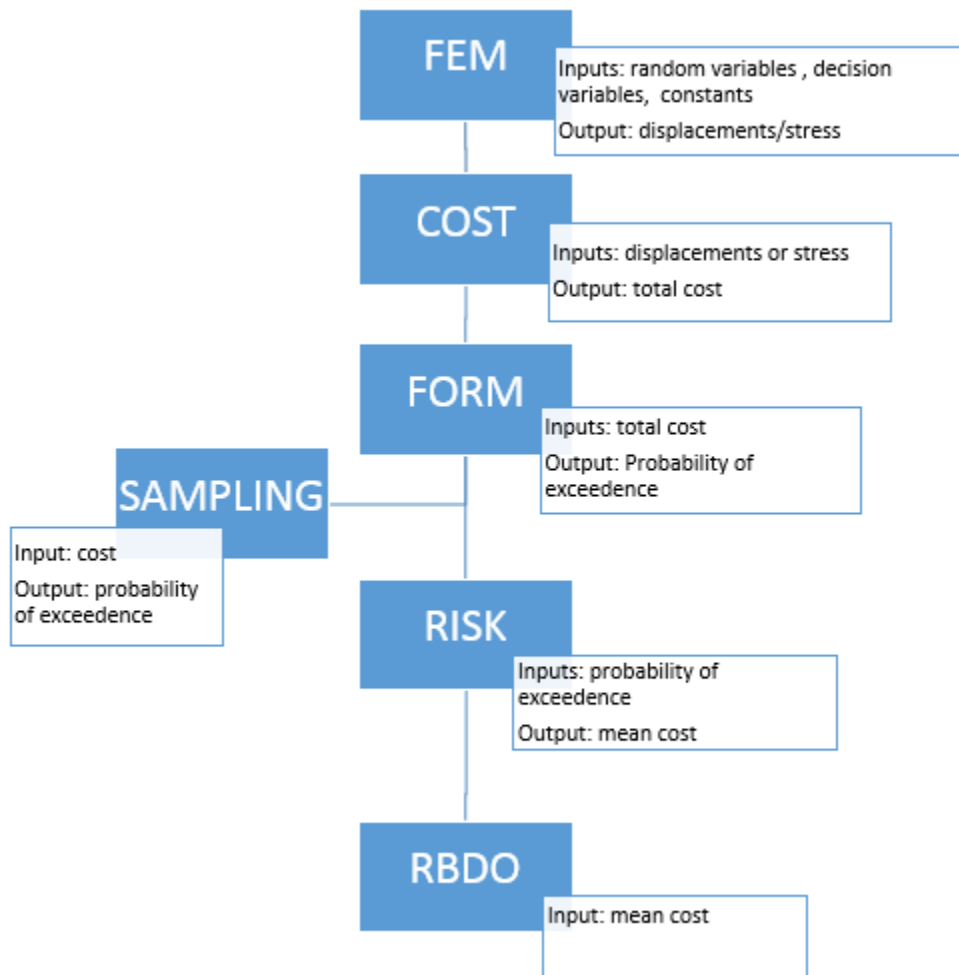


Figure 8. Optimization software architecture in Rts

The RBDO is the high-level orchestrating model that calls the run analysis method in RFunction starting with the down-stream FEM model first.

4.1 FEM Implementation

At the FEM level, parameters consist of material properties, section properties, coordinate properties, or loads and hazards. Table 3 below provides typical parameters used in Rts.

Table 3. Examples of typical FEM parameters used in Rts

Material Properties	Young's Modulus, Poisson's Ratio, Density
Section Properties	Member Thickness, Member Length/Width, Moment of Inertia
Points or Coordinates	Any point or node on the structure
Force Components	Parameters or force components that contribute to the response

Any parameter that contributes to the response can be analytically differentiated. The response from the FEM model is always a displacement which can be translated into a stress, force, or strain. The software architecture is setup so that the analytical derivative equations are programmed in on the element level. The stiffness matrix assembly routine then compiles the differentiated global sensitivity-stiffness matrix corresponding to each parameter. The sensitivity values for all parameters are solved and then saved in the RNode class for each node. When the user selects a response, the gradient values from that response's node are then utilized for calculating gradients downstream of FEM. For the thesis examples, the parameters used in FEM were:

$$\frac{du}{d\theta} = \left\{ \frac{du}{dE} \quad \frac{du}{dh} \quad \frac{du}{db} \quad \frac{du}{dF} \right\}^T$$

where E, h, b, F are the young's modulus, thickness, width, and a point load respectively.

4.2 COST Implementation

In this example implementation, the cost model involves only a simple algebraic expression:

$$C(\theta) = \theta_1 \cdot b \cdot h \cdot L + \theta_2 \cdot u \quad 120$$

where

θ_1 = a random variable that represents the cost of purchasing material and constructing a timber panel \$/m³,

b = the width of the panel in meters,

h = the thickness of the panel in meters,

L = the length of the panel in meters,

θ_2 = a random variable that represents the cost of damage due to the maximum deflection of the beam under a point load at its tip \$/m, and

$u(\theta)$ = the deflection of the tip of the beam.

The gradients inserted in RCostModel DDMMMap are:

$$\frac{dC}{d\theta_1} = b \cdot h \cdot L \quad 121$$

$$\frac{dC}{db} = \theta_1 \cdot h \cdot L \quad 122$$

$$\frac{dC}{dh} = \theta_1 \cdot b \cdot L \quad 123$$

$$\frac{dC}{dL} = \theta_1 \cdot b \cdot h \quad 124$$

$$\frac{dC}{d\theta_2} = u \quad 125$$

$$\frac{dC}{du} = \theta_2 \quad 126$$

The COST DDMMMap dependencies are then evaluated in RModel. From FEM, the gradients were:

$$\frac{du}{d\theta} = \left\{ \frac{du}{dE} \quad \frac{du}{dh} \quad \frac{du}{db} \quad \frac{du}{dF} \right\}^T \quad 127$$

Since the only parameters shared by the input and cost models are the thickness and beam width, the dependency calculations are:

$$\begin{aligned} \frac{dC}{dh} &= \frac{dC}{dh} + \frac{dC}{du} \frac{du}{dh} \\ \frac{dC}{db} &= \frac{dC}{db} + \frac{dC}{du} \frac{du}{db} \end{aligned} \quad 128$$

Giving the final cost gradient vector as:

$$\frac{dC}{d\theta} = \left\{ \frac{dC}{d\theta_1} \quad \frac{dC}{db} \quad \frac{dC}{dh} \quad \frac{dC}{dL} \quad \frac{dC}{d\theta_2} \quad \frac{dC}{du} \right\}^T \quad 129$$

4.3 RISK Implementation

The RISK model incorporates the FORM and sampling models. For gradient calculations, the DDM Map of each iteration of FORM at each cost threshold calculation is saved at convergence and utilized in calculating the final RISK gradient. Unlike the previous dependency calculations, the RISK model gradient calculations are done within the model itself. They are then directly saved into the RResponse class from within RISK. The dependency calculation within RModel is avoided in this step since RModel would only see the final COST response gradients from RResponse that are leftover from the last FORM iteration at the final threshold.

4.3.1 FORM and COST Gradients

Within RISK, at each cost threshold FORM convergence, the COST model gradients are saved along with the FORM gradient as described below. The FORM gradients utilized are described in section 3.3, equation 111 as:

$$\frac{\partial P_f}{\partial C_o} = \frac{-\varphi(\beta)}{\|\nabla G\|} \quad 130$$

The COST gradients are saved at the FORM design point, or when FORM converges. At this instance the realization of the cost function equals the cost threshold. Mathematically, this is simply:

$$\frac{\partial C}{\partial \theta} = \frac{\partial C_o}{\partial \theta} \quad 131$$

since $C = C_o$. These gradients are used in RISK to calculate the sensitivity of the mean.

4.3.2 RISK Sensitivity Calculations

The RISK gradients involve the derivatives of the Trapezoidal Rule at each exceedance probability/cost threshold pair. Since we are performing a summation to calculate the mean, our gradients must equally be accounted for. For example, we cannot solely use the gradients of the last value of the sum as RModel would, all of them must be represented equally. To accomplish this, we sum the products of the gradients within RISK to achieve the final RISK sensitivity. For each parameter, the DDM map $\frac{\partial C_o}{\partial \theta}$ from every COST threshold is multiplied through with the FORM gradient and summed up. The final RISK gradients are then saved to the global DDM Map.

This calculation is given as:

$$\begin{aligned} \frac{\partial \mu}{\partial \theta} = & \frac{1}{2} \left\{ \frac{\partial c_0}{\partial \theta} (1 + P_{f_0}) + c_0 \cdot \frac{\partial P_f}{\partial c_0} \frac{\partial c_0}{\partial \theta} \right. \\ & \left. + \sum_{i=0}^n \left\{ \left(\frac{\partial c_{i+1}}{\partial \theta} - \frac{\partial c_i}{\partial \theta} \right) \cdot (P_{f_{i+1}} + P_{f_i}) + (c_{i+1} - c_i) \cdot \left(\frac{\partial P_{f_{i+1}}}{\partial c_{i+1}} \frac{\partial c_{i+1}}{\partial \theta} + \frac{\partial P_{f_i}}{\partial c_i} \frac{\partial c_i}{\partial \theta} \right) \right\} \right\} \end{aligned} \quad 132$$

4.3.3 SAMPLING Gradients

The sampling model is only used as a tool to establish the approximate thresholds on either side of the estimated mean. Therefore, the sampling gradients are of no use.

4.4 RBDO Implementation

For this thesis, the optimization algorithm was implemented using the steepest descent method and a fixed step size utilizing the RISK gradients mentioned above. Note that for optimization routines involving nodal coordinates, the FEM model needs to be re-meshed at every perturbation of the decision variable.

Chapter 5: RBDO EXAMPLES IN RTS

The three examples implemented in Rts involve the optimization of the thickness and width of a cantilever timber beam. The first example uses material-based sensitivity implementations while the second example incorporates nodal sensitivity. Finally, a third example involves an unconstrained optimization on both thickness and width parameters. The analytical gradients were verified with finite difference approximations at the FORM level. Additionally, an analytical model of a cantilever beam was optimized in Excel using Solver and compared to the values from Rts. For each Rts example, the optimization was run twice; once with a starting point above the design value and once with the starting point below. The values of the various parameters are given in Table 4.

Table 4. Rts model list of parameters and values

Parameter	Type of Parameter	Distribution Type	Coefficient of Variation	Initial Value
Modulus of Elasticity (GPa)	Random Variable	Lognormal	0.15	13
Beam Width (m)	Decision Variable	-	-	3
Beam Length (m)	Constant	-	-	9
Poisson's Ratio	Constant	-	-	0.25
Beam Thickness (m)	Decision Variable	-	-	0.3
Cost of Beam Construction (\$/m ³)	Random Variable	Normal	0.20	200
Damage Cost Due to Deflection (\$/mm)	Random Variable	Normal	0.20	1000
Mass Density (kg/m ³)	Random Variable	Lognormal	0.05	500
Point Load (kN)	Random Variable	Normal	0.30	17

The model consists of random variables, constants, and decision variable parameters. The un-deformed cantilever beam FEM Model is shown in Figure 9.

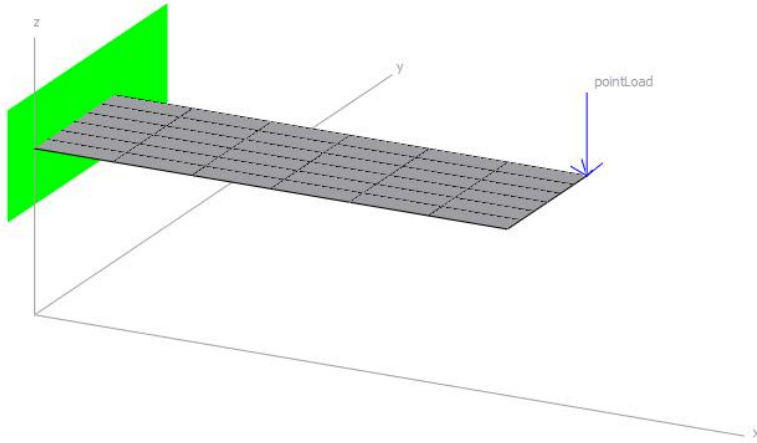
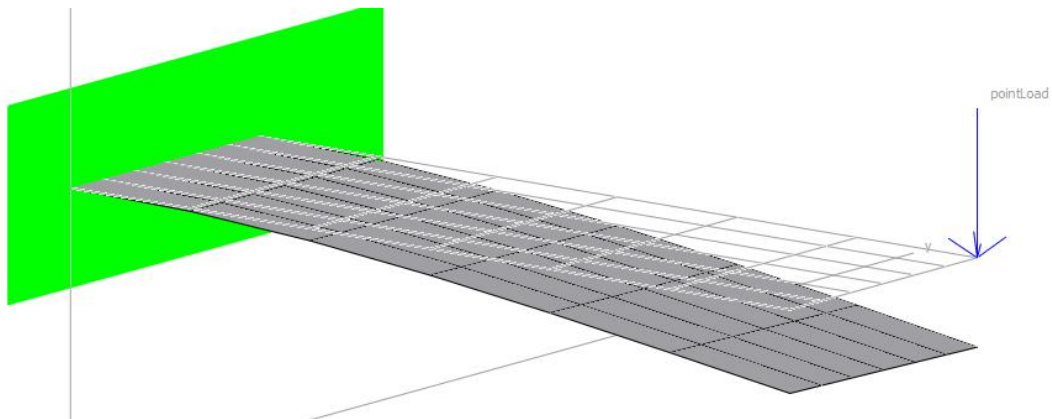


Figure 9. Cantilever beam model in Rts

The loading on the model includes self-weight and a point load. It can easily be extended to support distributed loading over an area as this is similar to the body-loading (or self-weight) already accounted for. The model consists of 36 elements; six in either direction. In this optimization, the four node bilinear Mindlin plate element was used. The deformed beam is shown below in Figure 10.



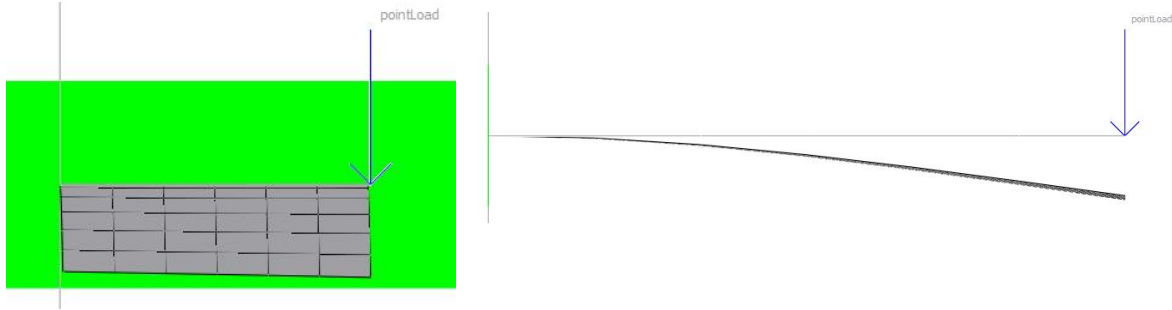


Figure 10. Deflected beam Rts model

The original un-deformed beam is shown as a light grey line. The Rts input files are given in Appendix A.1: Rts Input File Thickness Optimization and Appendix A.2: Rts Input File Width Optimization.

Table 5 below summarizes the FEM parameters used in the structural analysis.

Table 5. Optimization analysis parameters

Parameter	Value
Number of FEM Integration Points	1
Element Type	Bilinear Mindlin 4-Node
Integration Type	Gaussian
Optimization Type	Unconstrained
Search Direction	Steepest Descent
Step Size	Fixed

The results of the Rts examples and Excel model are summarized in Table 6.

Table 6. Rts and Excel optimization example results comparison

Example	Initial Value (m)	Final Design Value (m)	Iteration Number at Convergence	Time to Complete (min)	Mean Cost (\$)
Beam Thickness	0.3	0.1769	10	20	1311
Beam Thickness	0.05	0.1769	10	25.4	1311
Beam Width	0.7	1.1587	6	14.4	929

Example	Initial Value (m)	Final Design Value (m)	Iteration Number at Convergence	Time to Complete (min)	Mean Cost (\$)
Beam Width	3	1.1627	8	25.3	929
Excel Beam Thickness	0.3	0.1754	-	-	1303
Excel Beam Width	3	1.1507	-	-	921

The analytical Excel values match closely to those from Rts. The number of iterations to convergence depends on step size. All examples used a fixed step size. If the step size is too small, it will take longer to converge. In both optimization examples, the material and geometric properties are equivalent with the exception of the thickness being the decision variable in the first and the beam width in the other. The following sections showcase the Rts examples along with their respective GUI output. The input file for each optimization is given in the Appendices.

5.1 Optimization of Beam Thickness Using Material DDM Sensitivities

This section outlines an RBDO example using material sensitivities. In this case, the decision variable is the beam thickness. In the RBDO class, the steepest descent method is used with a step size of 0.0005. The optimization was started at higher and lower than design thickness values to illustrate convergence capabilities. The first optimization was started with an initial thickness of 0.05 m.

The GUI Rts output of the optimization is shown below.

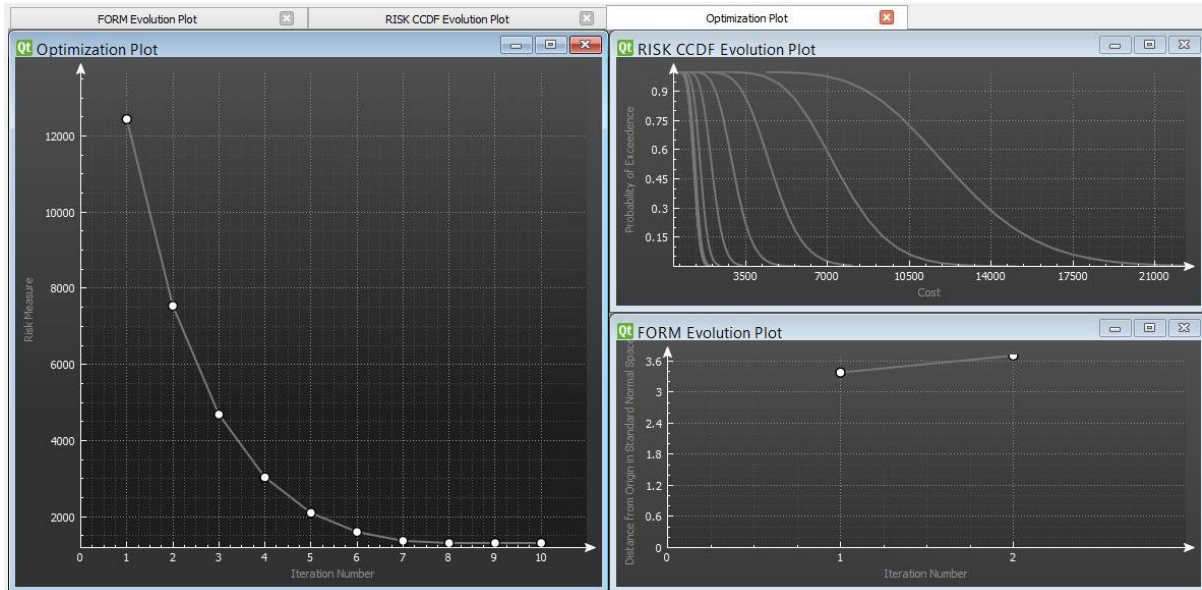


Figure 11. Thickness optimization Rts GUI Ouput

The Rts output to the user consists of both graphical and text components. The text component is given in Appendix A. Figure 11 illustrates the GUI output of the thickness optimization. The optimization iteration number and objective function value are plotted on the left. The RISK CCDF is plotted in the upper right hand corner while the FORM evolution plot is given in the bottom right.

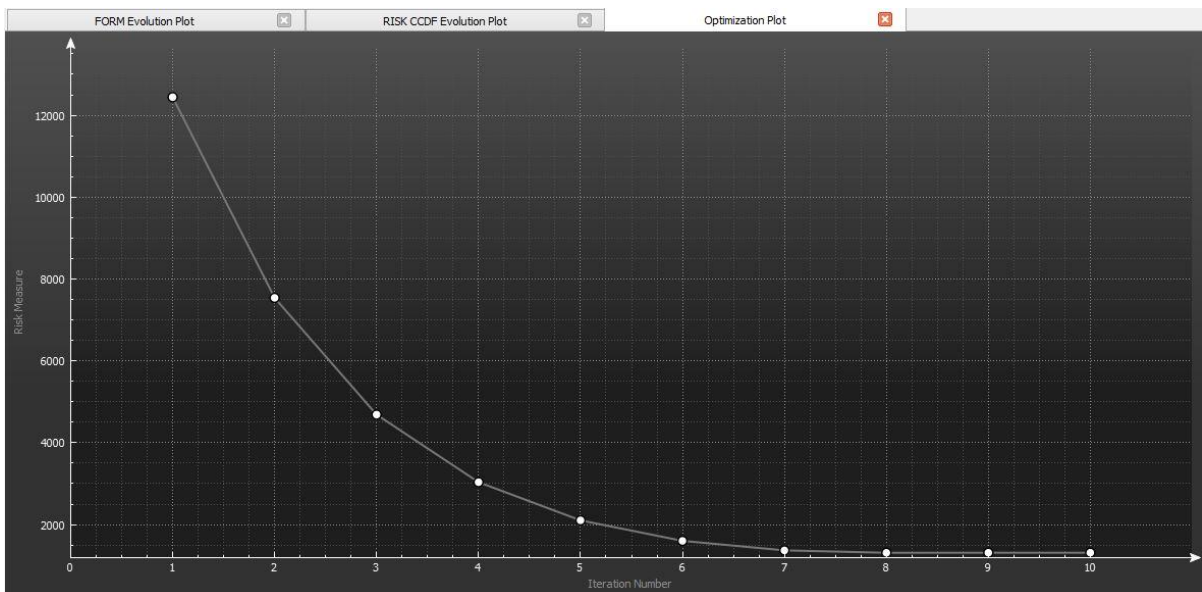


Figure 12. Optimization plot; iteration number vs. objective function value

Figure 12 shows the decreasing objective function value as the optimization progresses. At around the sixth iteration the thickness approaches the optimum cost. The beam is initially very thin therefore deflection governs and the high cost is due to a large deflection value.

Figure 13 below shows the evolution of the CCDF curve with each optimization iteration.

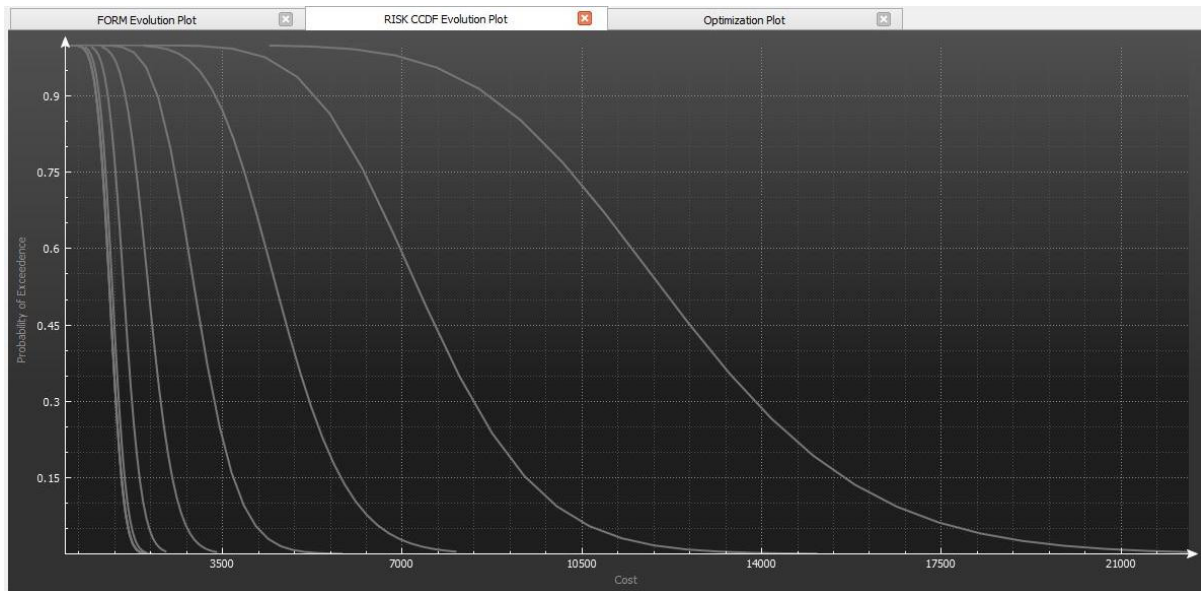


Figure 13. Thickness optimization CCDF

As the thickness approaches the optimum, the spacing between the curves decreases while the slope increases. The area under the leftmost curve represents the lowest mean cost; the optimization proceeds from right to left. The plot shown is the mean probabilistic cost of the beam vs. the probability of exceedance.

The second optimization started with a beam that was 0.3 m thick. In this case, the deflection was smaller while the material cost of manufacturing the beam governed. The GUI output is shown below while the text output can be seen in Appendix B.1: Material Optimization Output.

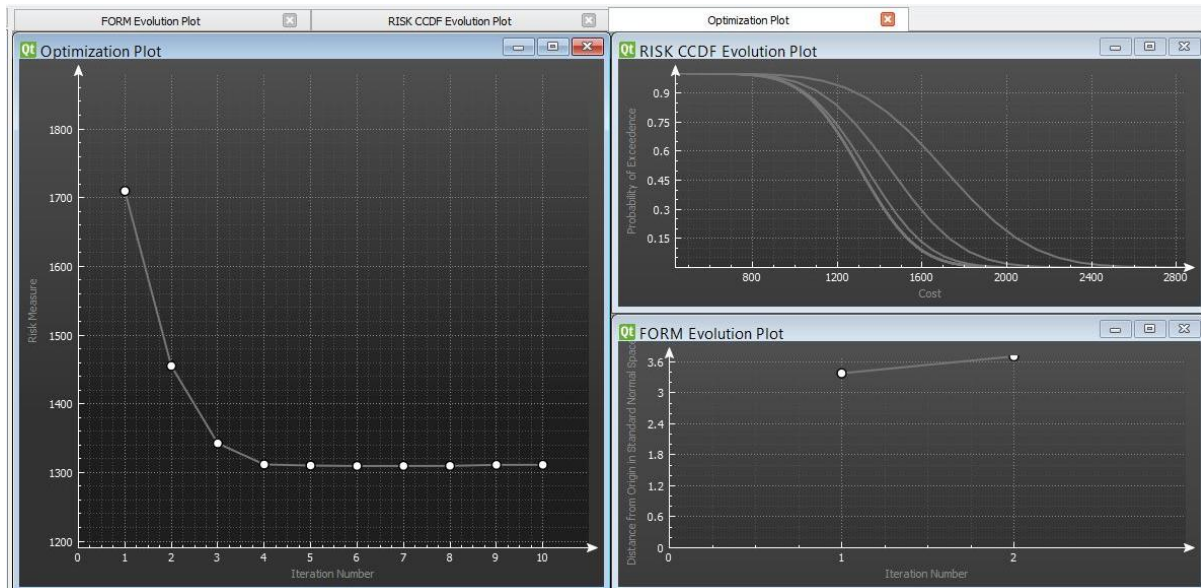


Figure 14. Thickness optimization full GUI

In this case the optimization approaches the design thickness within four iterations as can be seen in Figure 14 and Figure 15.

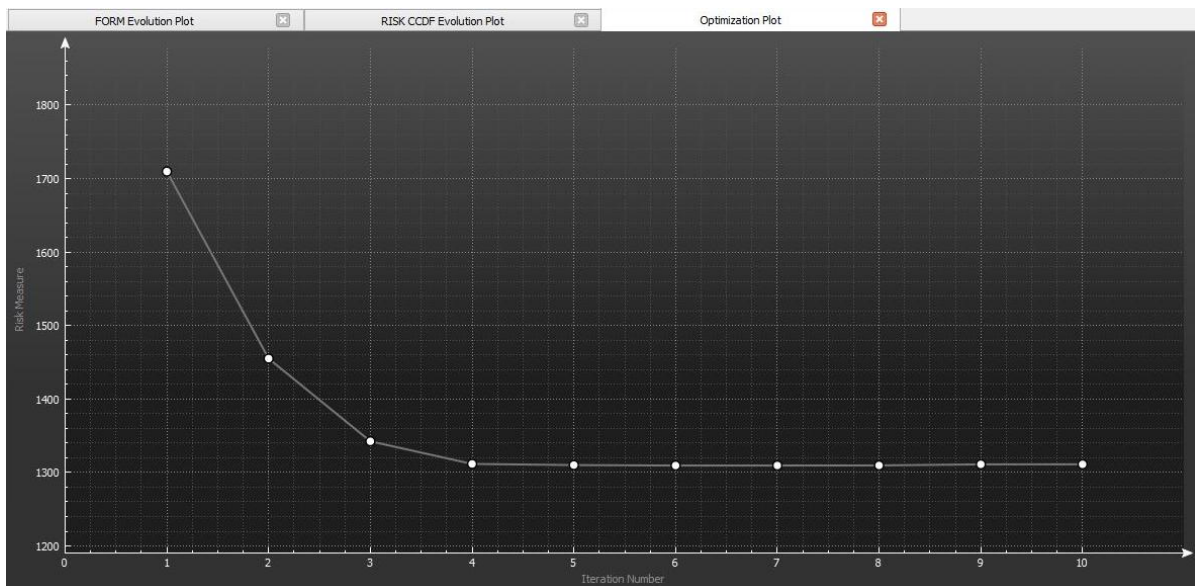


Figure 15. Thickness optimization plot

The CCDF evolution plot is additionally given below for reference.

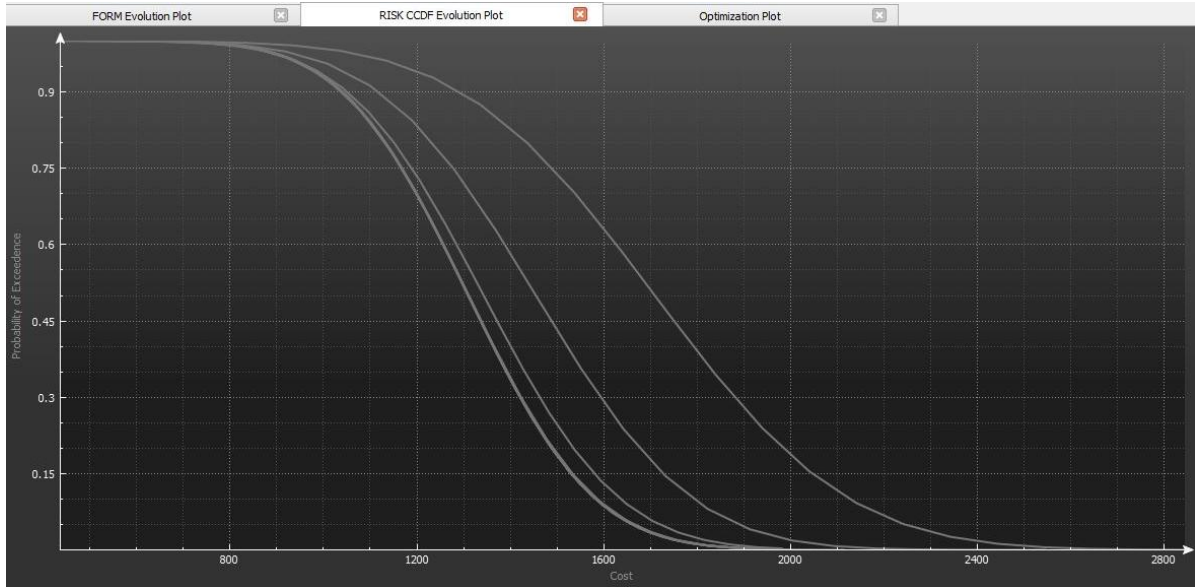


Figure 16. Thickness optimization CCDF Evolution

As can be seen in Figure 16, as the decision variable value approaches the design point, the CCDF curve spacing diminishes as it shifts leftward. In the thickness case, the final mean cost was \$1311.

5.2 Optimization of Beam Width Using Nodal DDM Sensitivities

In this section, the width of a timber beam is optimized. The parameters and their respective values are given in Table 4. In this optimization, the decision variable involves nodal coordinates; the beam width. As in the beam thickness example, two values were chosen above and below the design point. The first analysis starts with a width value of 0.7 m.

The Rts GUI output of the optimization is given below in Figure 17.

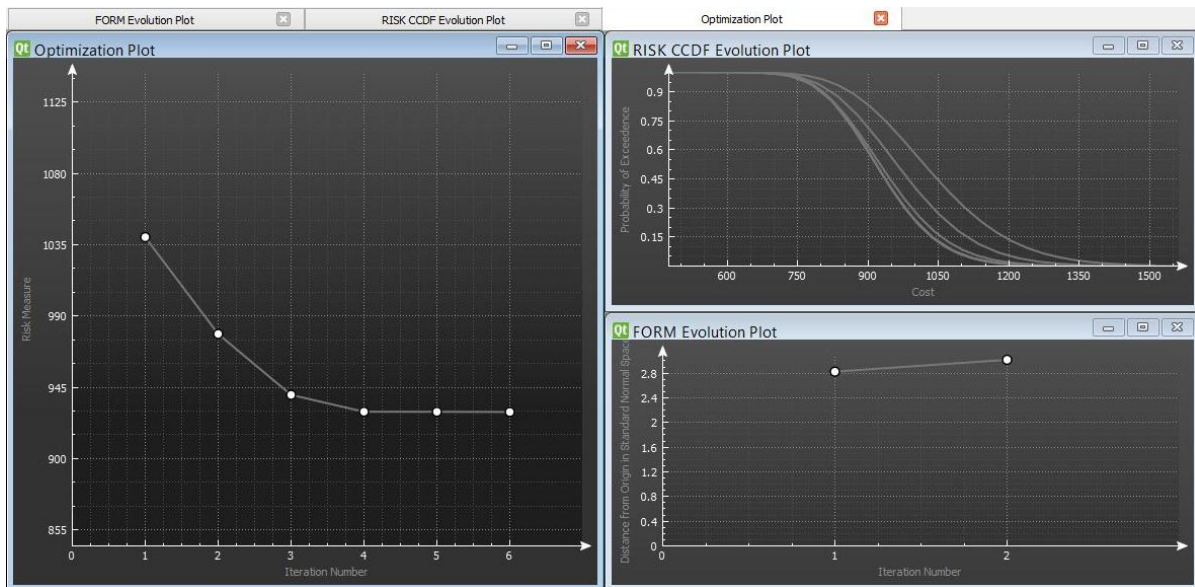


Figure 17. Width optimization full Rts GUI output

As can be seen in the optimization plot shown in Figure 18, the optimization approaches the design point after the third iteration.

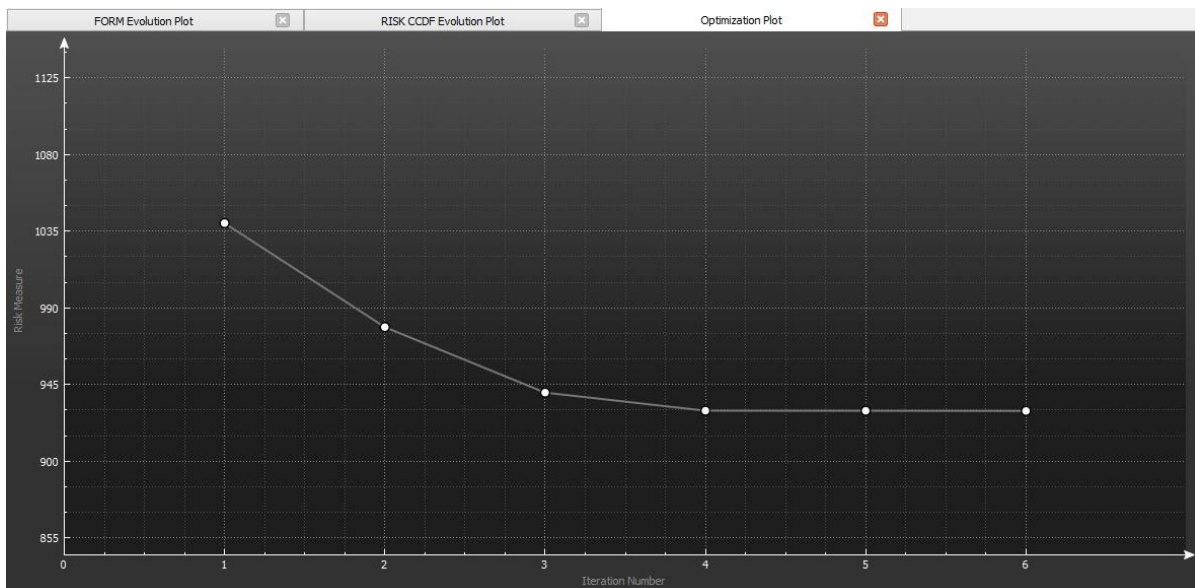


Figure 18. Width optimization plot; iteration number vs. objective function value

The CCDF evolution plot is also given below.

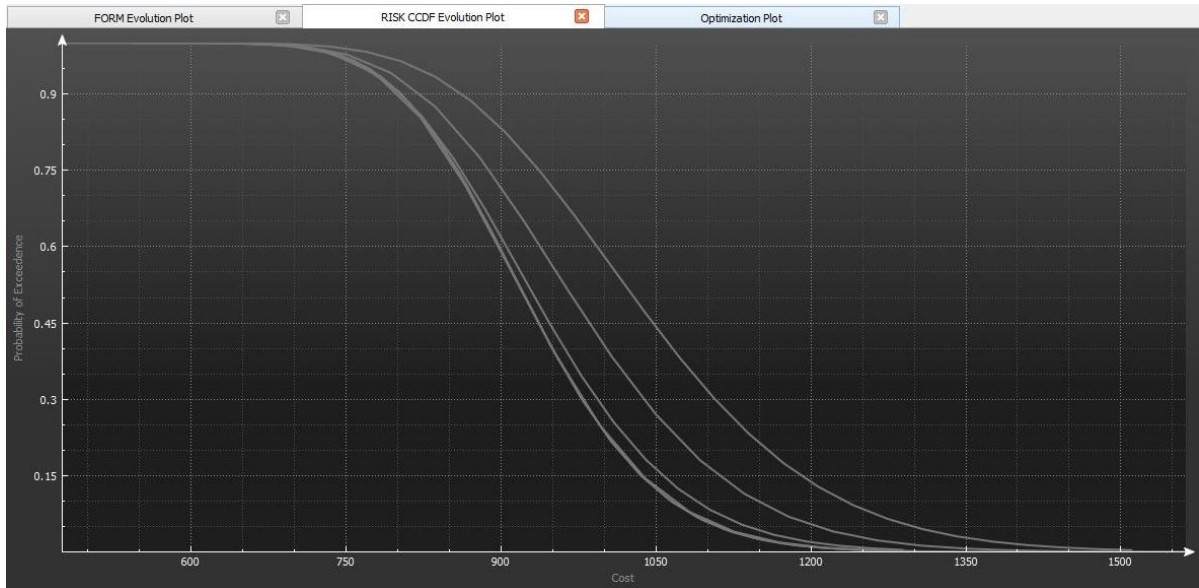


Figure 19. Width optimization RISK CCDF evolution plot

As in the thickness optimization, the RISK CCDF curve spacing becomes smaller as the curves proceed to the left. The optimization was repeated with an initial value of 3 m; well above the design value. The GUI output is presented below.

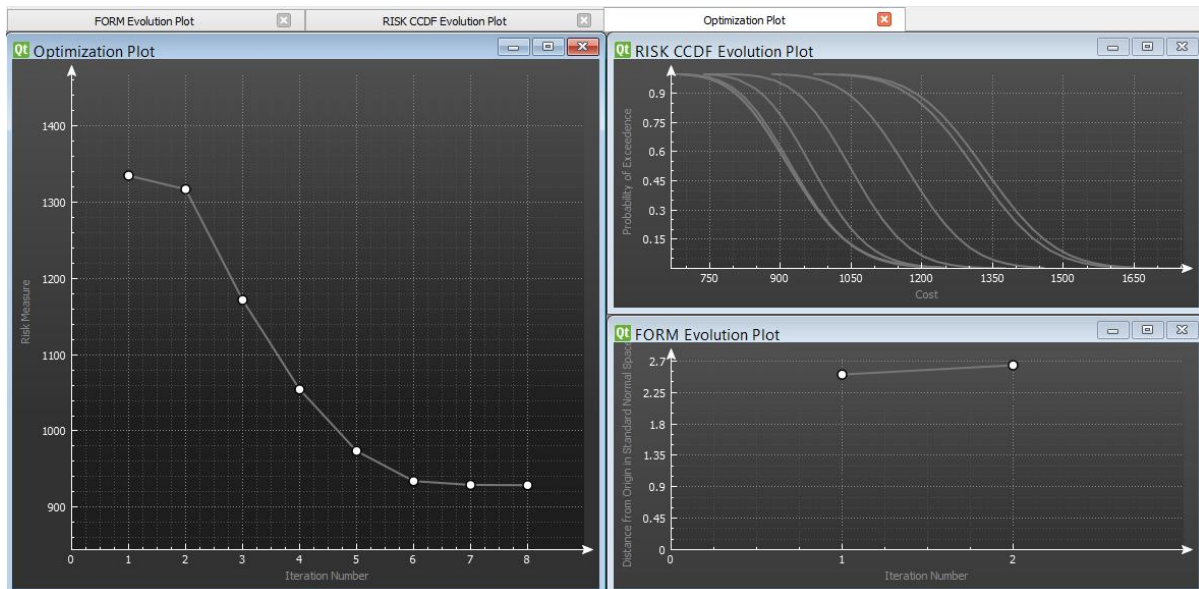


Figure 20. Width optimization Rts GUI Output

The optimization completed within eight steps with the decision variable approaching the optimum value at around the sixth step.

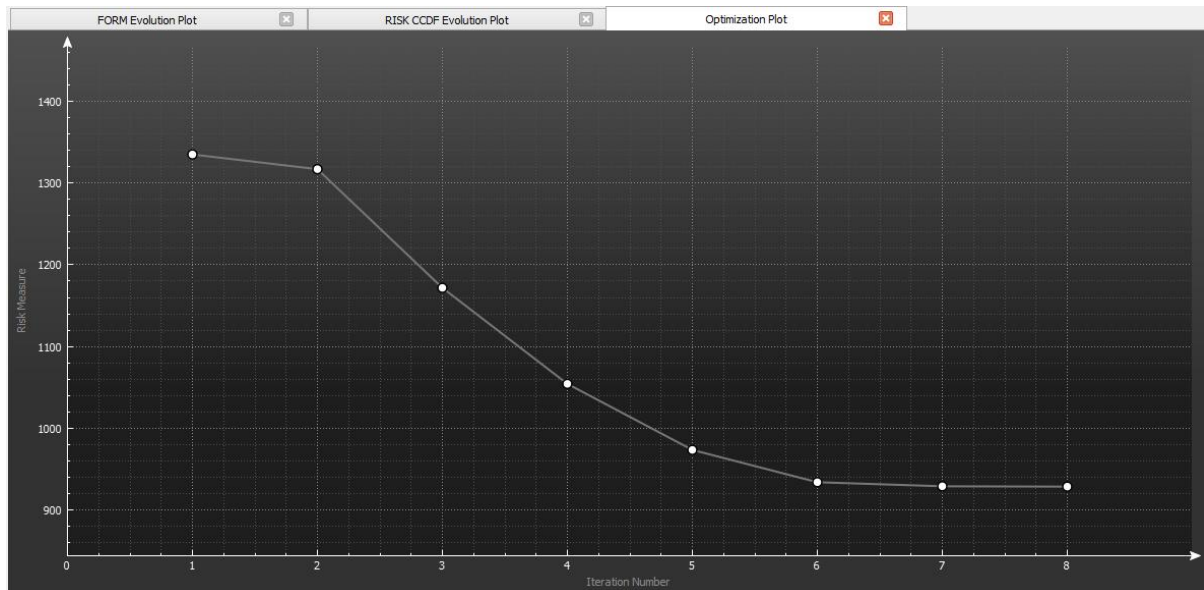


Figure 21. Width optimization; iteration number vs. objective function value

The final decision variable value at convergence was 1.1627 m after 8 steps.

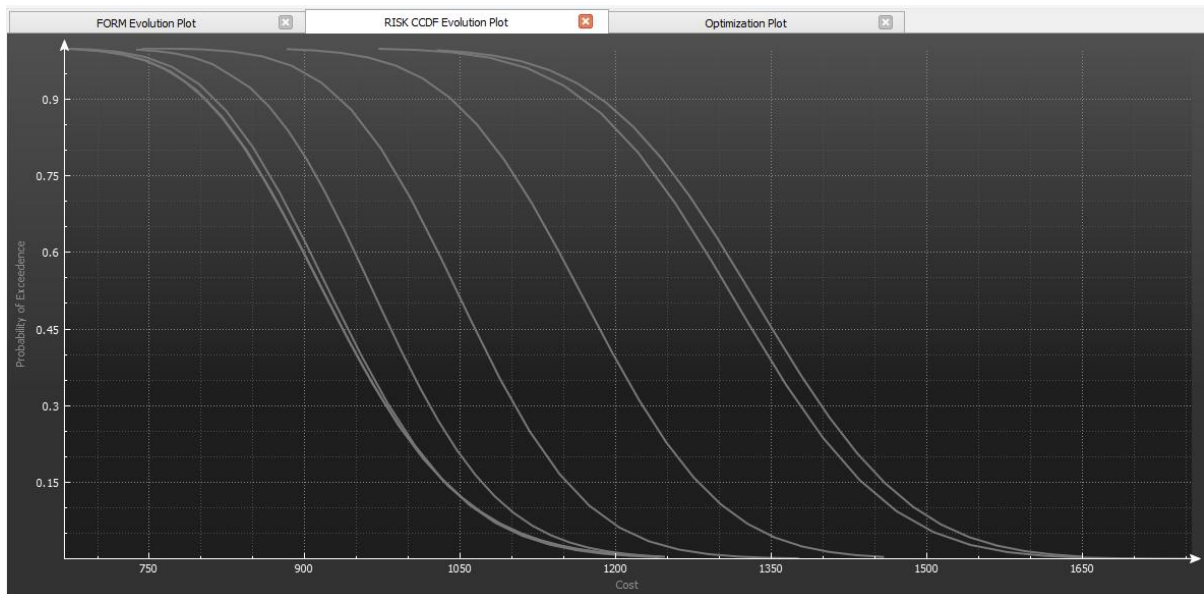


Figure 22. Width optimization CCDF curve

The mean cost was \$929 at the final iteration. The console output is given in Appendix B.2:

Nodal Optimization Output.

5.3 Simultaneous Optimization of a Beam Width and Thickness

Since this is an example of unconstrained optimization, as expected, the thickness of the beam keeps increasing while the width keeps decreasing. This is consistent with the fact that tall and slender beams are more efficient in bending. Figure 23 and Figure 24 highlight the 50 first steps of such an optimization.

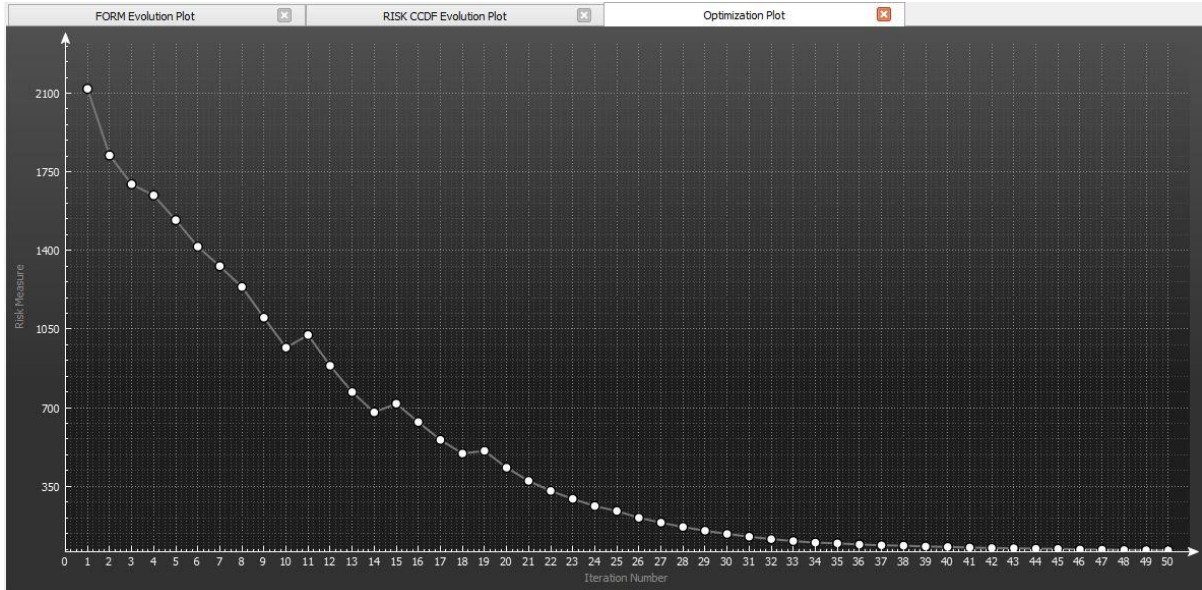


Figure 23. Iteration number vs. objective function value of simultaneous optimization

The initial parameter values were 0.1 and 5 for the beam thickness and width respectively.

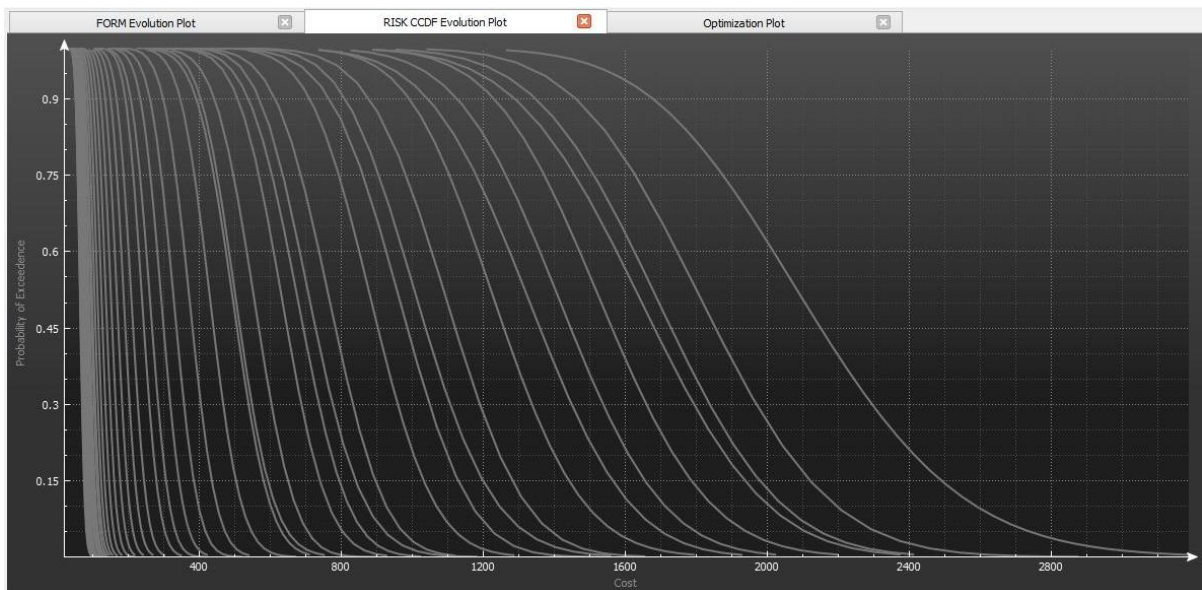


Figure 24. CCDF curve of simultaneous optimization

After 50 iterations, the final values for the thickness and width were 2.756 m and 0.00728 m. Eventually the optimization would break down since, unconstrained, the beam thickness would theoretically approach infinity while the width would approach zero. The mean cost after 50 iterations was \$64.

Chapter 6: CONCLUSION

The major contribution of this thesis is the implementation of nodal and material analytical gradients throughout the Rts framework starting at the element level up through to the optimization level. The elements implemented consist of the Bilinear-Mindlin four node and nine node plate elements. A framework was also put into place in where adding DDM capabilities is possible with any element; with analytical gradients hardcoded in at the element level.

An academic cost model was created to showcase the multi-model capabilities of Rts and the ability to calculate DDM dependencies of downstream models. A RISK model was implemented that incorporated a built-in FORM model with gradient-history capabilities and in-model DDM dependency calculations. Additionally, an RBDO optimization model was built upon to include DDM capabilities and downstream model control. Finally, two reliability-based design optimization examples were implemented that used both nodal and material sensitivities.

Future research should entail “piece-wise” optimization where the objective function is not smooth throughout the entire optimization stream; this would result in a discontinuity of the gradients. Such discontinuities would likely be an issue at the cost model level since cost models have the potential to be very complex, incorporating subjective decision making aspects. Additionally, more components and elements should be added at the FEM level to increase the capabilities of Rts along with implementation of the adjoint method and extending the analytical sensitivities implementation to incorporate inelastic and dynamic

analysis.

Using the Rts optimization framework involves engineering judgment such as selecting the optimum number of FEM elements, the number of FEM integration points, the number of RISK thresholds, the initial sampling number, and various step sizes throughout the models.

Application of this research provides the means for computer-aided reliability-based structural optimization. In summary, engineers and researchers could use Rts to find the optimum values of structural components to make the best decisions under uncertainty.

Bibliography

- Anderson, K. H. (2000). Analytic Full-Recursive Sensitivity Analysis for Multibody Systems Design. *Proceedings of the International Congress of Theoretical and Applied Mechanics*, Paper-1874.
- Christensen, P. W., & Klarbring, A. (2009). *An Introduction to Structural Optimization*. Sweden: Springer.
- Cook, R., Malkus, D. S., Plesha, M. E., & Witt, R. J. (2002). *Concepts and Applications of Finite Element Analysis 4th Ed*. Hoboken: Wiley.
- Eisley, J. (1989). *Mechanics of elastic structures: classical and finite element methods*. Michigan: Prentice Hall.
- Flaherty E, J. (2014). *Finite Element Analysis: Mesh Generation and Assembly*. New York: Rensselaer Polytechnic Institute.
- Haukaas, T., & Der Kiureghian, A. (April 2014). *Finite Element Reliability and Sensitivity Methods for Performance-Based Earthquake Engineering*. University of California, Berkeley: Pacific Earthquake Engineering Research Center.
- Mahsuli, M., & Haukaas, T. (2013). Computer Program for Multimodel Reliability and Optimization Analysis. *Journal of Computing in Civil Engineering* , 87-98.
- Singhal, S., Ghiocel, D. M., & Nikolaidis, E. (2004). *Engineering Design Reliability Handbook*. CRC Press.

APPENDICES

Appendix A

Rts Input File

A.1 Rts Input File Thickness Optimization

```
//      By Stevan Gavrilovic
//      Developed February 2015
//
//      Description: A thickness optimization of a cantilever with a point load using
//      cost as a function of the cantilever's volume

// -1)*****TOOLS USED BY THE VARIOUS METHODS*****

//LINEAR SOLVER
RInHouseLinearSolver |ObjectName: myLinearSolver |OutputLevel: Minimum

//PROBABILITY DISTRIBUTIONS
RInHouseProbabilityDistributions |ObjectName: theProbabilityDistributions
|OutputLevel: Minimum

//PROBABILITY TRANSFORMATIONS
RIndependenceProbabilityTransformation |ObjectName: myProbabilityTransformations
|OutputLevel: Minimum |ProbabilityDistributions: theProbabilityDistributions

//RANDOM NUMBER GENERATOR
RInHouseRandomNumberGenerator |ObjectName: myInHouseRandomNumberGenerator
|OutputLevel: Minimum |Seed: 0

//MATRIX ASSEMBLER
RConnectivityTableAssembler |ObjectName: myAssembler |OutputLevel: Minimum

// -2)*****DEFINE THE LOADING *****

//CONCENTRATED POINT LOAD
RContinuousRandomVariable |ObjectName: pointLoad |ProbabilityDistributions:
theProbabilityDistributions |DistributionType: Normal (mean, stdv) |Mean: -17000
|StandardDeviation: 500 |CurrentValue: -17000

// -3)*****DEFINE THE MODEL'S RANDOM VARIABLES*****

// MATERIAL PROPERTIES RANDOM VARIABLES ARE SET IN THE COMPONENT CODE

// COST RANDOM VARIABLES
// Cost to supply and install materials at current rate - based on volume in this case
RContinuousRandomVariable |ObjectName: thetaCLT |ProbabilityDistributions:
theProbabilityDistributions |DistributionType: Normal (mean, stdv) |Mean: 200
|CoefficientOfVariation: 0.2 |CurrentValue: 200

//Cost of repair from damage models
RContinuousRandomVariable |ObjectName: thetaDeflection |ProbabilityDistributions:
theProbabilityDistributions |DistributionType: Normal (mean, stdv) |Mean: -1000
|CoefficientOfVariation: 0.2 |CurrentValue: -1000
```

```

// -4)*****DEFINE THE OPTIMIZATION DECISION VARIABLES*****

// Beam thickness
RContinuousDecisionVariable |ObjectName: beamThickness |CurrentValue: 0.05
|InitialValue: 0.05 |UpperBound: 0 |LowerBound: 0 |IncrementalCost: 1

// -5)*****DEFINE THE COMPONENT CONSTANTS*****

RConstant |ObjectName: beamLength |CurrentValue: 9
RConstant |ObjectName: beamWidth |CurrentValue: 3

// -6)*****DEFINE THE COMPONENTS'S GEOMETRY*****

//FIXED WALL GEOMETRY
RPoint |ObjectName: pointA |XCoordinate: 0 |YCoordinate: -1 |ZCoordinate: 2
RPoint |ObjectName: pointB |XCoordinate: 0 |YCoordinate: 5 |ZCoordinate: 2
RPoint |ObjectName: pointC |XCoordinate: 0 |YCoordinate: 5 |ZCoordinate: 4
RPoint |ObjectName: pointD |XCoordinate: 0 |YCoordinate: -1 |ZCoordinate: 4

//CANTILEVER BEAM GEOMETRY
RPoint |ObjectName: horzPoint1 |XCoordinate: 0 |YCoordinate: 0
|ZCoordinate: 3
RPoint |ObjectName: horzPoint2 |XCoordinate: beamLength |YCoordinate: 0
|ZCoordinate: 3
RPoint |ObjectName: horzPoint3 |XCoordinate: beamLength |YCoordinate:
beamWidth |ZCoordinate: 3 |ZForce: pointLoad
RPoint |ObjectName: horzPoint4 |XCoordinate: 0 |YCoordinate:
beamWidth |ZCoordinate: 3

// -7)*****DEFINE THE SYSTEM COMPONENTS*****

//CLT BEAM COMPONENT (HORIZONTAL PLATE BILINEAR MINDLIN 4-NODE)
RPlateComponent |ObjectName: cantileverBeam |Point1: horzPoint1 |Point2: horzPoint2
|Point3: horzPoint3 |Point4: horzPoint4 |MeshOption: 106 |Thickness: beamThickness

//FIXED WALL COMPONENT
RFixedPlaneComponent |ObjectName: ground |Point1: pointA |Point2: pointB |Point3:
pointC |Point4: pointD

// -8)*****DEFINE THE STRUCTURAL ANALYSIS TYPE*****

// LINEAR STRUCTURAL ANALYSIS
RLinearStaticStructuralAnalysis |ObjectName: myStaticAnalysis |OutputLevel: Minimum
|Assembler: myAssembler |LinearSolver: myLinearSolver

// -9)*****DEFINE THE ORCHESTRATING MODEL TOOLS AND INPUTS*****

// OPTIMIZATION TOOLS
RGradientNormConvergenceCheck |ObjectName: myOptimizationConvergenceCheck
|OutputLevel: Maximum |Tolerance: 10

// Steepest Descent
RFixedStepSize |ObjectName: myOptimizationFixedStepSize |OutputLevel: Minimum
|StepSize: 0.000005
//|StepSize: 0.0001

```



```

RSteepestDescentSearchDirection |ObjectName: myOptimizationSearchDirection
|OutputLevel: Minimum

// BFGS
//RFixedStepSize |ObjectName: myOptimizationFixedStepSize |OutputLevel: Minimum
|StepSize: 0.00001
//RBFSSearchDirection |ObjectName: myOptimizationSearchDirection |OutputLevel:
Minimum

// -10)*****DEFINE THE ORCHESTRATING MODELS*****

// RESPONSE MODEL
RComponentResponseModel |ObjectName: myStaticBuilding |OutputLevel: Minimum
|StructuralAnalysis: myStaticAnalysis |Responses: horzPoint3.ZDisplacement

//*****COSTS*****

// COSTS - TOTAL
RSteveTestCostModel |ObjectName: costTotal |OutputLevel: Maximum |Expression:
thetaCLT*beamWidth*beamThickness*beamLength +
thetaDeflection*myStaticBuildinghorzPoint3ZDisplacement | GradientMethod: Direct
Differentiation

//*****

// SAMPLING MODEL
RSamplingModel |ObjectName: mySamplingCostModel |OutputLevel: Minimum |InputParameter:
costTotalResponse |Threshold: 1300 |MaxSamples: 1 |TargetCov: 0.005 |SamplingCentre:
Origin |RandomNumberGenerator: myInHouseRandomNumberGenerator
|ProbabilityTransformation: myProbabilityTransformations

// RISK MODEL
RRiskModel |ObjectName: myRiskCostModel |OutputLevel: Maximum |RiskMeasure: 1
|InputFromCost: costTotalResponse

// OPTIMIZATION MODEL
ROptimizationModel |ObjectName: myRiskOptimizationCostModel |OutputLevel: Maximum
|Objective: myRiskCostModelResponse |MaxSteps: 20 |SearchDirection:
myOptimizationSearchDirection |StepSize: myOptimizationFixedStepSize
|ConvergenceCheck: myOptimizationConvergenceCheck |GradientMethod: Direct
Differentiation

```

A.2 Rts Input File Width Optimization

```
//      By Stevan Gavrilovic
//      Developed February 2015
//
//      Description: A width optimization of a cantilever with a point load using cost
//      as a function of the cantilever's volume

// -1)*****TOOLS USED BY THE VARIOUS METHODS*****

//LINEAR SOLVER
RInHouseLinearSolver |ObjectName: myLinearSolver |OutputLevel: Minimum

//PROBABILITY DISTRIBUTIONS
RInHouseProbabilityDistributions |ObjectName: theProbabilityDistributions
|OutputLevel: Minimum

//PROBABILITY TRANSFORMATIONS
RIndependenceProbabilityTransformation |ObjectName: myProbabilityTransformations
|OutputLevel: Minimum |ProbabilityDistributions: theProbabilityDistributions

//RANDOM NUMBER GENERATOR
RInHouseRandomNumberGenerator |ObjectName: myInHouseRandomNumberGenerator
|OutputLevel: Minimum |Seed: 0

//MATRIX ASSEMBLER
RConnectivityTableAssembler |ObjectName: myAssembler |OutputLevel: Minimum

// -2)*****DEFINE THE LOADING *****

//CONCENTRATED POINT LOAD
RContinuousRandomVariable |ObjectName: pointLoad |ProbabilityDistributions:
theProbabilityDistributions |DistributionType: Normal (mean, stdv) |Mean: -17000
|StandardDeviation: 500 |CurrentValue: -17000

// -3)*****DEFINE THE MODEL'S RANDOM VARIABLES*****

// MATERIAL PROPERTIES RANDOM VARIABLES ARE SET IN THE COMPONENT CODE

// COST RANDOM VARIABLES
// Cost to supply and install materials at current rate - based on volume in this case
RContinuousRandomVariable |ObjectName: thetaCLT |ProbabilityDistributions:
theProbabilityDistributions |DistributionType: Normal (mean, stdv) |Mean: 200
|CoefficientOfVariation: 0.1 |CurrentValue: 200
//Cost of repair from damage models
RContinuousRandomVariable |ObjectName: thetaDeflection |ProbabilityDistributions:
theProbabilityDistributions |DistributionType: Normal (mean, stdv) |Mean: -1000
|CoefficientOfVariation: 0.1 |CurrentValue: -1000
```

```

// -4)*****DEFINE THE OPTIMIZATION DECISION VARIABLES*****

// Beam width
RContinuousDecisionVariable |ObjectName: beamWidth |CurrentValue: 3 |InitialValue: 3
|UpperBound: 0 |LowerBound: 0 |IncrementalCost: 1

// -5)*****DEFINE THE COMPONENT CONSTANTS*****

RConstant |ObjectName: beamLength |CurrentValue: 9
RConstant |ObjectName: beamThickness |CurrentValue: 0.2

// -6)*****DEFINE THE COMPONENTS'S GEOMETRY*****

//FIXED WALL GEOMETRY
RPoint |ObjectName: pointA |XCoordinate: 0 |YCoordinate: -1 |ZCoordinate: 2
RPoint |ObjectName: pointB |XCoordinate: 0 |YCoordinate: 5 |ZCoordinate: 2
RPoint |ObjectName: pointC |XCoordinate: 0 |YCoordinate: 5 |ZCoordinate: 4
RPoint |ObjectName: pointD |XCoordinate: 0 |YCoordinate: -1 |ZCoordinate: 4

//CANTILEVER BEAM GEOMETRY
RPoint |ObjectName: horzPoint1 |XCoordinate: 0 |YCoordinate: 0
|ZCoordinate: 3
RPoint |ObjectName: horzPoint2 |XCoordinate: beamLength |YCoordinate: 0
|ZCoordinate: 3
RPoint |ObjectName: horzPoint3 |XCoordinate: beamLength |YCoordinate:
beamWidth |ZCoordinate: 3 |ZForce: pointLoad
RPoint |ObjectName: horzPoint4 |XCoordinate: 0 |YCoordinate:
beamWidth |ZCoordinate: 3

// -7)*****DEFINE THE SYSTEM COMPONENTS*****

//CLT BEAM COMPONENT (HORIZONTAL PLATE BILINEAR MINDLIN 4-NODE)
RPlateComponent |ObjectName: cantileverBeam |Point1: horzPoint1 |Point2: horzPoint2
|Point3: horzPoint3 |Point4: horzPoint4 |MeshOption: 106 |Thickness: beamThickness

//FIXED WALL COMPONENT
RFixedPlaneComponent |ObjectName: ground |Point1: pointA |Point2: pointB |Point3:
pointC |Point4: pointD

// -8)*****DEFINE THE STRUCTURAL ANALYSIS TYPE*****

// LINEAR STRUCTURAL ANALYSIS
RLinearStaticStructuralAnalysis |ObjectName: myStaticAnalysis |OutputLevel: Minimum
|Assembler: myAssembler |LinearSolver: myLinearSolver

```

```

// -9)*****DEFINE THE ORCHESTRATING MODEL TOOLS AND INPUTS*****

// OPTIMIZATION TOOLS
RGradientNormConvergenceCheck |ObjectName: myOptimizationConvergenceCheck
|OutputLevel: Maximum |Tolerance: 10

// Steepest Descent
RFixedStepSize |ObjectName: myOptimizationFixedStepSize |OutputLevel: Minimum
|StepSize: 0.0008
RSteepestDescentSearchDirection |ObjectName: myOptimizationSearchDirection
|OutputLevel: Minimum

// BFGS
//RFixedStepSize |ObjectName: myOptimizationFixedStepSize |OutputLevel: Minimum
|StepSize: 1
//RBFGSSearchDirection |ObjectName: myOptimizationSearchDirection |OutputLevel:
Minimum

// -10)*****DEFINE THE ORCHESTRATING MODELS*****

// RESPONSE MODEL
RComponentResponseModel |ObjectName: myStaticBuilding |OutputLevel: Minimum
|StructuralAnalysis: myStaticAnalysis |Responses: horzPoint3.ZDisplacement

//*****COSTS*****

// COSTS - TOTAL
RSteveTestCostModel |ObjectName: costTotal |OutputLevel: Maximum |Expression:
thetaCLT*beamWidth*beamThickness*beamLength +
thetaDeflection*myStaticBuildinghorzPoint3ZDisplacement | GradientMethod: Direct
Differentiation

//*****

// SAMPLING MODEL
RSamplingModel |ObjectName: mySamplingCostModel |OutputLevel: Minimum |InputParameter:
costTotalResponse |Threshold: 1300 |MaxSamples: 1 |TargetCov: 0.005 |SamplingCentre:
Origin |RandomNumberGenerator: myInHouseRandomNumberGenerator
|ProbabilityTransformation: myProbabilityTransformations

// RISK MODEL
RRiskModel |ObjectName: myRiskCostModel |OutputLevel: Maximum |RiskMeasure: 1
|InputFromCost: costTotalResponse

// OPTIMIZATION MODEL
ROptimizationModel |ObjectName: myRiskOptimizationCostModel |OutputLevel: Maximum
|Objective: myRiskCostModelResponse |MaxSteps: 20 |SearchDirection:
myOptimizationSearchDirection |StepSize: myOptimizationFixedStepSize
|ConvergenceCheck: myOptimizationConvergenceCheck |GradientMethod: Direct
Differentiation

```

Appendix B

Rts Console Output

B.1 Material Optimization Output

The OPTIMIZATION analysis in "myRiskOptimizationCostModel" started...

The decision variable values initially are:

| 0.05 |

The RISK analysis in "myRiskCostModel" started...

Now running the initial sampling to determine thresholds...

The random variable named "cantileverBeamPlateDensity" is missing the probability distributions tool,

hence, it is creating a tool called "defaultInHouseProbabilityDistributions"

The random variable named "cantileverBeamPlateYoungsModulus" is missing the probability distributions tool,

hence, it is picking up the "defaultInHouseProbabilityDistributions" that is already available.

The sampling mean-value is:

13377.7

Reducing the std. dev. to mean to 4.75

Reducing the std. dev. to mean to 4.5

Reducing the std. dev. to mean to 4.25

Determining thresholds using points within 4.25 standard deviations from the mean.

The threshold 1 is 4426.55

The threshold 2 is 5240.3

The threshold 3 is 6054.04

The threshold 4 is 6867.79

The threshold 5 is 7681.53

The threshold 6 is 8495.27

The threshold 7 is 9309.02

The threshold 8 is 10122.8

The threshold 9 is 10936.5

The threshold 10 is 11750.2

The threshold 11 is 12564

The threshold 12 is 13377.7

The threshold 13 is 14191.5

The threshold 14 is 15005.2

The threshold 15 is 15819

The threshold 16 is 16632.7

The threshold 17 is 17446.5

The threshold 18 is 18260.2

The threshold 19 is 19073.9

The threshold 20 is 19887.7

The threshold 21 is 20701.4

The threshold 22 is 21515.2

The threshold 23 is 22328.9

The FORM analysis in RISK at threshold "4426.55" has started..

For exceedence probability curve (threshold number, threshold, probability):

```
1 , 4426.55 , 0.999156
2 , 5240.3 , 0.997142
3 , 6054.04 , 0.991911
4 , 6867.79 , 0.97968
5 , 7681.53 , 0.955488
6 , 8495.27 , 0.914106
7 , 9309.02 , 0.852095
8 , 10122.8 , 0.769566
9 , 10936.5 , 0.670758
10 , 11750.2 , 0.56303
11 , 12564 , 0.4549
12 , 13377.7 , 0.354049
13 , 14191.5 , 0.26585
14 , 15005.2 , 0.193012
15 , 15819 , 0.135802
16 , 16632.7 , 0.0928361
17 , 17446.5 , 0.0618074
18 , 18260.2 , 0.0401754
19 , 19073.9 , 0.025554
20 , 19887.7 , 0.015936
21 , 20701.4 , 0.00976417
22 , 21515.2 , 0.00588768
23 , 22328.9 , 0.00350003
```

The risk measure is: 12440.6

The analysis in "myRiskCostModel" completed in 126.855 seconds.

At optimization step 1 the objective function in "myRiskOptimizationCostModel" is 12440.6

The decision variable values (before updating) are:

| 0.05 |

The optimization gradient is currently:

|-1.38693e+06 |

The norm in "myOptimizationConvergenceCheck" is 1.38693e+06

Now running the intial sampling to determine thresholds...

The sampling mean-value is:

8115.12

The FORM analysis in RISK at threshold "1167.02" has started..

For exceedence probability curve (threshold number, threshold, probability):

```
1 , 1167.02 , 0.999994
2 , 1798.67 , 0.999952
3 , 2430.31 , 0.999699
4 , 3061.96 , 0.998346
5 , 3693.61 , 0.992898
6 , 4325.25 , 0.976226
7 , 4956.9 , 0.937162
8 , 5588.54 , 0.865241
9 , 6220.19 , 0.757978
10 , 6851.83 , 0.62449
11 , 7483.48 , 0.482125
12 , 8115.12 , 0.348988
13 , 8746.77 , 0.237641
14 , 9378.41 , 0.152967
15 , 10010.1 , 0.0935804
16 , 10641.7 , 0.0547176
```

17 , 11273.4 , 0.0307372
18 , 11905 , 0.0166718
19 , 12536.6 , 0.00876941
20 , 13168.3 , 0.00449065
21 , 13799.9 , 0.00224713
22 , 14431.6 , 0.00110201
23 , 15063.2 , 0.000531233

The risk measure is: 7537.98

The analysis in "myRiskCostModel" completed in 134.855 seconds.

At optimization step 2 the objective function in "myRiskOptimizationCostModel" is 7537.98

The decision variable values (before updating) are:

| 0.06 |

The optimization gradient is currently:

| -676328 |

The norm in "myOptimizationConvergenceCheck" is 676328

Now running the initial sampling to determine thresholds...

The sampling mean-value is:

4363.81

The FORM analysis in RISK at threshold "1986.6" has started..

For exceedence probability curve (threshold number, threshold, probability):

1 , 1986.6 , 0.998448
2 , 2202.71 , 0.996291
3 , 2418.82 , 0.991987
4 , 2634.93 , 0.983834
5 , 2851.04 , 0.969812
6 , 3067.15 , 0.947636
7 , 3283.26 , 0.915221
8 , 3499.37 , 0.871189
9 , 3715.48 , 0.815294
10 , 3931.59 , 0.748619
11 , 4147.7 , 0.673506
12 , 4363.81 , 0.593189
13 , 4579.92 , 0.511304
14 , 4796.03 , 0.431371
15 , 5012.14 , 0.356372
16 , 5228.25 , 0.288496
17 , 5444.36 , 0.229054
18 , 5660.47 , 0.178532
19 , 5876.58 , 0.13675
20 , 6092.69 , 0.103045
21 , 6308.8 , 0.0764659
22 , 6524.91 , 0.0559376
23 , 6741.02 , 0.0403797
24 , 6957.13 , 0.028791
25 , 7173.24 , 0.0202945
26 , 7389.35 , 0.0141546
27 , 7605.46 , 0.00977601
28 , 7821.57 , 0.00669111
29 , 8037.67 , 0.00454171

The risk measure is: 4685.46

The analysis in "myRiskCostModel" completed in 160.809 seconds.

At optimization step 3 the objective function in "myRiskOptimizationCostModel" is 4685.46

The decision variable values (before updating) are:

| 0.072 |

The optimization gradient is currently:

| -326423 |

The norm in "myOptimizationConvergenceCheck" is 326423

Now running the initial sampling to determine thresholds...

The sampling mean-value is:

3201.2

The FORM analysis in RISK at threshold "583.445" has started..

For exceedence probability curve (threshold number, threshold, probability):

1 , 583.445 , 0.999999

2 , 821.423 , 0.999986

3 , 1059.4 , 0.999894

4 , 1297.38 , 0.999295

5 , 1535.36 , 0.996365

6 , 1773.33 , 0.985669

7 , 2011.31 , 0.956515

8 , 2249.29 , 0.895825

9 , 2487.27 , 0.796305

10 , 2725.25 , 0.663538

11 , 2963.22 , 0.51507

12 , 3201.2 , 0.37217

13 , 3439.18 , 0.251086

14 , 3677.16 , 0.15899

15 , 3915.13 , 0.0950777

16 , 4153.11 , 0.054036

17 , 4391.09 , 0.0293678

18 , 4629.07 , 0.0153495

19 , 4867.05 , 0.00775516

20 , 5105.02 , 0.00380511

21 , 5343 , 0.00182038

22 , 5580.98 , 0.000852197

23 , 5818.96 , 0.000391622

The risk measure is: 3034.37

The analysis in "myRiskCostModel" completed in 151.07 seconds.

At optimization step 4 the objective function in "myRiskOptimizationCostModel" is 3034.37

The decision variable values (before updating) are:

| 0.0864 |

The optimization gradient is currently:

| -156205 |

The norm in "myOptimizationConvergenceCheck" is 156205

Now running the initial sampling to determine thresholds...

The sampling mean-value is:

2180.97

The FORM analysis in RISK at threshold "1241.83" has started..

For exceedence probability curve (threshold number, threshold, probability):

```
1 , 1156.45 , 0.996412
2 , 1241.83 , 0.992054
3 , 1327.2 , 0.983251
4 , 1412.58 , 0.969421
5 , 1497.96 , 0.946143
6 , 1583.33 , 0.911998
7 , 1668.71 , 0.86522
8 , 1754.09 , 0.805533
9 , 1839.46 , 0.73421
10 , 1924.84 , 0.653985
11 , 2010.22 , 0.568674
12 , 2095.6 , 0.48251
13 , 2180.97 , 0.399518
14 , 2266.35 , 0.322951
15 , 2351.73 , 0.255069
16 , 2437.1 , 0.197035
17 , 2522.48 , 0.149025
18 , 2607.86 , 0.110496
19 , 2693.23 , 0.0804189
20 , 2778.61 , 0.0575161
21 , 2863.99 , 0.0404777
22 , 2949.37 , 0.0280638
23 , 3034.74 , 0.0191895
24 , 3120.12 , 0.0129541
25 , 3205.5 , 0.00864244
26 , 3290.87 , 0.00570406
27 , 3376.25 , 0.00372727
```

The risk measure is: 2102.07

The analysis in "myRiskCostModel" completed in 170.443 seconds.

At optimization step 5 the objective function in "myRiskOptimizationCostModel" is 2102.07

The decision variable values (before updating) are:

```
| 0.10368 |
```

The optimization gradient is currently:

```
| -72471.3 |
```

The norm in "myOptimizationConvergenceCheck" is 72471.3

Now running the intial sampling to determine thresholds...

The sampling mean-value is:

```
1642.72
```

The FORM analysis in RISK at threshold "958.301" has started..

]

For exceedence probability curve (threshold number, threshold, probability):

```
1 , 958.301 , 0.995644
2 , 1020.52 , 0.990432
3 , 1082.74 , 0.980777
4 , 1144.96 , 0.964031
5 , 1207.18 , 0.937402
6 , 1269.4 , 0.898241
7 , 1331.62 , 0.844844
8 , 1393.84 , 0.777098
9 , 1456.06 , 0.696833
10 , 1518.28 , 0.607699
11 , 1580.5 , 0.514576
12 , 1642.72 , 0.422701
```

13 , 1704.94 , 0.336783
14 , 1767.16 , 0.260348
15 , 1829.38 , 0.195427
16 , 1891.6 , 0.142601
17 , 1953.82 , 0.101288
18 , 2016.04 , 0.0701341
19 , 2078.26 , 0.0474155
20 , 2140.48 , 0.0313502
21 , 2202.7 , 0.0203044
22 , 2264.91 , 0.0129024
23 , 2327.13 , 0.00805538
24 , 2389.35 , 0.0049488

The risk measure is: 1600.91

The analysis in "myRiskCostModel" completed in 149.305 seconds.

At optimization step 6 the objective function in "myRiskOptimizationCostModel" is 1600.91

The decision variable values (before updating) are:

| 0.124416 |

The optimization gradient is currently:

| -30620.5 |

The norm in "myOptimizationConvergenceCheck" is 30620.5

Now running the initial sampling to determine thresholds...

The sampling mean-value is:

1395.66

The FORM analysis in RISK at threshold "835.815" has started..

For exceedence probability curve (threshold number, threshold, probability):

1 , 835.815 , 0.995175
2 , 886.71 , 0.990117
3 , 937.605 , 0.981052
4 , 988.499 , 0.965794
5 , 1039.39 , 0.941821
6 , 1090.29 , 0.906558
7 , 1141.18 , 0.857983
8 , 1192.08 , 0.795202
9 , 1242.97 , 0.719032
10 , 1293.87 , 0.632093
11 , 1344.76 , 0.538682
12 , 1395.66 , 0.443984
13 , 1446.55 , 0.353325
14 , 1497.45 , 0.271159
15 , 1548.34 , 0.200587
16 , 1599.24 , 0.14299
17 , 1650.13 , 0.0982695
18 , 1701.03 , 0.0651413
19 , 1751.92 , 0.0416967
20 , 1802.82 , 0.0257984
21 , 1853.71 , 0.0154531
22 , 1904.61 , 0.00897344
23 , 1955.5 , 0.00506102
24 , 2006.39 , 0.00277696

The risk measure is: 1368.18

The analysis in "myRiskCostModel" completed in 152.31 seconds.

At optimization step 7 the objective function in "myRiskOptimizationCostModel" is 1368.18

The decision variable values (before updating) are:

| 0.149299 |

The optimization gradient is currently:

| -9924.72 |

The norm in "myOptimizationConvergenceCheck" is 9924.72

Now running the initial sampling to determine thresholds...

The sampling mean-value is:

1272.44

The FORM analysis in RISK at threshold "799.887" has started..

For exceedence probability curve (threshold number, threshold, probability):

1 , 756.928 , 0.995887

2 , 799.887 , 0.99257

3 , 842.846 , 0.987083

4 , 885.805 , 0.978456

5 , 928.764 , 0.965373

6 , 971.723 , 0.94641

7 , 1014.68 , 0.920067

8 , 1057.64 , 0.884998

9 , 1100.6 , 0.840251

10 , 1143.56 , 0.785517

11 , 1186.52 , 0.721325

12 , 1229.48 , 0.649128

13 , 1272.44 , 0.571239

14 , 1315.4 , 0.49062

15 , 1358.35 , 0.410564

16 , 1401.31 , 0.33422

17 , 1444.27 , 0.264337

18 , 1487.23 , 0.202902

19 , 1530.19 , 0.15102

20 , 1573.15 , 0.108915

21 , 1616.11 , 0.0760691

22 , 1659.07 , 0.0514298

23 , 1702.03 , 0.0336498

24 , 1744.99 , 0.0213028

25 , 1787.95 , 0.0130482

26 , 1830.9 , 0.00773371

27 , 1873.86 , 0.00443503

The risk measure is: 1309.9

The analysis in "myRiskCostModel" completed in 165.379 seconds.

At optimization step 8 the objective function in "myRiskOptimizationCostModel" is 1309.9

The decision variable values (before updating) are:

| 0.179159 |

The optimization gradient is currently:

| 463.061 |

The norm in "myOptimizationConvergenceCheck" is 463.061

Now running the initial sampling to determine thresholds...

The sampling mean-value is:

1201.57

The FORM analysis in RISK at threshold "436.831" has started..

For exceedence probability curve (threshold number, threshold, probability):

1	, 436.831	, 0.999988
2	, 506.353	, 0.999947
3	, 575.874	, 0.999798
4	, 645.396	, 0.99931
5	, 714.918	, 0.997887
6	, 784.439	, 0.994188
7	, 853.961	, 0.985616
8	, 923.483	, 0.967894
9	, 993.004	, 0.935187
10	, 1062.53	, 0.881261
11	, 1132.05	, 0.801788
12	, 1201.57	, 0.696986
13	, 1271.09	, 0.573216
14	, 1340.61	, 0.442181
15	, 1410.13	, 0.317682
16	, 1479.66	, 0.211386
17	, 1549.18	, 0.129758
18	, 1618.7	, 0.0732669
19	, 1688.22	, 0.0379927
20	, 1757.74	, 0.0180762
21	, 1827.26	, 0.0078909
22	, 1896.79	, 0.00316212
23	, 1966.31	, 0.00116498

The risk measure is: 1311.07

The analysis in "myRiskCostModel" completed in 147.4 seconds.

At optimization step 9 the objective function in "myRiskOptimizationCostModel" is 1311.07

The decision variable values (before updating) are:

| 0.176844 |

The optimization gradient is currently:

| -23.6451 |

The norm in "myOptimizationConvergenceCheck" is 23.6451

Now running the intial sampling to determine thresholds...

The sampling mean-value is:

1302.9

The FORM analysis in RISK at threshold "491.8" has started..

For exceedence probability curve (threshold number, threshold, probability):

1	, 491.8	, 0.99996
2	, 565.536	, 0.999833
3	, 639.272	, 0.999376
4	, 713.008	, 0.997942
5	, 786.744	, 0.993988
6	, 860.48	, 0.984399
7	, 934.216	, 0.963942
8	, 1007.95	, 0.925523
9	, 1081.69	, 0.861935
10	, 1155.42	, 0.769137
11	, 1229.16	, 0.649533
12	, 1302.9	, 0.513312
13	, 1376.63	, 0.376037
14	, 1450.37	, 0.253419
15	, 1524.1	, 0.156248

16 , 1597.84 , 0.0877811
17 , 1671.58 , 0.0448277
18 , 1745.31 , 0.0207849
19 , 1819.05 , 0.00874595
20 , 1892.78 , 0.00334271
21 , 1966.52 , 0.00116177
22 , 2040.26 , 0.000368003
23 , 2113.99 , 0.000106563

The risk measure is: 1311.12

The analysis in "myRiskCostModel" completed in 142.001 seconds.

At optimization step 10 the objective function in "myRiskOptimizationCostModel" is 1311.12

The decision variable values (before updating) are:

| 0.176962 |

The optimization gradient is currently:

| 4.40024 |

The norm in "myOptimizationConvergenceCheck" is 4.40024

The analysis in "myRiskOptimizationCostModel" completed in 1521.39 seconds after 10 iterations.

The decision variable values are:

| 0.176962 |

B.2 Nodal Optimization Output

The OPTIMIZATION analysis in "myRiskOptimizationCostModel" started...

The decision variable values initially are:

| 0.7 |

Remeshing...

The RISK analysis in "myRiskCostModel" started...

Now running the initial sampling to determine thresholds...

The random variable named "cantileverBeamPlateDensity" is missing the probability distributions tool,

hence, it is creating a tool called "defaultInHouseProbabilityDistributions"

The random variable named "cantileverBeamPlateYoungsModulus" is missing the probability distributions tool,

hence, it is picking up the "defaultInHouseProbabilityDistributions" that is already available.

The sampling mean-value is:

1004.92

Determining thresholds using points within 5 standard deviations from the mean.

The threshold 1 is 634.338

The threshold 2 is 668.028

The threshold 3 is 701.717

The threshold 4 is 735.406

The threshold 5 is 769.096

The threshold 6 is 802.785

The threshold 7 is 836.474

The threshold 8 is 870.163

The threshold 9 is 903.853

The threshold 10 is 937.542

The threshold 11 is 971.231

The threshold 12 is 1004.92

The threshold 13 is 1038.61

The threshold 14 is 1072.3

The threshold 15 is 1105.99

The threshold 16 is 1139.68

The threshold 17 is 1173.37

The threshold 18 is 1207.06

The threshold 19 is 1240.75

The threshold 20 is 1274.44

The threshold 21 is 1308.12

The threshold 22 is 1341.81

The threshold 23 is 1375.5

The FORM analysis in RISK at threshold "634.338" has started..

For exceedence probability curve (threshold number, threshold, probability):

1 , 634.338 , 0.999818

2 , 668.028 , 0.999231

3 , 701.717 , 0.997472

4 , 735.406 , 0.992944

5 , 769.096 , 0.983086

6 , 802.785 , 0.964475

7 , 836.474 , 0.933504

8 , 870.163 , 0.887399

9 , 903.853 , 0.825207

```

10 , 937.542 , 0.748344
11 , 971.231 , 0.660449
12 , 1004.92 , 0.566655
13 , 1038.61 , 0.472551
14 , 1072.3 , 0.383202
15 , 1105.99 , 0.302454
16 , 1139.68 , 0.232637
17 , 1173.37 , 0.174626
18 , 1207.06 , 0.128117
19 , 1240.75 , 0.0920114
20 , 1274.44 , 0.0647866
21 , 1308.12 , 0.0447947
22 , 1341.81 , 0.0304497
23 , 1375.5 , 0.0203791
24 , 1409.19 , 0.0134457
25 , 1442.88 , 0.00875553
26 , 1476.57 , 0.00563327
27 , 1510.26 , 0.00358474

```

The risk measure is: 1039.71

The analysis in "myRiskCostModel" completed in 150.531 seconds.

At optimization step 1 the objective function in "myRiskOptimizationCostModel" is 1039.71

The decision variable values (before updating) are:

```
| 0.7 |
```

The optimization gradient is currently:

```
| -1262.8 |
```

The norm in "myOptimizationConvergenceCheck" is 1262.8

Remeshing...

Now running the initial sampling to determine thresholds...

The sampling mean-value is:

1092.86

The FORM analysis in RISK at threshold "622.284" has started..

For exceedence probability curve (threshold number, threshold, probability):

```

1 , 622.284 , 0.999784
2 , 665.064 , 0.998504
3 , 707.843 , 0.993338
4 , 750.622 , 0.977566
5 , 793.401 , 0.941252
6 , 836.181 , 0.874994
7 , 878.96 , 0.775951
8 , 921.739 , 0.651027
9 , 964.519 , 0.514741
10 , 1007.3 , 0.383419
11 , 1050.08 , 0.269669
12 , 1092.86 , 0.179752
13 , 1135.64 , 0.114059
14 , 1178.42 , 0.0692195
15 , 1221.19 , 0.040368
16 , 1263.97 , 0.022715
17 , 1306.75 , 0.0123839
18 , 1349.53 , 0.00656609
19 , 1392.31 , 0.00339668
20 , 1435.09 , 0.00171954
21 , 1477.87 , 0.00085415

```

22 , 1520.65 , 0.000417306
23 , 1563.43 , 0.000200952
The risk measure is: 978.65
The analysis in "myRiskCostModel" completed in 132.532 seconds.

At optimization step 2 the objective function in "myRiskOptimizationCostModel" is 978.65

The decision variable values (before updating) are:

| 0.82628 |

The optimization gradient is currently:

| -707.401 |

The norm in "myOptimizationConvergenceCheck" is 707.401

Remeshing...

Now running the initial sampling to determine thresholds...

The sampling mean-value is:

884.863

The FORM analysis in RISK at threshold "543.2" has started..

For exceedence probability curve (threshold number, threshold, probability):

1 , 543.2 , 0.999998
2 , 574.261 , 0.999983
3 , 605.321 , 0.999896
4 , 636.381 , 0.999487
5 , 667.441 , 0.997976
6 , 698.501 , 0.993469
7 , 729.562 , 0.982399
8 , 760.622 , 0.959557
9 , 791.682 , 0.919249
10 , 822.742 , 0.857407
11 , 853.802 , 0.773656
12 , 884.863 , 0.672181
13 , 915.923 , 0.560819
14 , 946.983 , 0.448896
15 , 978.043 , 0.344871
16 , 1009.1 , 0.254682
17 , 1040.16 , 0.181158
18 , 1071.22 , 0.124426
19 , 1102.28 , 0.0827362
20 , 1133.34 , 0.0534026
21 , 1164.4 , 0.0335452
22 , 1195.46 , 0.0205575
23 , 1226.52 , 0.0123191
24 , 1257.58 , 0.00723492
25 , 1288.65 , 0.00417135

The risk measure is: 940.211

The analysis in "myRiskCostModel" completed in 140.086 seconds.

At optimization step 3 the objective function in "myRiskOptimizationCostModel" is 940.211

The decision variable values (before updating) are:

| 0.991536 |

The optimization gradient is currently:

| -269.989 |

The norm in "myOptimizationConvergenceCheck" is 269.989

Remeshing...

Now running the initial sampling to determine thresholds...

The sampling mean-value is:

952.356

The FORM analysis in RISK at threshold "474.981" has started..

For exceedence probability curve (threshold number, threshold, probability):

1	, 474.981	, 1
2	, 518.379	, 1
3	, 561.777	, 0.999996
4	, 605.174	, 0.999929
5	, 648.572	, 0.999305
6	, 691.97	, 0.995564
7	, 735.367	, 0.980605
8	, 778.765	, 0.938733
9	, 822.163	, 0.853051
10	, 865.56	, 0.719508
11	, 908.958	, 0.554938
12	, 952.356	, 0.38913
13	, 995.753	, 0.24845
14	, 1039.15	, 0.145306
15	, 1082.55	, 0.0784869
16	, 1125.95	, 0.0395102
17	, 1169.34	, 0.018705
18	, 1212.74	, 0.008394
19	, 1256.14	, 0.00359881
20	, 1299.54	, 0.00148379
21	, 1342.93	, 0.000591961
22	, 1386.33	, 0.000229596
23	, 1429.73	, 8.69831e-05

The risk measure is: 929.596

The analysis in "myRiskCostModel" completed in 136.315 seconds.

At optimization step 4 the objective function in "myRiskOptimizationCostModel" is 929.596

The decision variable values (before updating) are:

| 1.12653 |

The optimization gradient is currently:

| -49.2708 |

The norm in "myOptimizationConvergenceCheck" is 49.2708

Remeshing...

Now running the intial sampling to determine thresholds...

The sampling mean-value is:

950.429

The FORM analysis in RISK at threshold "645.637" has started..

For exceedence probability curve (threshold number, threshold, probability):

1	, 645.637	, 0.999466
2	, 673.345	, 0.998071
3	, 701.053	, 0.994249
4	, 728.762	, 0.98513
5	, 756.47	, 0.966506
6	, 784.179	, 0.933334
7	, 811.887	, 0.881292
8	, 839.595	, 0.808548
9	, 867.304	, 0.71705

```

10 , 895.012 , 0.612473
11 , 922.72 , 0.502873
12 , 950.429 , 0.396665
13 , 978.137 , 0.300778
14 , 1005.85 , 0.219533
15 , 1033.55 , 0.154572
16 , 1061.26 , 0.105192
17 , 1088.97 , 0.0693675
18 , 1116.68 , 0.0444355
19 , 1144.39 , 0.0277164
20 , 1172.1 , 0.0168729
21 , 1199.8 , 0.0100493
22 , 1227.51 , 0.00586534
23 , 1255.22 , 0.00336194

```

The risk measure is: 929.53

The analysis in "myRiskCostModel" completed in 132.779 seconds.

At optimization step 5 the objective function in "myRiskOptimizationCostModel" is 929.53

The decision variable values (before updating) are:

```
| 1.15117 |
```

The optimization gradient is currently:

```
| -15.0959 |
```

The norm in "myOptimizationConvergenceCheck" is 15.0959

Remeshing...

Now running the initial sampling to determine thresholds...

The sampling mean-value is:

```
977.628
```

The FORM analysis in RISK at threshold "655.419" has started..

For exceedence probability curve (threshold number, threshold, probability):

```

1 , 655.419 , 0.999172
2 , 684.711 , 0.996975
3 , 714.002 , 0.991025
4 , 743.294 , 0.97715
5 , 772.586 , 0.949763
6 , 801.878 , 0.90307
7 , 831.169 , 0.833486
8 , 860.461 , 0.74169
9 , 889.753 , 0.633254
10 , 919.045 , 0.517304
11 , 948.336 , 0.403949
12 , 977.628 , 0.301682
13 , 1006.92 , 0.215835
14 , 1036.21 , 0.148271
15 , 1065.5 , 0.0980654
16 , 1094.8 , 0.0626141
17 , 1124.09 , 0.038723
18 , 1153.38 , 0.0232563
19 , 1182.67 , 0.0136014
20 , 1211.96 , 0.00776656
21 , 1241.25 , 0.00434014
22 , 1270.55 , 0.00237838
23 , 1299.84 , 0.00128124

```

The risk measure is: 929.448

The analysis in "myRiskCostModel" completed in 132.997 seconds.

At optimization step 6 the objective function in "myRiskOptimizationCostModel" is 929.448
The decision variable values (before updating) are:
| 1.15871 |
The optimization gradient is currently:
| -5.76984 |
The norm in "myOptimizationConvergenceCheck" is 5.76984
The analysis in "myRiskOptimizationCostModel" completed in 866.808 seconds after 6 iterations.

The decision variable values are:
| 1.15871 |