

An Adaptive Response Surface Method for Engineering Analysis

Irfan Kaymaz

A thesis submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Engineering, Department of Mechanical Engineering.

November, 2000

Abstract

Uncertainty evaluation in structural analysis has been gaining more interest recently, and several methods have been developed to carry out the evaluation. Although these methods have been applied to various problems in different engineering fields varying from aerospace to coastal engineering, the conventional methods for structural reliability analysis have the limitation that they are computationally expensive, to the extent that substantial simplifications have to be applied, and for some problems it is impossible to apply the methods.

Approximation methods such as the Response Surface Method (RSM), in which an approximation function is used to model the performance of the structure under study, can be used to alleviate performance limitations. For structural reliability analysis, the common approach in the RSM is to use a second-degree polynomial as a response surface function, but in many reliability problems this may not be the best choice. In the present thesis an adaptive response surface method, called ADAPRES, is developed, in which the formation of the response surface function starts with a linear function and higher terms are added according to an assessment of modelling error.

Uncertainty evaluation of a real engineering system often involves many different analyses with the output from one analysis forming the input for another, and interaction and communication between analyses is crucial to the overall evaluation. For each analysis, a response surface function can be used in place of the actual performance function. In order to carry out different analyses efficiently, formation of the response surface function for each analysis step can be carried out concurrently as well as being distributed to computing workstations in a network provided that the problem permits. Therefore, a network tool, called ADAPRES_NET, has been developed, by which the time consuming evaluation of the performance function can be distributed on computing workstations over a network as well as handling communication and data transformation between necessary applications. ADAPRES_NET models the dynamic relationship between computing processes using Petri nets.

The applicability and efficiency of ADAPRES and ADAPRES_NET have been demonstrated for example analyses taken from the literature, and for a case problem which includes the optimisation of a connecting rod subjected to uncertain problem parameters. The example applications show that ADAPRES can give a reduction in the computational effort required while leading to equally or more satisfactory results compared with standard techniques. The suitability of the approach for engineering structural reliability assessment is evaluated, and suggestions are made for its future development.

Acknowledgements

I would like to offer my thanks to my supervisor Mr. Chris McMahon, for his guidance and encouragement throughout the work. I could not have asked for more patient or helpful supervision.

This PhD study was financed by Ataturk University in Turkey. I am grateful to for the support.

My thanks also go to all those at 83 Woodland Road (past and present) for helping to provide a wonderful working environment. In particular, I would like to thank Rose Crossland whose help and advice throughout my PhD were invaluable, and David Brittan whom I could not have done without (particularly during the early days of learning C). I am grateful to Kwanghoon Lee and Yaowu Liu whom I shared the same room with, especially for their understanding when I was frustrated because of the errors in the C programs codes.

I would also like to thank all researchers who put their works on the Internet, especially to André L. Tits from Electrical Engineering and the Institute for Systems Research in the University of Maryland, who kindly provided me with the optimisation program, CFSQP.

I would also like to thank the Turkish community in Bristol for providing a friendly environment.

Finally, I would like to thank my wife for her constant support and patience in bringing up two lovely kids, especially during the last few months of this thesis writing.

To my father

Declaration

The accompanying dissertation entitled “An Adaptive Response Surface Method for Engineering Analysis” is based on work carried out by the author at the University of Bristol between October 1996 and November 2000.

All of the work and ideas in this dissertation are original unless otherwise acknowledged either in the text, or by reference.

This work has not been submitted for a degree or diploma at this or any other University.

The views expressed in this dissertation are those of the author and not those of the University.

Signed

Date

Notations and Abbreviations

Notations

β	safety index (reliability index)
β_{RF}	safety index from Rackwitz-Fiessler method
β_{sim}	safety index from Monte Carlo simulation
$\mu()$	membership function
μ_x	mean value of random variable x
$\Phi()$	standard distribution function
$\Pi()$	possibility distribution
σ_x	standard deviation of random variable x
Adj_R^2	adjusted coefficient of determination
b	coefficients of response surface function
<i>COV</i>	coefficient of variation
$F(\mathbf{X})$	objective function
$F_x(x)$	cumulative distribution of random variable x
$f_x(x)$	probability density distribution of random variable x
f_i	an arbitrary factor for a response surface function
$g(\mathbf{X})$	performance function
Perc_dif	percentage difference
SS_E	error or residual sum of squares
SS_R	regression sum of squares
S_{yy}	total sum of squares
U	a vector including transformed random variables
$Var()$	Variance
X	random variable
X'	transformed (reduced) random variable
x	an instance of random variable x
$\bar{y}$	response function

Abbreviations

<i>ADAPRES</i>	adaptive response surface method
<i>CAD</i>	computer-aided design
<i>CAE</i>	computer-aided engineering
<i>CAM</i>	computer-aided manufacturing
<i>CCD:</i>	central composite design
<i>DACE</i>	design and analysis of computer experiments
<i>DoE</i>	design of experiment
<i>FEA</i>	finite element analysis
<i>FORM</i>	first-order reliability method
<i>FOSM</i>	first-order second-moment method
<i>MCS</i>	Monte Carlo simulation
<i>MDO</i>	multidisciplinary design optimisation
<i>MSE</i>	mean squared error
<i>MVFSOM</i>	mean value first-order moment method
<i>POF</i>	probability of failure
<i>RMS</i>	root mean squared error
<i>RSF</i>	response surface function
<i>RSM</i>	response surface method
<i>STEP</i>	standard for the exchange of product data

Glossary

Central Composite Design

A design of experiments method by which the experimental points are selected especially for a second-order response surface.

Client/Server

A computational architecture that involves client processes requesting services from server processes.

Design of Experiments

A methodology to choose experimental points.

Design point

Most probable failure point on the limit state.

EXPRESS

Information modelling language, describing the characteristics of information.

Failure state

A region in which the component or system fails for all realisations of the random variables of the problem.

IDEF0

IDEF is an acronym for ICAM DEFinition, which is an acronym for Integrated Computer Aided Manufacturing--an initiative conducted by the U.S. Air Force in the 1970s. IDEF0 is a methodology to assist with the analysis and documentation of the activities and processes in an engineering organisation.

Limit state

A surface dividing the basic random variable space into two regions namely a failure region and a safe region.

Monte Carlo simulation

A simulation method which randomly generates values for uncertain variables a great number of times to simulate a model.

Objective function

A function associated with an optimisation problem that determines how good a solution is.

Performance function

A function including the functional relationship of the basic variables of a component or a system.

Petri net

A formal and graphically appealing language which is appropriate for modelling systems with concurrency.

Random numbers

A series of randomly generated numbers which is distributed uniformly between 0 and 1.

Random variable

A function that associates a unique numerical value with every outcome of an experiment.

RBSO

Reliability based structural optimisation.

Region of interest

A region from which most of the contribution to the computation of the probability of failure is obtained.

Reliability index

The minimum distance from the limit state to the origin of the transformed space.

Response surface function

A function which is formed to represent a real performance function.

Safe state

A region in which the component or system functions without failure for all realisations of the random variables of the problem.

Socket

A special file which provides a way to communicate with other programs using standard Unix file descriptors.

Transformed space

A co-ordinate system in which random variables are normalised with their mean value and standard deviation.

Transformed (reduced) variable

Representation of random variables in a transformed space.

Contents

1. INTRODUCTION.....	1-1
1.1 ASPECTS OF ENGINEERING DESIGN.....	1-2
1.2 UNCERTAINTY IN ENGINEERING DESIGN	1-4
1.2.1 <i>The Classification of Uncertainties in Engineering Design</i>	1-5
1.2.2 <i>Uncertainties in Structural Reliability Analysis</i>	1-6
1.3 OBJECTIVES OF THE THESIS	1-9
1.4 OUTLINE OF THE THESIS.....	1-11
 2. THE ENGINEERING DESIGN PROCESS.....	 2-1
2.1 AN OVERVIEW OF ENGINEERING DESIGN	2-2
2.2 DESIGN AND THE DESIGN PROCESS.....	2-6
2.3 MODELLING THE DESIGN PROCESS	2-7
2.3.1 <i>Modelling Techniques</i>	2-9
2.3.1.1 Data Flow Diagram:.....	2-9
2.3.1.2 IDEF:	2-10
2.3.1.3 STEP and EXPRESS:	2-12
2.4 PETRI NETS.....	2-13
2.4.1 <i>Basic Notations</i>	2-13
2.4.2 <i>Marking</i>	2-14
2.4.3 <i>Firing of a Transition</i>	2-14
2.4.4 <i>The Process Controller of Petri Nets</i>	2-15
2.4.5 <i>An Example of Petri Nets</i>	2-15
2.4.6 <i>Comparison of Petri Net with Other Techniques</i>	2-16
2.5 DESIGN INTEGRATION VIA OPTIMISATION	2-18

2.6	CFSQP: AN OPTIMISATION TOOL	2-24
2.6.1	<i>The Key Features of CFSQP</i>	2-25
2.7	DISTRIBUTED COMPUTING TECHNIQUES.....	2-25
2.8	CONCLUSION	2-27
3.	STRUCTURAL RELIABILITY ANALYSIS.....	3-1
3.1	REPRESENTATION OF UNCERTAINTY	3-2
3.1.1	<i>Interval Mathematics</i>	3-2
3.1.2	<i>Fuzzy Theory</i>	3-3
3.1.3	<i>Probability Theory</i>	3-6
3.2	OVERVIEW OF STRUCTURAL RELIABILITY METHODS	3-7
3.2.1	<i>General Review of First-Order Second Moment Methods</i>	3-8
3.2.2	<i>Review for the SORM</i>	3-10
3.2.3	<i>Review of the Monte Carlo Methods</i>	3-12
3.3	STRUCTURAL RELIABILITY METHODS: DETAILED DESCRIPTION	3-13
3.3.1	<i>Stress-Strength Interference Method</i>	3-14
3.4	THE FIRST-ORDER SECOND-MOMENT CODE FORMULATION (FOSM)	3-18
3.4.1	<i>The Hasofer-Lind Method</i>	3-19
3.4.2	<i>Linear Performance Function</i>	3-20
3.4.3	<i>Non-linear Performance Function</i>	3-21
3.4.4	<i>First-Order Reliability Method (FORM)</i>	3-22
3.4.5	<i>Second-Order Reliability Method (SORM)</i>	3-23
3.5	MONTÉ CARLO SIMULATION	3-25
3.5.1	<i>Generation of Numbers Required for MCS</i>	3-27
3.5.2	<i>A Particular Approach to Generating Normally Distributed Random Variables</i>	3-30

3.6	VARIANCE REDUCTION TECHNIQUES.....	3-32
3.6.1	<i>Importance Sampling (IS)</i>	3-32
3.7	COMPARISON OF THE FORM RESULTS WITH THAT OF MCS	3-33
3.8	CONCLUSION	3-35
4.	RESPONSE SURFACE METHODS	4-1
4.1	OVERVIEW OF THE RESPONSE SURFACE METHODS.....	4-2
4.2	THEORETICAL BACKGROUND OF THE RESPONSE SURFACE METHODS	4-7
4.1.1	<i>Bucher's Response Surface Method</i>	4-8
4.1.2	<i>Rajashekhar's Response Surface Method</i>	4-9
4.1.3	<i>Kim's Response Surface Method</i>	4-12
4.3	MODELLING ERROR ASSESSMENT AND MODEL CHOICE FOR A RSF	4-15
4.4	REVIEW OF ISSUES IN RESPONSE SURFACE METHODS	4-17
4.5	NEURAL NETWORKS	4-18
4.6	CONCLUSION	4-21
5.	THE ADAPTIVE RESPONSE SURFACE METHOD.....	5-1
5.1	BASIC STEPS OF ADAPRES.....	5-2
5.1.1	<i>Design of Experiments (DoE)</i>	5-5
5.1.2	<i>Determining the Value of the Arbitrary Function</i>	5-6
5.1.3	<i>The Effect of The Random Variables on The Performance Function</i>	5-8
5.1.3.1	Generating the Design Matrix Dynamically	5-10
5.1.4	<i>The Statistical Criteria Used in ADAPRES</i>	5-11
5.1.5	<i>Generating the Terms in a RSF</i>	5-13
5.2	RELIABILITY ASSESSMENT IN ADARES.....	5-15

5.2.1	<i>The FORM in ADAPRES</i>	5-16
5.2.2	<i>MCS in ADAPRES</i>	5-17
5.3	REDUCING THE STAGES OF FORMING RSF	5-18
5.4	EXAMPLES	5-19
5.4.1	<i>Example 1</i>	5-19
5.1.2	<i>Example 2</i>	5-21
5.1.3	<i>Example 3</i>	5-22
5.5	CONCLUSION	5-26
6.	A TOOL MANAGEMENT SYSTEM: ADAPRES_NET	6-1
6.1	INTRODUCTION	6-2
6.2	GENERAL STRUCTURE OF ADAPRES_NET	6-2
6.3	FUNDAMENTALS OF NETWORKING	6-3
6.3.1	<i>Client/Server Concepts</i>	6-4
6.3.2	<i>Socket Programming</i>	6-6
6.3.2.1	<i>Client/Server Based Network Programming</i>	6-7
6.3.3	<i>Remote Shell Invocation</i>	6-8
6.3.4	<i>Concurrent Process Calls</i>	6-9
6.4	FEATURES OF ADAPRES_NET	6-11
6.4.1	<i>Preparing the Data Files</i>	6-13
6.4.2	<i>Server Side Functions of ADAPRES_NET</i>	6-15
6.4.3	<i>RemoteSend</i>	6-16
6.4.4	<i>Remote Invocation of the Programs on The Network</i>	6-17
6.4.5	<i>RemoteReceive</i>	6-17
6.4.6	<i>Client Side Functions of ADAPRES_NET</i>	6-18
6.5	CONCLUSION	6-19

7. CASE STUDY: AN APPLICATION OF ADAPRES_NET TO AN ENGINEERING PROBLEM.....	7-1
7.1 PROBLEM DESCRIPTION	7-2
7.2 PARAMETRIC MODELLING OF THE CONNECTING ROD	7-3
7.2.1 FEA Model of the Connecting Rod	7-4
7.2.2 Applied Loads	7-5
7.3 APPLIED ANALYSES: STRESS ANALYSIS AND BUCKLING ANALYSIS.....	7-6
7.3.1 Stress Analysis.....	7-6
7.3.2 Buckling Analysis of the Connecting Rod	7-7
7.3.2.1 Euler Buckling Analysis	7-7
7.3.2.2 Building the Model	7-9
7.3.2.3 Obtaining the Static and the Buckling Solution	7-10
7.3.2.4 Expanding the Solution and Reviewing the Result	7-10
7.4 RESPONSE SURFACE APPROXIMATION BY ADAPRES.....	7-10
7.5 DISTRIBUTING THE JOBS BY ADAPRES_NET	7-11
7.5.1 Tasks in Main_Controller:.....	7-11
7.5.2 Tasks in Workstation_1(Brunel):	7-12
7.5.3 Tasks in Workstation_2(Athena):.....	7-13
7.6 RESULTS AND DISCUSSION	7-13
7.7 CONCLUSION	7-18
 8. CONCLUDING REMARKS AND FUTURE WORKS	 8-1
8.1 DEVELOPMENT IN THE PRESENT THESIS	8-2
8.2 SUMMARY OF CONCLUSIONS	8-3
8.3 LIMITATIONS	8-4

8.3.1	<i>Limitations of Structural Reliability Analysis</i>	8-4
8.3.2	<i>Limitations of ADAPRES</i>	8-5
8.3.3	<i>Limitations of ADAPRES_NET</i>	8-6
8.4	FUTURE WORKS.....	8-6
8.4.1	<i>Future Works on the Structural Reliability Methods</i>	8-6
8.4.2	<i>Future Works on ADAPRES</i>	8-7
8.4.3	<i>Future Works on ADAPRES_NET</i>	8-7
8.5	PUBLICATIONS ARISING FROM WORK REPORTED IN THIS THESIS.....	8-8
	REFERENCES.....	REF_1

List of Appendices

A. THE PROCESS CONTROLLER IN THE PETRI NET TOOL MANAGER	A-1
B. THE STRUCTURE OF ADAPRES	B-1
C. APDL FILES FOR THE BUCKLING ANALYSIS AND STRESS ANALYSIS	C-1

Chapter 1

Introduction

Engineering design is the process of devising a system, component, or process to meet some desired needs. The design process is related to the process of actions and decisions that are taken during design in order to arrive at a completed product design. The engineering design process is a complex process which is often complicated by the presence of uncertainties. The first section of this chapter introduces the aspects of engineering design in which recent developments in engineering design analysis are also explored. In the second section, uncertainties occurring in engineering design are represented in general, and a classification of the related uncertainties is given along with more specific consideration of uncertainties that occur in structural reliability analysis. Objectives of the present thesis are stated in the third section. In the final section of this chapter, the subjects included in the thesis are outlined.

1.1 Aspects of Engineering Design

Engineering design is the process of devising a system, component, or process to meet some desired needs. It is a decision-making process (often iterative), in which basic sciences, mathematics, and engineering sciences are applied to convert resources optimally to meet a stated goal. Various methods can be used to achieve the design goals [Abet 2000]. One approach might be using experimentation where several designs of the component are built and tested. The design providing maximum economy while satisfying the constraints on the design would be chosen for the final product. It is clear that this procedure is effective but it is expensive and time-consuming, and because testing happens at the end of the design process it is difficult to make changes to a design that fails under test. Therefore, a second approach has been developed over the years, based on the definition of the design process analytically and then obtaining the best solution using differential calculus or the calculus of variations. Although this is a more attractive procedure than the first one, it is rarely possible in practical applications to obtain a direct analytical solution to a design problem because of the complexity of the design and analysis problem. Thus, various approximation methods have been developed such as finite element method, the finite difference method, simulation methods, by which engineering problems are simplified in order to make a judgement about the performance of the proposed algorithm get an approximate solutions while satisfying the criteria set for the design. These type of analyses are based on imperfect understanding of applied loads, of material properties and behaviour etc, and therefore assessment is generally carried out today by means of deterministic analysis with factors of safety applied to take into account the uncertainty. Designs made with factors of safety are often overdesigned (indeed, this is often true for designed checked by testing also). Overdesign is a problem in a number of aspects. Today there is a great pressure to take weight out of mechanical artefacts in order to save the cost of materials, but also to allow reductions in energy consumption (a 10% reduction in the structural mass of a vehicle will allow a smaller engine, smaller fuel tank etc and might save £1000+ in fuel costs in the life of a car). Two techniques that are seen as being very important in achieving improved designs are probabilistic design and optimisation. Probabilistic design enables the designer to design for a specific reliability or for a specific

proportion of product to be produced within specification; and hence maximise safety, quality and economy. On the other hand optimisation involves techniques for the automation of the required computations to search for improved design. The former allows a better understanding of uncertainty to be applied, and the latter allows the “best” design to be identified within certain constraints.

Probabilistic design is also important in design areas where exhaustive testing is difficult or impossible, but where the integrity of a design over its lifetime is very important, as in civil, structural and offshore engineering. It is also very important when the loads to which a design will be subject are uncertain (e.g. storm, wind or wave loads in civil engineering, uncertain road or driver loads in automotive engineering). As Paul Rubbert, the chief aerodynamicist for Boeing, has pointed out, uncertainty management has become the dominant theme in practical applications like aerospace design [Caltech 1997]. Therefore, the use of probabilistic design in different engineering fields has significantly increased. The works carried out in South West Research Centre in the USA (SWRI), for example, involves using probabilistic analysis in designing critical space Shuttle engine components as well as the blades for gas turbines, high-pressure piping systems, an injector components, and a hot gas transfer tube liner [SWRI 2000]. Probabilistic design has been also found many application fields in coastal engineering because of the random nature of hydraulic loads, and the complexity of structural and environmental responses [PROVERBS 2000].

Beside the above applications of probabilistic design, several special conferences and networks have been established in order to improve the techniques used in probabilistic analysis, such as IASSAR (International Association for Structural Safety and Reliability) [IASSAR 2000] and ESREL (European Safety and Reliability Conference) [ESREL 99], as well as establishing communication between industry using probabilistic design and academics developing the methods used in the domain, such as ASRANET (Integrating Advanced Structural Analysis with Structural Reliability Analysis) [ASRANET 2000].

With the great developments in computer technology that make it possible to store a large amount of data as well as achieving a significant increase in the speed of computers, engineering analyses such as optimisation and probabilistic design that

require extensive computation have been applied to more complex but more realistic problems without having much penalty in the simplification of the representation of the actual engineering problem. In the early days of finite element analysis, for instance, problems were solved by rooms of people processing information through mechanical calculators. Each person received a program sheet with numbers, operations, and fields of results. These results were combined with those of other calculations and passed along in another program sheet to another calculator [Taylor 1992]. Today, however, even a desktop PC is sufficient to run most FEA packages for many problems. For the more complex problems, supercomputers can be used, for which more recently IBM announced a supercomputer that is about 2 million times more powerful than today's top desktop PCs [IBM 2000]. This development in computing capability now means that repetitive analysis required for probabilistic design by means of the technique called Monte Carlo simulation has now become possible to use for engineering problems after the development of computing facilities. The increase in computer power also yields the ability to model more complex shapes by the representation of components of the design by such means as solid models in computer-aided design, as well as allowing more accurate modelling of parts in finite element analysis.

Despite the increase in computing power, however, optimisation and probabilistic analysis are still very computationally expensive approaches for problems including geometrically complex components as well as more realistic analysis types such as non-linear analysis. Therefore, an approximation technique, called the response surface method (RSM), has been used to model the relationship between the performance of the design system and the external loads to which it is subjected, called the performance function, by means of a simple function. Such response surface functions (RSF) have been used for some years [Bucher 1989] [Rajashekhar 1993] but because they usually use quite simple functions they can be a poor match onto the actual performance function. The first contribution of this thesis is addressing this issue.

1.2 Uncertainty in Engineering Design

As pointed out in the previous section of this chapter, in engineering design one of the main objectives is to design a component or a structure with the least cost but with

a specified durability for which many techniques and analytical methods exist along with their combination under the name of optimisation. However, as noted these methods currently mainly deal with the design variables using a deterministic approach which assumes that the value of the design variables has single numerical value. In reality engineering problems are normally complicated by the presence of uncertainties because they include some level of uncertainty due to normal variations or due to uncontrolled or unpredictable effects. Although individual uncertainties relating to a component or the structure may not be so large as to cause failure, the combination of these uncertainties can lead to large and unexpected failures of the structure which might cause catastrophic events such as aircraft accidents which bring about not only loss of a huge amount of money but also much more seriously human lives. Therefore, uncertainty evaluation has become one of the important aspects of engineering design in recent years. In order to deal with uncertainties, first of all, their presence in engineering design should be understood, thus the following section includes a brief classification of uncertainties in an engineering design.

1.2.1 The Classification of Uncertainties in Engineering Design

The classification of uncertainties differs according to which part of the design process is being focused on since the whole design process, which is explained in more detail in Chapter 2, involves many different aspects of uncertainties. Therefore, different classifications have been proposed in the literature, based on author's viewpoint concerning engineering design [McMahon 1994] [Ullman 1995] [Antonsson 1995].

A comprehensive classification of uncertainties in the design process was given by Crossland [Crossland 1997] as:

- Linguistically vagueness in description of design parameters (for example, definitions such as “high speed”, or “low strength”).
- Numerically imprecise or uncertain terms (for example, “loading between 100N and 200N”).
- Categorically imprecise or uncertain expression (for example, “the pack will either be placed on the back or under the saddle”).

- Incomplete information (for example, information concerning some aspect of design may be missing altogether).
- Degrees of abstraction (for example, “power pack” is more abstract than “nickel-cadmium battery” or “fly-wheel battery”).
- Inconsistencies (for example, the target weight is inconsistent with the required power output).

The classification given above is a general classification of uncertainties in the design process. In the present thesis, however, structural reliability analysis can be utilised to represent some of the above uncertainties, which can be given as: numerical uncertainty (Data Uncertainty), categorically uncertain (Occurrence Uncertainty) and incomplete information (Model Uncertainty), which are explained in the following section.

1.2.2 Uncertainties in Structural Reliability Analysis

Most important types of uncertainties in the evaluation of a structural reliability analysis can be classified into three main groups [Siddal 1983] [Toft-Christensen 1982], which are data uncertainty, occurrence uncertainty and model uncertainty, briefly explained below.

1.2.2.1 Data Uncertainty

The input parameters of a structural reliability analysis might involve some uncertainties in their values since most of them are measured values such as the speed of the wind in a wind turbine design or a strength of a material used in a structural analysis, even though they could be measured up to any desired precision. For instance, in fifty tests of ultimate tensile strength of SAE 4340 cold drawn and annealed steel, the outcome of each test is different from other result, ranging from 95.63 psi to 108.430 psi, as given in Table 1.1.

Some of data uncertainties arise from the two sources of error noted by Ditlevsen and Madsen [Ditlevsen 1996]:

- (1) The fluctuation in the value of material properties may be due to uncontrollable factors in the manufacturing process, which are called

inherent attributes. The inherent attributes cannot be eliminated by increasing the number of the experiments determining the properties. These type of uncertainties are classified as irreducible uncertainties [Isukapalli 1999], which is a result of the presence of natural variations.

- (2) Another reason for the fluctuations results from the test equipment which is used to measure the values. This kind of fluctuation can be limited by doing more tests to refine the values, and by investigating the variations in the equipment. This type of uncertainty is classified as reducible uncertainty [Isukapalli 1999].

Another type of data uncertainty can arise if the uncertain parameters are represented by a probability distribution function, based on the set of data used to obtain the parameters of the probability distribution. If a small sample size is selected, the obtained probability distribution may not represent the actual distribution of the uncertain parameter, which is also classified as statistical uncertainty by Thoft-Christensen and Baker [Thoft-Christensen 1982].

1.2.2.2 Occurrence Uncertainty

This type of uncertainty can be considered when an event may or may not occur or when the time of its occurrence may be uncertain. If a random load sequence is taken

Table 1.1: Ultimate tensile strength data of wire specimens of SAE 4340 Cold-drawn and Annealed Steel Wires, adapted from [Haugen 1980].

Ultimate Tensile Strength (psi)									
1	103.779	11	102.325	21	103.633	31	106.540	41	101.453
2	102.906	12	105.377	22	102.616	32	101.744	42	103.779
3	104.796	13	104.796	23	106.831	33	103.799	43	101.162
4	103.197	14	106.831	24	102.325	34	100.145	44	102.470
5	100.872	15	105.087	25	104.651	35	103.799	45	105.232
6	97.383	16	103.633	26	105.807	36	103.488	46	105.337
7	101.162	17	107.848	27	106.395	37	102.906	47	101.744
8	98.110	18	95.63	28	100.872	38	101.017	48	106.104
9	104.651	19	105.831	29	104.360	39	104.651	49	100.726
10	102.906	20	108.430	30	104.360	40	103.197	50	101.598

into account, the timing of the load that the structure encounters can not be determined exactly, and the occurrence of this type of loading is considered uncertain. In structural reliability analysis, for example, when the occurrence of failure is being considered as the major characteristic to assess the reliability of a system, information on whether the system fails under the applied loads or the timing of the occurrence of failure is always important and this event is also uncertain.

1.2.2.3 Modelling Uncertainty

Models showing the relationship between the applied loads on design system and the performance parameters of that design system are simplified representations of the real system being studied. Therefore, various models can be established to represent the system. In structural reliability analysis, for instance, determining the failure condition based on the parameters might be proposed in different forms that can yield different results. Simplification of the real system can also involve uncertainty in the prediction of the result by the comparison of their predictions to those of more detailed models. For example, the most common model uncertainty can occur when a model for the problem is used instead of the actual structure, which is very common in complex structural analysis such as the design of bridge or predicting the actual load acting on the structure. The model which is supposed to have all properties of the actual structure must be designed to counter the loads. However, the common way, as Ditlevsen and Madsen [Ditlevsen 1996] indicate, is to consider only the major loads or boundaries of the structure and neglect the loads and boundaries which are assumed insignificant. Poor models that only have some properties or conditions of actual case can cause model uncertainty.

These major types of uncertainties in engineering design have been of interest especially in the fields where both human lives and a great amount of money can be put in danger such as space programs, aircraft and offshore industries and public transport. Therefore, several methods have been proposed to represent uncertainties in an engineering design process, some of which are represented in Chapter 3.

The main focus of this thesis is to develop a technique to improve the ability to analysis systems in which data and occurrence uncertainties are modelled. Modelling uncertainty is not, in general, addressed in this thesis except in so far that by

improving the efficiency of modelling better models may be used.

1.3 Objectives of the Thesis

The primary objective of the present thesis is to develop a better response surface method for a structural reliability analysis along with forming a probabilistic design system in which the analyses required for a design evaluation may be distributed over a computer network in order to get the benefit of running multiple analyses concurrently.

As will be stated in Chapter 3, the conventional methods for structural reliability analysis have some limitations such as being computationally expensive, even for some problems, it is impossible to apply those methods. Since Monte Carlo simulation, for instance, is a technique that involves repeated execution of a performance function for different values of the input variables in order to develop a representation of variation in performance function. As it requires a large number of samples for the random variables of the structural reliability problem, it may be impossible to use Monte Carlo simulation with a Finite Element Analysis which may take considerable time to carry out the analysis required for a structural reliability evaluation. If, however, an approximation function can be substituted for the real performance function, Monte Carlo simulation or any other structural reliability methods can be utilised to obtain the probability of failure of the system. Some approximation function are called “response surface function”. Conventional response surface methods use a fixed type of a response function for the structural reliability problem such as a linear or a quadratic polynomial. However, for implicit problems in which the relationship of the random variables in the performance function are not defined explicitly, it is almost impossible to assume a fixed type of response function. Therefore, a second-order polynomial has been used widely as the type of the response surface function. However, an adaptive response surface method is developed in the present thesis, in which the formation of the response surface function starts with a linear function and higher terms are added according to an assessment of modelling error. Hence, first objective of the present thesis can be stated as

“the development of computationally efficient as well as more accurate model of

the real performance function in order to use with structural reliability methods such as First Order Method, Second Order Method or Monte Carlo simulation”.

This objective is accomplished via the development of an ADAPtive RESponse surface method, called ADAPRES, in which the response function as well as the design matrix including the experimental points from which the response surface function is formed are determined according to a modelling error evaluation which is explained in depth in Chapter 4.

The second objective of the thesis is to develop a probabilistic design system in which a structural reliability analysis can be handled efficiently in terms of time and accuracy by using one of the distributed computing techniques. Since the reliability evaluation of a real engineering system often involves many different analyses, the interaction and communication with each other are crucial to have a satisfactory design. Fatigue analysis, for example, requires results from stress analysis, for which the response surface function of the stress analysis becomes an input for fatigue analysis. More complicated problems such as designing vehicle involves different analyses such as modal analysis, fatigue analysis, dynamic analysis in which the output from one analysis may form the input for another, stress analysis as are described in Chapter 6. For each analysis, a response surface function can be used in place of the performance function established based on the design constraints. In order to carry out different analyses efficiently, each formation of the response surface function can be carried out concurrently as well as distributing to other workstations over the network provided that the problem permits. Therefore, the second aim of the present thesis can be stated as:

“the implementation of a structural reliability problem in a probabilistic design system in which the evaluation of multiple performance functions can be distributed over a computer network and called sequentially or concurrently as required in order to reduce the time and computational burden of structural reliability analysis”.

To accomplish this objective, a Petri net based network tool, called ADAPRES_NET, has been developed, by which the time consuming evaluation of the performance function can be distributed on the workstations over the network. The communication and data transformation between necessary applications are handled by a Petri net

based design process modelling technique, as is explained in Chapter 2.

It will be seen from the above that the thesis concentrates on the development of a response surface approach for structural reliability analysis. It is believed that the technique may also be extended to be applied to optimisation problems, in which variation in performance with variation in design variables is explored. Therefore, the case study which is given in Chapter 7, is selected as an optimisation problem in which the probabilistic evaluation of the constraints are carried out by means of the proposed response surface method.

1.4 Outline of The Thesis

The thesis is divided into eight chapters, including this one. The contents of the remaining chapters are summarised below.

Chapter 2: The Engineering Design Process

Design and the design process for an engineering problem are described. This is followed by a review of more specific design models especially for multidisciplinary design. Recent design modelling techniques are summarised. The remaining sections in this chapter review the Petri net modelling technique by which ADAPRES_NET is built along with an example of the implementation of Petri nets for an optimisation problem.

Chapter 3: Structural Reliability Analysis

Basic methods applied in structural reliability analysis such as the First Order Second Moment method and Monte Carlo simulation are summarised. Some deficiencies related to those methods are also stated and the need of using an approximation method such as the Response Surface Method is also explained. An algorithm to find the reliability index which is the one of the most important element of the structural reliability methods is also proposed. The comparison of the outcome of the proposed algorithm with that of conventional structural reliability methods is also shown by an example.

Chapter 4: Response Surface Methods

A general overview and review of the response surface approach is given, which is followed by more specific information on the application of the response surface to

structural reliability problems. Some deficiencies of the existing response surface methods for structural reliability analysis are described, along with the necessity of using a more advanced approach that can alleviate those deficiencies.

Chapter 5: The Adaptive Response Surface Method

In this chapter an improved response surface method, ADAPRES, which includes a modeling error assessment, is explained in detail. By this method, the importance of term(s) can be determined without carrying out reliability analysis, which enables ADAPRES to form a RSF which might include cross terms or higher order terms, depending on the modeling error evaluation. ADAPRES also makes use of a design of experiments approach by which the experimental points can be replaced close to the limit state. Two examples are given in the chapter, showing the application of ADAPRES in which better results are obtained than by using a conventional response surface method.

Chapter 6: A Tool Management System: ADAPRES_NET

Network programming is explained in the context of developing a probabilistic design system, called ADAPRES_NET, in which the evaluation of performance functions required by structural reliability analysis can be distributed over the network. In addition, concurrency can be established in order to run the necessary programs on different workstations in parallel, so the computational time spent on the evaluation of each performance function can be reduced.

Chapter 7: Case Study: An application of ADAPRES_NET to an Engineering Problem

In order to show the application of ADAPRES and ADAPRES_NET, a well-known mechanical part, a Connecting Rod, is selected which is subjected to random loading as well as having random material properties. The design problem is assigned as a reliability based structural optimization in which obtaining the minimum volume is selected as the objective function while the constraint functions are formed as probabilistic functions which include stress analysis and linear buckling analysis.

Chapter 8: Concluding Remarks and Further Works

The final chapter summarises the conclusions of the thesis and discusses some limitations of the algorithms developed during the this study. Future directions for the

research following on from thesis are presented. Finally, the publications resulting from this work are listed.

Chapter 2

The Engineering Design Process

In this chapter, engineering design and the engineering design process are explored. In the first section an overview is presented for the definition and representation of engineering design. Recent developments in engineering design evaluation are also described. The second section contains a definition for design and design process while the third section explores methodologies used to model the design process such as Data Flow diagrams, the IDEF modelling method and the EXPRESS language. A more specific modelling technique, Petri nets, is represented in detail in this fourth section since it is used to model the Probabilistic Design System proposed in the present thesis. In the fifth section, integration of different analysis via multidisciplinary structural optimisation is represented briefly along with the optimisation methods. The optimisation tool used in the work is explained in the sixth section, and in the final section, distributed computing techniques are summarised.

2.1 An Overview of Engineering Design

Design can be described as the process of transforming a set of functional specifications and requirements into a complete description of a physical product or system which meets those specifications and requirements [Ertas 1996]. During this transformation process, the design engineer makes decisions about function, shape, material properties, manufacturing technologies etc., based on information provided by handbooks, standards, numeric analyses, company practices, and personal intuition and experience.

The design process is related to the process of actions and decisions that are taken during design in order to arrive at a completed product design. The design process has been analysed by many researchers, [Pahl 1984], [Ullman 1992], [Taguchi 1986], [McMahon 1995], in which decomposition and classification of the design process were represented according to the researchers' particular point of view on engineering design. Each of the representation broadly agrees that design proceeds from clarification of the requirements through explanation of concepts that may meet the requirements (conceptual design), selection of a chosen concept and then refinement and development of this concept for example through embodiment and detail design (detail design).

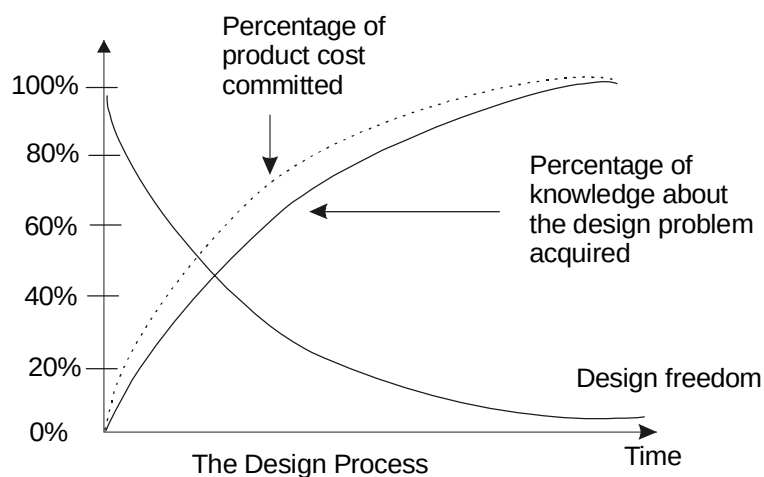


Figure 2.1: Knowledge increases as the design process progresses [Ullman 1992].

The traditional way of controlling the information flow in the product design process is to apply sequential tasks in which the design stages mentioned above are done one after one. However, as Prasad pointed out [Prasad 1995], sequential engineering processes have several disadvantages such as limited information exchange between different stages of the process, and a long elapsed timescale. Design decisions must be made in the early design stages. However, at those stages the product being designed is not very well understood [Chang 1999], and information on the product is accumulated while the design process progresses [Ullman 1992], as illustrated in Figure 2.1.

In a sequential design process, it is said that the design is “thrown over the wall” from the design department to the manufacturing department [Crossland 1997]. In such a process it is difficult to incorporate knowledge and understanding obtained in

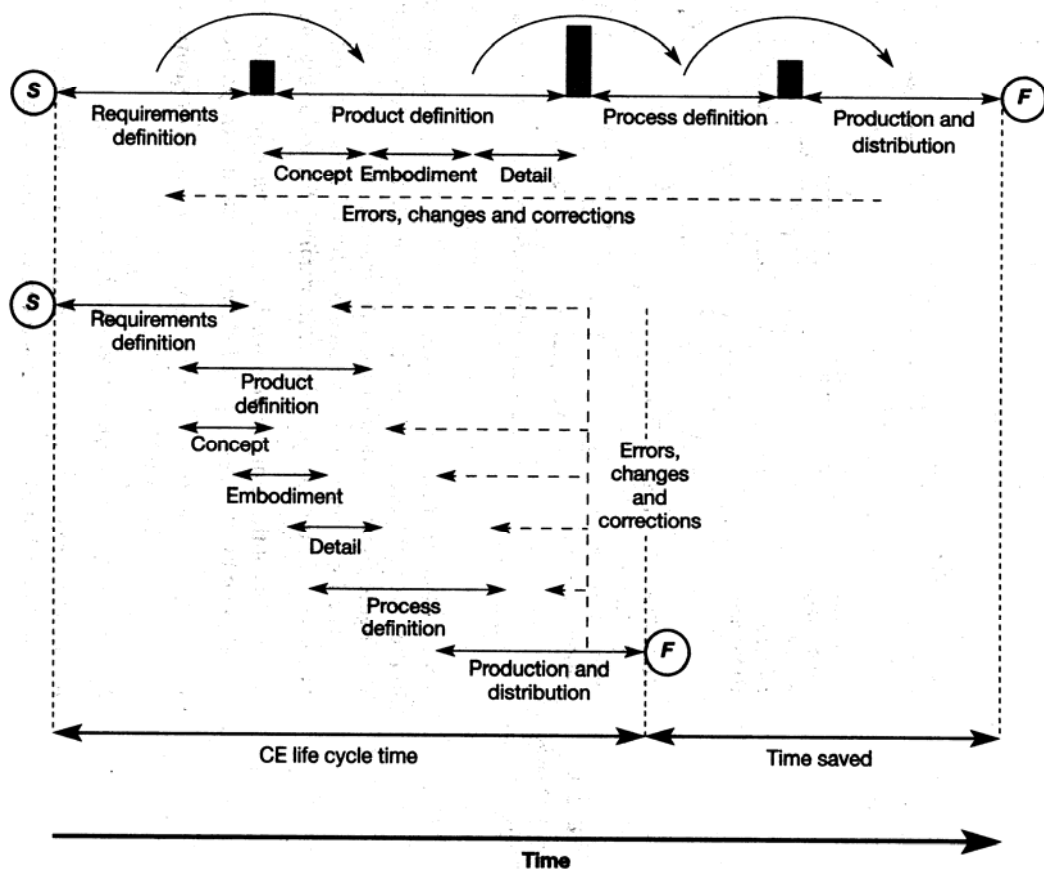


Figure 2.2: Sequential design versus concurrent product design, from Start, S, to Finish, F, taken from [McMahon 1998].

manufacture into the early design process. Therefore a parallel concurrent approach to the modelling and manufacturing the product was suggested under the name of *Concurrent Engineering* [Brooks 1992] [Harkins 1993] [Jacob 1993]. In Concurrent Engineering the detail design considerations and design and planning of the manufacturing processes are begun while the conceptual design is taking place so that manufacturing considerations can be better taken into account in the conceptual design. This approach also shortens the overall time for the process as shown in Figure 2.2.

Although the definition of the Concurrent Engineering can be easily extracted from the term, several definition were also proposed in the literature as follows:

Winner *et al.* [Winner 1988] describes Concurrent Engineering as “a systematic approach to the integrated, concurrent design of products and their related processes including manufacture and support”. Similar descriptions for Concurrent Engineering were made in various references [Jagannathan 1991] [Dean 1992] , which emphasise the following points:

- Parallel and integrated processes
- Integration specialists
- Integrating Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), Computer Aided Engineering (CAE) and analytical tools
- Increasing the performance of the product design

A justification of the use of concurrent engineering can be well represented with the design process model proposed by Ohsuga [Ohsuga 1989] in which the design process again is described as a series of stages, progressing from requirements through conceptual design and preliminary design to detail design, which is illustrated in Figure 2.3. For each stage, however, models of the design are developed through a process of analysis and evaluation leading to modification and refinement of the models. The early stages of the design process, involve evaluation of the design from a number of viewpoints to establish its fitness to the given requirements. The process in that stage is repeated until the design is at a point where it can be developed in

more depth, and then passed to other stages of the design process. In the preliminary design stage the design is refined, and evaluation and modification repeated at a greater level of detail. Finally, the detail design phase proceeds in a similar fashion to complete the definition of the design for manufacture. Ohsuga's model demonstrates that it is not necessary to carry out the design process strictly sequential, which gives one of the reason to use concurrent design process. Evaluation of the design can be from many different viewpoints at each stage, involving consideration both of overall concept and of engineering detail.

In practice the evaluation of designs often involves complex networks of analyses in which the output from one analysis forms the input for another - for example in the design of a car, the loads that are used in stress analysis of the parts of a suspension system come from analysis of the kinematic and dynamic behaviours of the suspension, and the output from this stress analysis is then used as the input to a

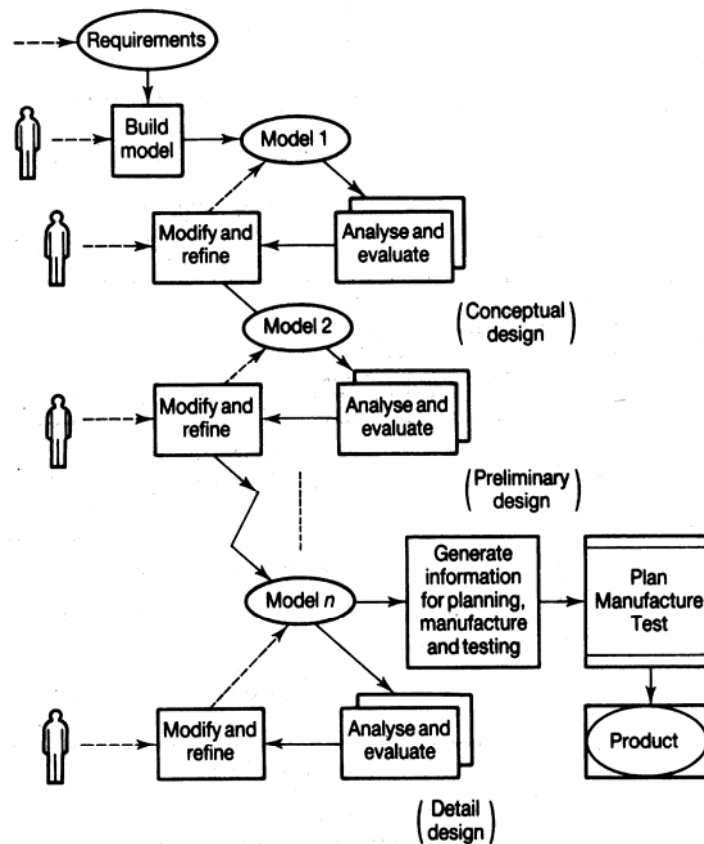


Figure 2.3: The design process proposed by Ohsuga [Ohsuga 1989].

subsequent fatigue analysis. Therefore, developing a design methodology in order to model the interaction of the different analyses in the different design stages, shown in Figure 2.3, become very important issue for which emerging computer and network facilities can be used to facilitate the design problem.

With the development in computing facilities as well as in network technology, interactions between analyses can be modelled in a network in which specialists from different engineering fields could contribute to the design process by enabling exchange of related data along with establishing concurrency of the evaluation of the processes. For that purpose, Concurrent Engineering can also be applied to the integration of design and analysis techniques, network techniques and data exchange between different stages of a design process. Therefore, there has been a growing interest in modelling the relationships between the activities in the design process, and the data flows in these activities, leading the development of various modelling techniques [Glenney 1985] [Chajzman 1988], which are explained in the next section.

2.2 Design and the Design Process

The engineering design process is related to the process of actions and decisions that are taken during design in order to arrive at a completed product design. One of the most widely accepted models of the design process is given by Pahl and Beitz [Pahl 1984], for which the main steps of the process are illustrated in Figure 2.4. As indicated in the figure, the overall design process is broken down into a number of stages; *i.e.* clarification of the task, conceptual design, embodiment design and detail design. Each of these stages can then be considered independently, the interactions between them being kept to a minimum. The major advantage of this approach is the simplification of the subsequent design process for the individual modules. Unfortunately, in practice the main steps cannot be clearly defined for most engineering design problems. Moreover, there is always feedback to previous stages and often iteration between stages [McMahon 1998]. Therefore, many different methodologies have been proposed to overcome the deficiencies of the Pahl and Beitz's model, some of which can be given as [Hubka 1984] [Taguchi 1986] [Ohsuga 1989].

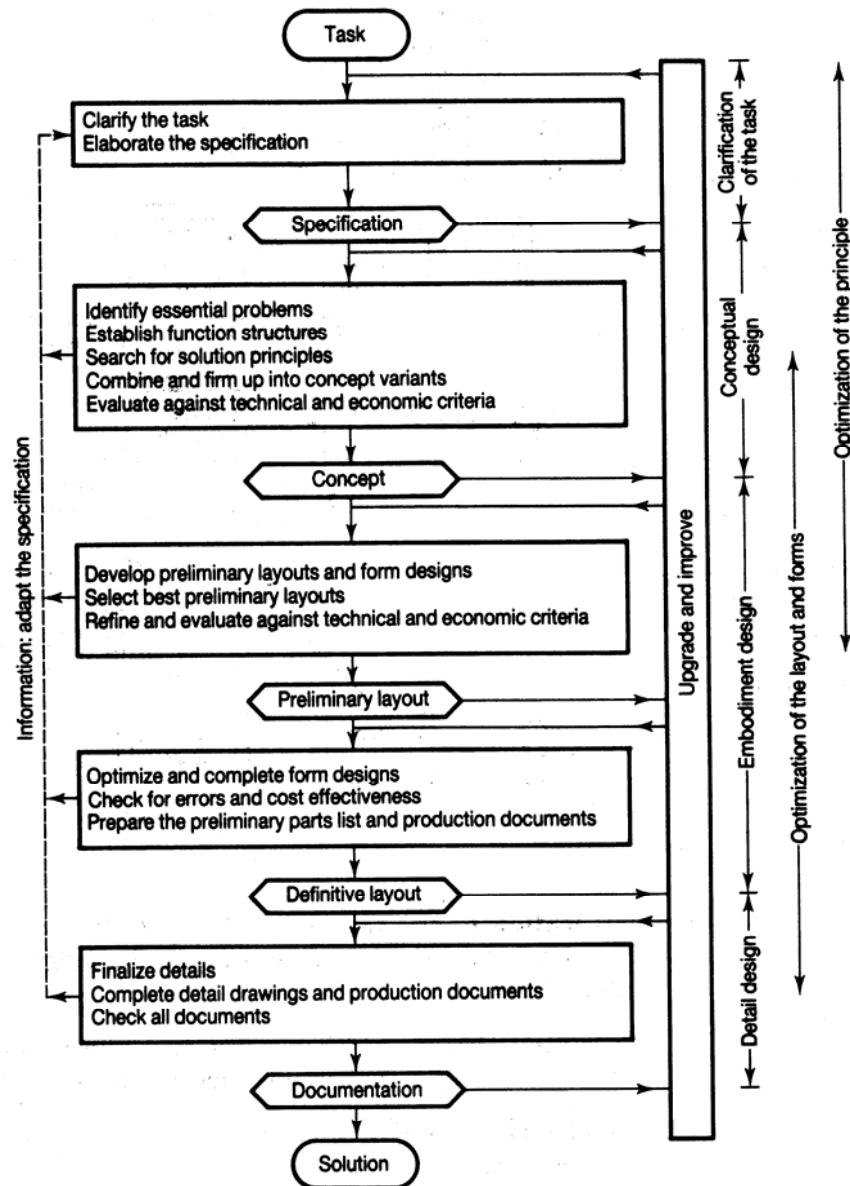


Figure 2.4: Main steps of Pahl and Beitz model for the design process, taken from [Pahl 1984].

2.3 Modelling the Design Process

Many activities, which some of them are illustrated in Figure 2.4, are involved in the development of a product design, defined as the design process. If the manufacturing side of the product design is left out from this concept, individual activities can be classified according to the stage in the process - for example specification, concept, preliminary layout, definitive layout as in Pahl and Beitz'

model, or according to the domain of the evaluation for analysis activities: for example mechanism analysis, structural analysis and so on. Even in one activity of the design process, many sub-activities on tasks may be present. Computational finite element analysis of a mechanical component, for instance, may include mesh model creation, static analysis, modal analysis, dynamic analysis, fatigue calculation, reliability analysis, optimisation and so on, each of which satisfies the needs of some different engineering task. In order to have a successful design, the collaboration between the specialists who carry out the activities will therefore be very important.

Computers are applied to almost all of the processes in the design of mechanical components and systems to improve traditional design processes by means of the advantages of computer-aided design such as handling very complex structural analysis by means of Finite Element Analysis (FEA), representing the geometry in CAD environment by using a boundary representation geometric model and so on. Therefore, more and more computational methods and software are being developed to meet the needs of different levels of design and analysis, resulting in different standards and adopting different theories. A number of software systems for the same kind of analyses can be obtained in the market although they may be different in operation and features as well as having their own input and output forms of data and interfaces.

The importance of data transfer in the design process has been realised by many designers and researchers since different types of data exist in the design process. Due to different types of software used in the design process, a great effort has been made to specify the standard for data exchange between the software in recent decades, which has led to the production of the *STandard for the Exchange of Product data* (STEP) [ISO 1993].

In order to manage the activities involved in the design of a complex product as well as enabling the transfer of the required data among the design activities, a high level modelling technique which can model the process well is very necessary to assist in understanding and ultimately automating the design process. Such a model will reflect all the relationships between the activities and the configuration of the integration system. Users can design the systems and tools used in the design process with the help of the modelling techniques, and make modifications to them after

discovering problems in operation of the system through analysing the models created.

An overview and evaluation of some modelling techniques, which include Data Flow Diagrams, the IDEF series of techniques and the EXPRESS language are given briefly in the following sections. Only one modelling techniques, the Petri net approach, is to be presented in detail since it forms the basis of the developed probabilistic design system, which is explained in Chapter 6.

2.3.1 Modelling Techniques

A large number of modelling techniques for various purposes such as information modelling, software engineering and various engineering design and analysis procedures have been presented in recent decades in various forms [Brookes 1994]. Some of the most widely used techniques are given in the present thesis along with a detail explanation of the use of Petri nets that are used to model a probabilistic design system in which uncertainty evaluation is handled in a distributed concurrent environment.

2.3.1.1 Data Flow Diagram:

A data flow diagram describes the functional relationships of the values calculated or computed by a system, including input values, output values, and internal data stores. A data flow diagram is a graph showing the flow of data values from their sources in objects through processes that transform them to their destination in other objects. The main elements of a data flow diagram, illustrated for a simple process of structural analysis, are given as follows [Yourdon 1989]:

1. Processes that transform data values, are represented by an ellipse or a circle containing a description of the transformation, usually its name.
2. Data flows that move data are represented by arrows between the producer and consumer of the data. They are also labelled with a description of the data, usually its name or type.
3. Actor objects that produce and consume data, are shown as rectangles in Figure 2.5.

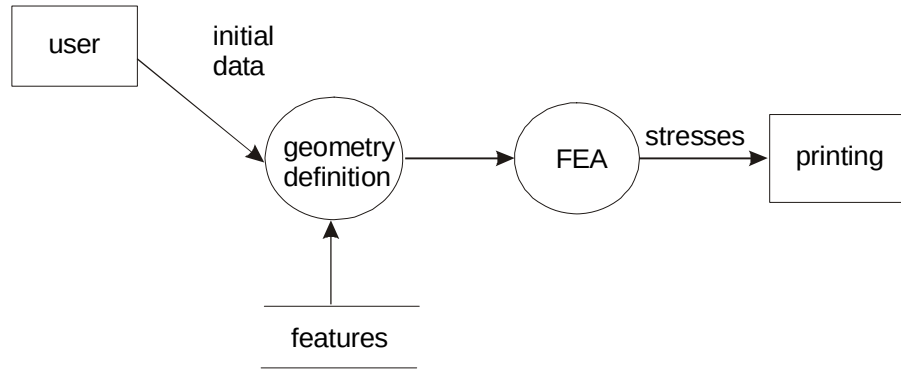


Figure 2.5:A Data Flow Diagram for a simple FEA analysis.

4. Data store objects that store data passively are represented by a pair of parallel lines containing the name of the store. Unlike an actor, a data store does not generate any operations on its own but response to requests to store and access data.

It is straightforward for a Data Flow Diagram to express the direction of a data flow, the source and the destination. However, a Data Flow Diagram does not show control information, such as the time at which processes are executed or a decision is made. The model shows a computation and the functional derivation of the data values in it without indicating how, when, or why the values are computed. Since a Data Flow Diagram is easy to understand and use, many applications are modelled by this technique [DeMarco 1979] [Austin 1996].

2.3.1.2 IDEF:

The IDEF methodology was developed for the US Air Force as a part of their Integrated Computer Aided Manufacturing project [AFSC 1981]. It was designed to assist with the analysis and documentation of the activities and processes found within engineering organisations. Although IDEF is a set of several methodologies, such as IDEF0, IDEF1x, IDEF2 to IDEF5 [Godwin 1989] [IDEF 2000], only IDEF0 is reviewed in this chapter since it is the most widely used technique for process modelling.

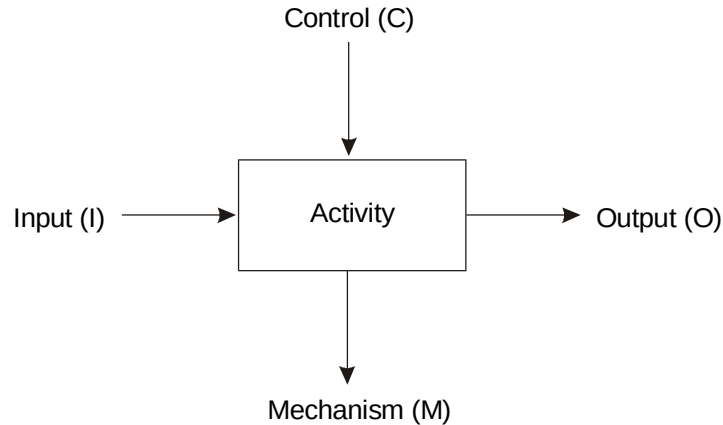


Figure 2.6: The activity box in IDEF0.

An IDEF0 model is a set of hierarchically linked diagrams with some associated supporting text. Each box as given in Figure 2.6 on a diagram represents a function: activity or process which is described by an active verb phrase contained within the box. Arrows represent the relationship between activities in terms of the information used, produced or required by activities. Arrows entering the left side of a box are inputs (I) to the activity, arrows entering the top of the box are controls (C) on the activity such as constraints in an optimisation problem and arrows leaving the right side of a box are outputs (O) as a result of the activity. On the other hand, a mechanism (M) is a person, system or device associated with carrying out the activity and is shown as an arrow entering the bottom of the box.

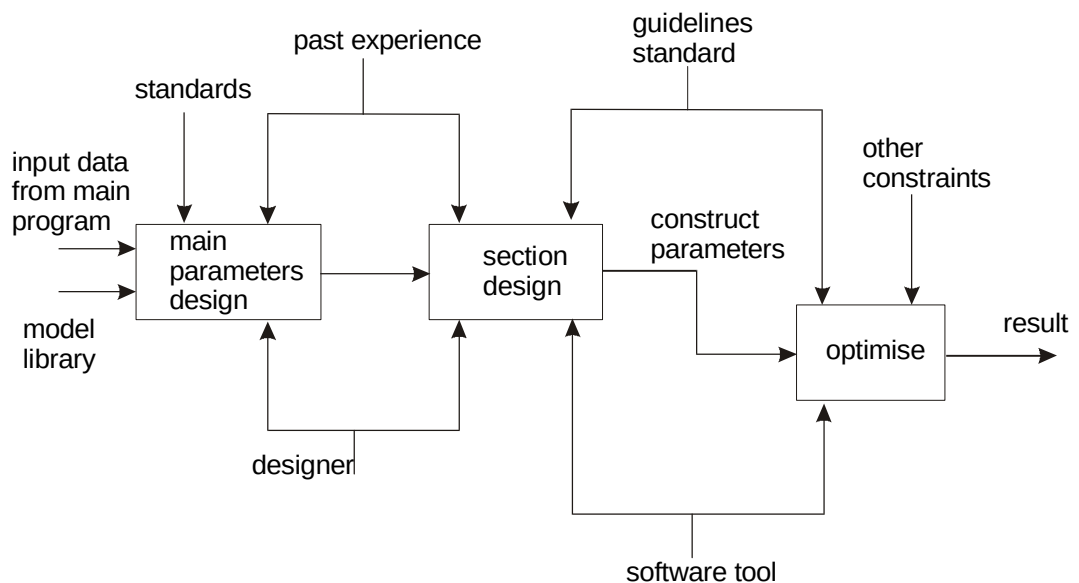


Figure 2.7: A structural optimisation problem modelled with IDEF0.

A simple example is given in Figure 2.7 in which the structural optimisation of a mechanical system design is modelled with an IDEF0 diagram. In the design of the main parameters the input data comes from the main system and the existing models. The main parameters can be designed by the designer under the constraints of standards and past experience in the stage. In the design of section design all of the cross sections of the structure will be specified by designer with the help of a software tool under past experience, guidelines and standards. Finally, the construction parameters will be optimised by an optimisation program under certain constraints.

Although IDEF0 is a more powerful modelling technique than data flow diagram, several deficiencies have been reported by many researchers [Malhotra 1992] [Brown 1989] [Godwin 1989], classified as: weak cohesion between the function, information, and dynamic models; difficulty in capturing the complete semantics of the modelled domain in the information model; and poor adaptation of the method to modelling dynamic situations (both the dynamic state of the process, and processes in which activities and their relationships change with time).

2.3.1.3 STEP and EXPRESS:

The STandard for the Exchange of Product model data (STEP) is an international standard for the computer interpretable representation of product data [ISO 1993]. The primary goal of the standard is to provide a neutral format for product data exchange throughout the life cycle of a product. STEP aims to create "standard bridges" between the various software packages used during the product's life cycle such as CAD, FEA and other engineering analysis packages. The purpose of these standard bridges is to provide a standard format for data transfer between the multitude of STEP compliant software [STEPTOOLS 2000].

EXPRESS is an information modelling language, describing the characteristics of information that may someday exist in an information base. It was initially developed to document formal information models without ambiguity for mechanical products. EXPRESS is one of the technologies that has developed as part of the STEP standard for product model data exchange [ISO 1993].

EXPRESS is both computer-interpretable and human-readable. It conforms to a formal syntax so that the EXPRESS model can be validated and processed by

computer software, for which many different software systems have been developed. It can also be presented to a human reader in a form that allows a data specification to be readily understood.

EXPRESS emerges as a solution to enable the various application areas to work together by providing a means to define data models which can enable vendors to incorporate neutral data exchange capabilities into their solutions. EXPRESS has been used successfully as a data software specification language for CAD packages; to define compiler data structures; and as a neutral data specification language for a variety of database packages.[ST-REPOSITORY 2000].

From the basic EXPRESS language, many advanced modelling techniques have been emerged such as EXPRESS-G [Booch 1994], using graphical symbols to represent modelled elements in diagrammatic form and EXPRESS-I which allows creating instances of EXPRESS components that have values instead of references to datatypes [Wilson 1996].

2.4 Petri Nets

Petri nets were originally developed by Carl Adam Petri in 1962 for his PhD and later described by Peterson [Peterson 1981]. Since then, the topic has been the subject of considerable research in a number of different disciplines, including engineering, manufacturing, business, chemistry and mathematics.

Most theoretical works about Petri nets are based on a formal definition of Petri net structures [Reisig 1985]. However, a graphical representation of a Petri net structure is much more useful for illustrating the concepts of Petri net theory. Therefore, an explanation of Petri nets is given for the graphical representation of Petri nets in the next section.

2.4.1 Basic Notations

A Petri net is a directed graph comprising two types of nodes called places and transitions. A place is represented by a circle and a transition by a bar, as illustrated in Figure 2.8. Directed arcs (arrows) connect the places and the transitions, with some arcs directed from the places to the transitions and other arcs directed from transition

to place. An arc directed from a place to a transition defines the place to be an input of the transition. Multiple inputs to a transition are indicated by multiple arcs from input places to the transition. An output place is indicated by an arc from the transition to the place. Again, multiple outputs are represented by multiple arcs.

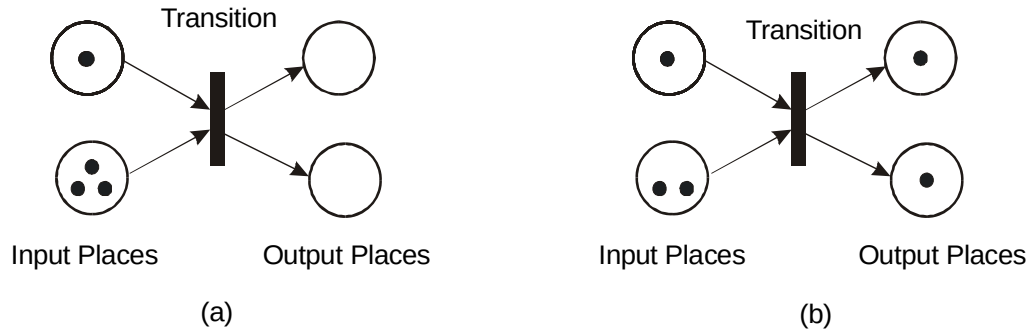


Figure 2.8: Basic components of a Petri net, before firing(a), after firing (b).

2.4.2 Marking

A marked Petri net means every place contains an integer (positive or zero) of **tokens** or **marks**, which are represented by small dots in places in Figure 2.8. The number of tokens contained in a place P_i will be called either $M(P_i)$ or m_i . For the example in Figure 2.8(a), which shows a net before firing, we have $m_1 = 1$, $m_2 = 3$. The net marking, M , is defined by the vector of these markings, i.e. $M = (m_1, m_2, \dots, m_n)$, in which n denotes the number of the places in the Petri net. The marking at a certain time defines the state of the Petri net, or the state of the system described by the Petri net. The evolution of the state thus corresponds to an evolution of the marking, an evolution which is caused by **firing** of transitions.

2.4.3 Firing of a Transition

The execution of a Petri net is controlled by the number and distribution of tokens in the Petri net. Tokens reside in the places and control the execution of the transitions of the net. A Petri net executes by firing transitions. A transition fires by removing tokens from its input places and creating new tokens which are distributed to its output places. A transition can only be fired if each of the input places of this transition contains at least one token. The transition is then said to be enabled. The transitions in Figure 2.8(a) are enabled because in each of places P_1 and P_2 contain at

least one token. A transition fires by removing at least one enabling token from its input places and then depositing into each of its output places one token for each arc from the transition to the place Figure 2.8(b). Multiple tokens are produced for multiple output arcs. Firing a transition will in general change the marking of the Petri net to a new marking. Notice that since only enabled transitions may fire, the number of tokens in each place always remains non-negative when a transition is fired. Firing a transition can never try to remove a token which is not there. If there are not enough tokens in any input place of a transition, the transition is not enabled and cannot fire.

2.4.4 The Process Controller of Petri Nets

For a design process for which the processes required for the analyses are well defined, Petri nets can be used to enable the automation of the execution of the analyses as well as the data transfer. This task is achieved by using a controller program which automatically manages the sequential and concurrent execution of the different processes used in the design analysis. This program is implemented as a process controller, and the dependencies and the state of the Petri nets are controller by means of a matrix notation which is explained in detail in Appendix A. Petri net initialisation is achieved by the controller reading a user-prepared input file which contains the relationship between the transitions and the places as well as including the initial marking of the places.

2.4.5 An Example of Petri Nets

To explain how a Petri net can integrate different analyses as well as enabling data transfer among them, an example is adopted from McMahon and Meng [McMahon 1996], in which an encastre beam subjected to a single tip load is taken as an optimisation problem. The Petri nets graph showing the relationship and dependency of each analysis is given in Figure 2.9, and the transitions and the places are explained in Table 2.1, for which the matrices used by the process controller to enable the tasks in the Petri nets are given in Appendix A.

The process of assessment comprises functions for the evaluation of the transitions given in Table 2.1. Multiple tokens are shown where a place has multiple links to

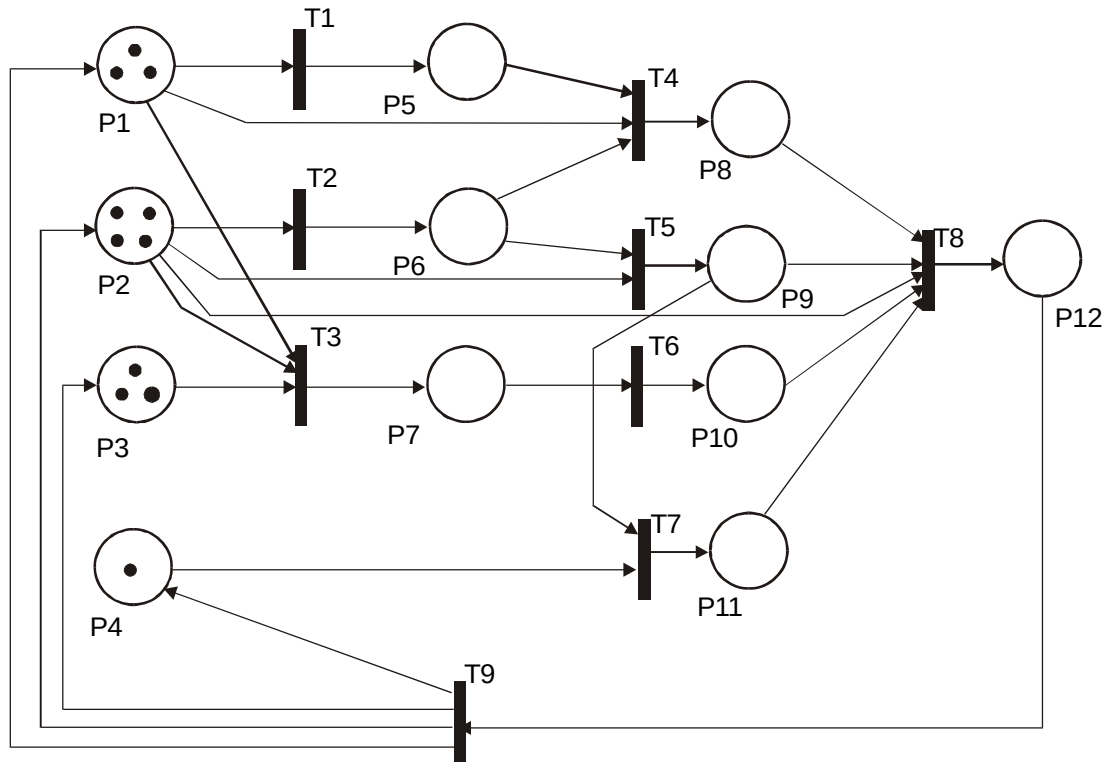


Figure 2.9: Petri nets for the optimisation of an Encastre Beam, adapted from McMahon and Meng [McMahon 1996].

transitions. The iteration required by the optimisation program is driven by firing T_9 for each iteration.

2.4.6 Comparison of Petri Net with Other Techniques

The modelling techniques given in this chapter are concerned with the modelling of a design system, and should be able to model the processes included in the system, the relationships, data exchange and sharing between them, the dynamic state of the process and other factors. These modelling techniques can be grouped into two classes according to their characteristics. Data flow diagrams, IDEF0 and Petri nets are mainly designed to describe functions and connections between them. So they can be considered to be function (process)- relationship modelling techniques. In contrast, EXPRESS places its emphasis on describing entities (objects) and the relationships between them. So they can be considered to be entity-relationship modelling techniques. Data flow diagrams can be used to model the processes and data flow.

Table 2.1: Explanation of the transitions and places in Figure 2.9.

Transitions	Task
T ₁	calculation of the inertia moment
T ₂	calculation of the moment
T ₃	calculation of the weight
T ₄	calculation of the stress
T ₅	calculation of the deflection
T ₆	calculation of the cost
T ₇	calculation of the constraint functions
T ₈	collecting the results
T ₉	assessment/iteration control

Places	Task
P ₁	beam dimensions
P ₂	problem parameters
P ₃	material parameters
P ₄	limits
P ₅	inertia moment
P ₆	moment
P ₇	weight
P ₈	stress
P ₉	deflection
P ₁₀	cost
P ₁₁	constraint functions
P ₁₂	results

However it is difficult for the technique to describe the dynamic behaviour and concurrent processes in a system. IDEF0 models not only process and data flow, but also describe the controls on a process and the mechanism of implementing the process. An IDEF0 model can be created in multiple levels of detail. So a system can be described in concise and hierarchical form. However, an IDEF0 diagram cannot easily be used to represent the dynamic behaviour of a system. EXPRESS is more suitable for representation of the entities that go to make up design information and models, such as geometric modelling entities.

Petri nets enable discrete event systems to be modelled. Petri nets allow reference to both process and data in a system, although additional representations (such as EXPRESS for example) are required for the detailed representation of both

process and data. They are especially useful for modelling the dynamic behaviour of a system through the firing of transition by moving the tokens. Petri nets make it possible to model and visualise behaviours comprising concurrence, synchronisation and resource sharing. Petri nets can also be used to describe the decomposition of a process into more detailed processes.

As pointed out by Ohsuga [Ohsuga 1989] in the earlier section of this chapter, the design process involves a series of stages in which models of the design are developed through a process of analysis and evaluation leading to modification and refinement of the model. The models that are proposed of the emerging design are models that describe the physical characteristics that the design will have, which are called the *design parameters*. The evaluations that are made are to evaluate how the design will perform and can therefore be called the *performance parameters*. Each of the performance parameters describes the performance of the design from a different viewpoint or perspective. The whole design must be satisfactory from all of the viewpoints required in the requirements specification, not just from a single viewpoint, and the task of the design team is to come up with a design that matches all of these viewpoints. Therefore multidisciplinary design optimisation in which different disciplines in engineering design contribute to the design of the product has gained interest in recent years, for which the basic structure is explained in the next section.

2.5 Design Integration via Optimisation

One of the main applications of the integration of multiple viewpoints in product design is the use of optimisation, which is a formal method that applies only to design improvement within the constraints of a particular design concept [Taylor 1992]. Although an optimisation problem can be broadly divided into two main groups, depending on whether the variables in the design problem are subjected to any limitations (*a constrained optimisation problem*), or no limits are imposed on the values of the variables (*an unconstrained optimisation problem*), most engineering optimisation problems includes constraints on the variables. Therefore, optimisation problems can be represented by the following three basic ingredients:

- An *objective function* which is to be minimised or maximised, depending on

the objective of the design problem. For instance, in designing an automobile suspension arm, the objective function could be to maximise the strength.

- A set of variables, called *design variables*, which affect the value of the objective function. In designing an automobile suspension arm, for example, the variables could be the dimensions of the automobile suspension arm and applied loads.
- A set of *constraints* that allow the unknowns to take on certain values but exclude others. For example, in designing an automobile suspension arm, it might be required to limit the weight of the suspension arm and to constrain its dimensions.

In a mathematical notation, above ingredients can be represented for a constrained optimisation problem as:

$$\begin{aligned}
 &\min F(\mathbf{X}) \\
 &\text{subject to} \\
 &\quad g_j(\mathbf{X}) \leq 0, \quad j=1, \dots, m \\
 &\quad \mathbf{x}_i^l \leq \mathbf{x}_i \leq \mathbf{x}_i^u
 \end{aligned} \tag{2.1}$$

where $\mathbf{X}$ is a vector of design variables and $F(\mathbf{X})$ is the objective function, and constraint functions are denoted as $g_j(\mathbf{X})$, in which m represents the number of the constraints of the design problem. The lower and upper values of the design variables $\mathbf{x}_i$ are denoted as $\mathbf{x}_i^l$ and $\mathbf{x}_i^u$, respectively.

In order to obtain the optimum value for the design variables, different methods have been developed, based on the type of the optimisation problem that can be broadly classified into two groups: transformation methods and search methods.

Transformation methods, such as the *Sequential Unconstrained Minimisation Techniques* (SUMT) [Fiacco 1990] and the multiplier methods, solve constrained optimisation problems by transforming them to an unconstrained optimisation problems by using transformation functions with a set of penalty parameters. The transformation function is then minimized by adjusting penalty parameters based on certain conditions.

Search methods, such as *Sequential Linear Programming* (SLP) [Vanderplaats 1984], *Sequential Quadratic Programming* (SQP) [Bazaraa 1993], and the gradient projection method [Gabriele 1977], directly solve the constrained optimization problem. The search method starts with an initial design and iteratively improves it. The design change is obtained using a two-phase approach: first the search direction is determined by solving an optimization subproblem with linearized constraints, and then the step length is determined by performing the line search. In this present thesis an optimization program for which detailed information is given in Section 2.6 is used as the optimisation tool for both structural reliability analysis and the case study given in Chapter 7.

In the last decade, stochastic optimisation methods called evolutionary optimisation methods have been applied to engineering optimisation problems [Keane 2000]. They can be classified into two groups according to the underlying conceptual approach used in the development of the methods. These groups are Simulated Annealing and Evolutionary Algorithm.

Simulated Annealing [Kirkpatrick 1983] is a generalisation of a local search algorithm based on an analogy with the physical annealing process which is a thermal process applied to a solid that leaves it in a low energy state. This is achieved through slow cooling to allow ample time for the redistribution of the particles of the sample. Basically, the analogy is formed as follows:

- A method for generating random changes in the state space is defined.
- The objective (cost) function of the problem is equivalent to energy in the mechanical process.
- A control parameter is defined ('temperature') which is lowered in accordance with an annealing schedule.
- An acceptance distribution is defined, based on a suitable physical law which determines, at a given 'temperature', whether a given generated state which does not lead to a lower cost is to be accepted.

The simulated annealing algorithm requires three things:

1. A starting point: a position within the search space.

2. A move generator: a mechanism for generating next states.
3. A cooling schedule: a mechanism to determine how often to reduce the temperature and by what amount.

Starting from an initial point, the algorithm takes a series of steps that are generated by the move generator, and the function is evaluated at each step. When minimising a function, any downhill step is accepted and the process repeats from this new point. However, the key to the capability of the algorithm to escape from local minima is the inclusion of the acceptance test; this allows acceptance of a solution of an uphill step resulting in the increase of the objective function. As the optimisation process proceeds, the length of the steps decline and the algorithm closes in on the global optimum [Van Laarhoven 1987].

The term “evolutionary algorithm” stands for a family of stochastic problem solvers based on principles that were adapted from Darwin’s idea of natural selection [Fogel 1998]. In this paradigm, achieving a solution to a given optimisation problem is seen as a survival task: possible solutions compete with each other for survival, and this competition is the driving force behind the progress that leads to an optimal solution. Among the evolutionary algorithms, genetic algorithms (GAs) have mainly been used for optimisation problems [Goldberg 1989]. Gas operate by generating a large set of possible solution to a given problem. Each of those solutions is evaluated and a *fitness level* identified for each solution. New solutions are generated by combining the characteristics of pairs of high fitness solutions, using the concept of chromosomes for each solution in the generation of new solutions (elements of the chromosome for each parent are used), and using mutation to introduce random variations into chromosomes. The parent solutions that were more *fit* are more likely to reproduce, while those that were less fit are less likely to do so. After a number of repetitions of the generate-test-regenerate cycle, the GA should result in a solution with the highest possible fitness. One of the advantages of using a genetic algorithm is that it treats the objective function as a “black box”, not requiring it to possess any special properties (such as continuity or differentiability). GA can also escape converging to a local minimum, which is the main problem of gradient-based optimisation methods. Therefore, GA is more suitable for the problem having many local minimum points. However, GA algorithm involves intensive computations, and

hence it is slower than gradient-based optimisation methods. Since most structural reliability problems do not have many local minimum points [Ditlevsen 1996], a gradient-based optimisation tool, CFSQP explained in Section 2.6, is used for the structural reliability problems in the thesis.

Most realistic optimisation problems, particularly those in design, require the simultaneous optimisation of more than one objective function with multiple constraints from different engineering disciplines. Designing the suspension arm of a vehicle [Tsai 1995], for example, needs to combine many specialist views from different disciplines, such as reliability specialist, structural analysis specialist, so on, as illustrated in Figure 2.10. In most design situations the specialists aim is just to satisfy the design requirements. Increasingly, however, the aim is to optimise the design in which case all these specialists' views can be combined using the term

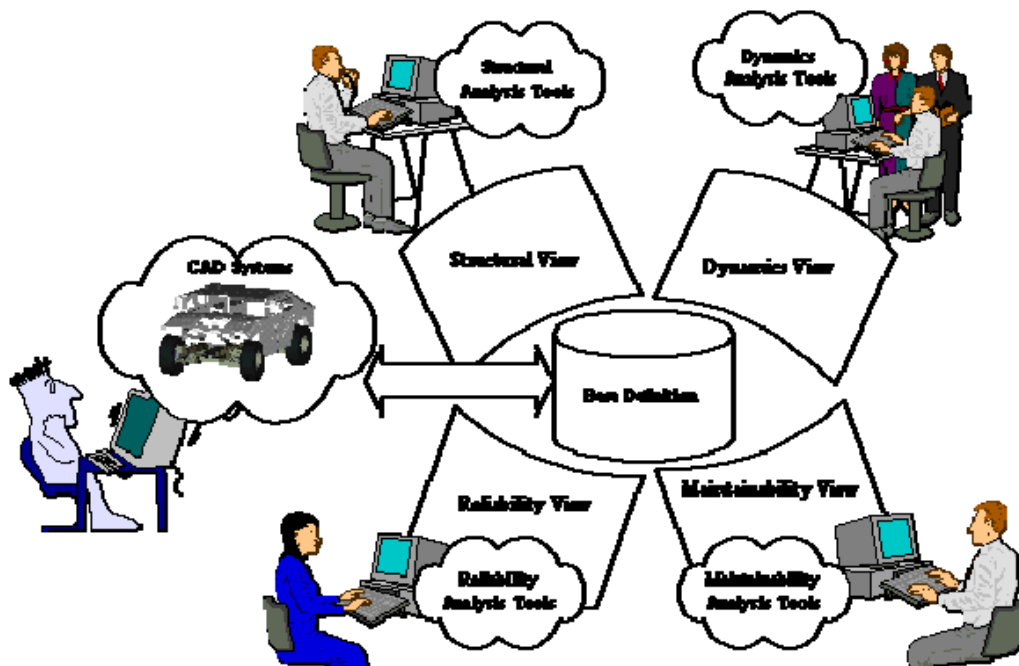


Figure 2.10: Integration of different analyses in a collaborative concurrent engineering environment, adapted from Tsai, *et al.* [Tsai 1995].

Multidisciplinary Design Optimisation (MDO) [Sobieszczanski-Sobieski 1984], in which all disciplines contribute either to define the goals of the design such as

designing the product with minimum cost, or to specify the constraints in the design problem. MDO has been heavily used for complex system designs such as aircraft design [Huang 1996] or high mobility multi-purpose wheeled vehicles [Wu 1992].

In the MDO, where design parameters are considered deterministic, then the methods used to obtain the optimum value are also deterministic. As stated in the first chapter of the present thesis, design problems often involve uncertainty that requires a more rigorous approach to the design problem. Therefore, the optimisation problem involving uncertainty can be rewritten as follows:

$$\begin{aligned} &\min F(\mathbf{X}) \\ &\text{subject to} \\ &P[g_j(\mathbf{X}) \geq 0] \leq p_j \quad j=1, \dots, m \end{aligned} \tag{2.2}$$

where $\mathbf{X}$ is the vector of random variables involved in the optimisation problem, p_i is the prescribed failure probability limit for the j th performance function and m is the total number of prescribed probabilistic constraints. By comparison with the deterministic optimisation problem given in Eq. (2.1), Reliability Based Structural Optimisation (RBSO) involves more evaluations for the constraints, depending on the method selected for the evaluation of the probabilistic constraints, some of these methods such as First Order Reliability Method (FORM), Second Order Reliability Method (SORM), and Monte Carlo simulation are given in detail in Chapter 3.

Much research has been carried out to develop a methodology to combine uncertainty evaluation for the probabilistic constraints with the optimisation problem [Rao 1980] [Nikolaidis 1988] [Enevoldsen 1994a] [Enevoldsen 1994b] [Chandu 1995] [Yu 1998]. Most of the methods for uncertainty evaluation include sub-optimisation problem to find some reliability data, as is explained in next chapter. Therefore, in this chapter only the overall system for uncertainty evaluation is explained in conjunction with the methods used for reliability analysis.

For the design of a complex system such as an aircraft or an automobile, the analyses required for the evaluation of both objective functions and constraints become very time consuming. Therefore, approximation techniques such as the response surface method have been increasingly used in order to reduce the

computational burden, as is explained in Chapter 4. As the design tasks also become more decentralised, communication requirements become more severe. Therefore, the integration of different disciplines in order to enable not only efficient computation for the evaluation of the different specialists' viewpoints, but also to supply the required data to the specialists in the environment becomes crucial in complex system design. In the present thesis, integration of different analyses is taken into account to represent the importance of developing a system in which the interaction and communication of different processes can be handled efficiently in terms of the time and cost, which is broadly explained in Chapter 6.

2.6 CFSQP: An Optimisation Tool

CFSQP is a set of C program functions, which is capable of tackling optimisation problems of almost all kinds, which can include multiple competing linear/nonlinear objective functions (minimax problems), linear/nonlinear inequality constraints and linear/nonlinear equality constraints. CFSQP has been developed at the Institute for System Research (ISR) at the University of Maryland [CFSQP 1998].

The algorithms in CFSQP are based on the concept of feasible sequential quadratic programming. For problems without nonlinear equality constraints, starting with a feasible point (provided by the user or generated automatically) these algorithms produce successive iterates that all satisfy the constraints. After feasibility has been reached, the objective function can be decreased either after each iteration with an Armijo-type [Panier 1993] arc search or after at most three iterations with a nonmonotone line search. The user has the option to choose one of the two searches. The merit function used in both searches is the objective function itself. Equality constraints are handled by way of an exact, differentiable penalty function that penalises negative values, while positive values are precluded by the same scheme used for inequality constraints. In addition, CFSQP includes a special scheme for efficiently handling problems with many more objectives or constraints than variables.

2.6.1 The Key Features of CFSQP

The features of CFSQP that also the basis for the choice of CFSQP in the present

research are summarised below.

- Efficiently handling of problems with many “sequentially related” objectives and/or constraints.
- Availability of the C language source code for the application, allowing it to easily be integrated into the research work.
- Ability to search for an initial feasible point satisfying all linear constraints and nonlinear inequality constraints.
- Ability to generate iterates satisfying all linear constraints and nonlinear inequality constraints (mandatory for many applications), starting from a feasible point.
- It is not necessary for the user to provide finite difference approximations to gradients.
- Global convergence and fast (two-step superlinear) local convergence.
- Availability of the original authors to discuss any theoretical and practical issues related to CFSQP.

CFSQP has been used widely in many engineering applications as an optimisation tool. Example applications include the optimisation of a laterally loaded welded rectangular cellular plate [Farkas 1994], a mesh reduction approach to parametric surface polygonization [Khan 1995] and high performance supersonic missile inlet design using automated optimisation [Zha 1996].

2.7 Distributed Computing Techniques

A distributed computing environment is one in which a process or program is running in many different address spaces or, simply, a single program is running on many different computers. The classic example of a distributed computing environment is a cluster of workstations connected to a mainframe. Before the establishment of a standard for distributed computing techniques, distributed computing environments were platform, hardware, and network specific which means a program running distributively on one vendor's system could not be expected to operate on that of another. Therefore the application of distributed computing in

engineering application followed the same way in which Unix network programming techniques, which are explained in Chapter 6, were mostly used. The Distributed Numerical Computing Environment, (DNC), which was developed at the University of Maryland [Voon 1991], for instance, was developed to tackle the computational burden for structural optimisation problems. Although DNC reduces the computational time of an optimisation problem, it has some drawbacks such as being platform, hardware and network dependent since it was developed for one platform. Therefore, it requires considerable knowledge and efforts on network computing to use DNC in another platform. That kind of drawbacks has led to the development of heterogeneous distributed computing environment such as Parallel Virtual Machine.

Parallel Virtual Machine (PVM) is a software environment for heterogeneous distributed computing, which was developed at Oak Ridge National Laboratory by Vaidy Sunderam and Al Geist [Geist 1995]. It allows a user to create and access a parallel computing system made from a collection of distributed processors, and treat the resulting system as a single virtual machine (hence the name, parallel virtual machine).

The hardware in a user's virtual machine may be single processor workstations, vector machines or parallel supercomputers or any combination of those. The individual elements may all be of a single type (homogeneous) or all different (heterogeneous) or any mixture, as long as all machines used are connected through one or more networks. These networks may be as small as a LAN connecting machines in the same room, as large as the Internet connecting machines across the world or any combination. This ability to bring together diverse resources under a central control allows the PVM user to divide a problem into subtasks and assign each one to be executed on the processor architecture that is best suited for that subtask.

The PVM has been applied to structural optimisation problems to distribute the tasks around the computer networks in order to reduce the computational time for the analyses. Some of the applications of the PVM involves the optimisation of the nuclear core reload design [Schütz 2000], and development of a multidisciplinary design optimisation test simulator [Hulme 1997].

The advances in object-oriented programming has led to a distributed computing

technique, CORBA by which a single program can run on many different machines on different platforms. CORBA is an open standard distributed computing environment defined by the Object Management Group (OMG) [Orfali 1997]. Unlike other technologies, such as Microsoft's COM+, DCE, or Java RMI [Vogel 1998], CORBA is not tied to a specific vendor, platform, or programming language. Programs written to the CORBA standard run inter-operable on all platforms, hardware and networks. Moreover, in contrast to non-standard environments, CORBA is extensible. CORBA technology is based on an Object Request Broker Architecture which interfaces objects in disparate address spaces and provides services for their management. Its central components are the Interface Definition Language (IDL) and the Object Request Broker (ORB). IDL is a language for defining the interfaces of CORBA objects. The ORB provides a standard set of services through which actions may be taken on distributed objects. The Interface Repository (IR) is a standard service availing objects and their interfaces.

The application of CORBA -based distributed computing techniques to engineering design has been emerging in recent years, especially in optimisation field. A CORBA-based distributed optimisation system, OptServ [OptServ 2000], for instance, has been developed to solve general nonlinear, constrained optimisation problems. More recently, a distributed design system based on CORBA has been proposed by Michelana *et al.* [Michelana 1999].

2.8 Conclusion

The main aim of this chapter was to give a definition to engineering design and the design process along with a review of the some of the most widely used modelling techniques of the design process. The necessity of the integration of analyses in a design process was also represented, for which design optimisation and multidisciplinary optimisation were reviewed.

In the first section, an overview was represented for the definition and representation of engineering design and in addition recent developments in engineering design evaluation were also given. The second section contains a definition for design and design process while the third section explores the methodologies used to model the design process such as Data Flow diagram, IDEF

and EXPRESS. Since a more specific modelling technique, Petri nets, is used to model the design process in the present thesis, it was represented in detail. In the fifth section, integration of different analyses via multidisciplinary structural optimisation is represented briefly, for which an application containing the optimisation of the connecting rod under uncertain influence of the input parameters is given in Chapter 7. The optimisation tool used in the work is explained in the sixth section, and in the final section, distributed computing techniques are summarised.

Chapter 3

Structural Reliability Analysis

This chapter is devoted to an explanation of the basic techniques of reliability-based analyses. In the first section the three basic methods which are used to represent uncertainties explained in the first chapter are given. The second section includes an overview of the structural reliability methods, in which a classification for the methods is also presented. The third section presents a method called the stress-strength interference model that forms the basis for the more developed methods given in the following section. The fourth section includes more detailed information and the formulation for the most widely used structural reliability methods, based on the second moment code formulation, such as the First-Order Reliability Method (FORM) and the Second-Order Reliability Method (SORM). In addition, some basic terms used in these methods are also explained in this section while the fifth section gives basic information on an alternative approach to structural reliability problems, which is Monte Carlo simulation, along with some more advanced simulation methods such as Importance Sampling. Finally, this chapter concludes with a comparison of the results of the FORM/SORM with that of MCS, for which an alternative approach to find the value of the design point is also introduced.

3.1 Representation of Uncertainty

Various approaches for representing uncertainty in engineering design have been proposed in the domains of applicability, some of the widely used are described in this section [Klir 1994]:

3.1.1 Interval Mathematics

The simplest method for modelling data uncertainty is to estimate the bounds on various models' outputs based on the bounds of the model inputs and parameters, represented by their upper and lower limits without probability information [Alefeld 1983]. If an uncertain parameter x_i is known to be between $x_i - \varepsilon_i$ and $x_i + \varepsilon_i$, the interval representation of x_i is given by $[x_i - \varepsilon_i, x_i + \varepsilon_i]$, in which $\mp \varepsilon_i$ shows the interval boundaries of the random variable x_i . Accordingly, the model outputs would also belong to another interval due to the interval representation of the input parameters. The rules for interval mathematics are given by [Moore 1966] as follows:

Let

X_l represents the lower boundary of uncertain parameter X

X_u represents the upper boundary of uncertain parameter X

And for three uncertain parameters

$[X_l, X_u]$ represents the interval $X = \{x : x \geq X_l, x \leq X_u\}$

$[Y_l, Y_u]$ represents the interval $Y = \{y : y \geq Y_l, y \leq Y_u\}$

$[Z_l, Z_u]$ represents the interval $Z = \{z : z \geq Z_l, z \leq Z_u\}$

Basic arithmetic procedures are given as

$$X + Y = [X_l + Y_l, X_u + Y_u]$$

$$X - Y = [X_l - Y_l, X_u - Y_u]$$

$$X * Y = [\min(X_l * Y_l, X_u * Y_u, X_l * Y_u, X_u * Y_l), \max(X_l * Y_l, X_u * Y_u, X_l * Y_u, X_u * Y_l)]$$

$$X / Y = [[X_l, X_u] * [1/Y_l, 1/Y_u]] \text{ if } 0 \notin [Y_l, Y_u]$$

Although the above arithmetical procedures are same as in ordinary arithmetic, the distributive law is not always the same, which can be shown as:

$$X * (Y + Z) \neq (X * Y) + (X * Z)$$

but the following expression is obtained:

$$X * (Y + Z) \subseteq (X * Y) + (X * Z)$$

which is known as subdistributivity [Moore 1966]. In order to avoid subdistributivity, the vertex method can be used, for which more detailed information can be found in [Choobineh 1992].

As interval mathematics is a very simple representation of the uncertain parameters, it can address problems of uncertainty analysis that cannot be studied without having probability distribution of the uncertain parameters, such as in probability theory. However, this method does not represent adequate information on the nature of the uncertain parameters and clearly leads to the ignoring of information such as, mean values, variance or probability distribution when these are known.

3.1.2 Fuzzy Theory

Fuzzy theory has been developed to facilitate uncertainty analysis of systems, based on the concept of fuzzy sets which is first introduced by Zadeh [Zadeh 1965]. In ordinary set theory, an element is either a member or not a member of a set, which is described in a function called membership function $\mu_A(x)$ as:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

in which 0 represents non-membership and 1 represents definite membership. In the fuzzy set, whether or not an element belongs to a set may be expressed not just by 1 or 0, but by any continuous value between 0 and 1, indicating different degrees of the membership of the element, as is given below:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \\ p : 0 < p < 1 & \text{if } x \text{ partially belongs to } A \end{cases}$$

where p gets value like 0.3 for a weak membership while 0.9 shows a strong membership.

Fuzzy sets can be represented numerically by making use of α level cuts. An α level cut is defined as the real interval where the membership function is larger than a given value α and is written mathematically for a generic fuzzy set A as follows [Klir 1995]:

$${}^\alpha A = \{x \mid \mu_A(x) \geq \alpha\}$$

which can also be shown in Figure 3.1 where it is assumed that A has a triangular and symmetric membership function, and lower and upper bounds of the α level cut are ${}^\alpha a_1$ and ${}^\alpha a_2$.

A normal fuzzy set is one that has at least one element whose membership function is equal to 1, while a convex fuzzy set is one for which for all possible α cuts are convex. Thus, the fuzzy set A shown in Figure 3.1 is a fuzzy number, and it can be represented by only three variables: x_L , x_C and x_U [Ross 1994].

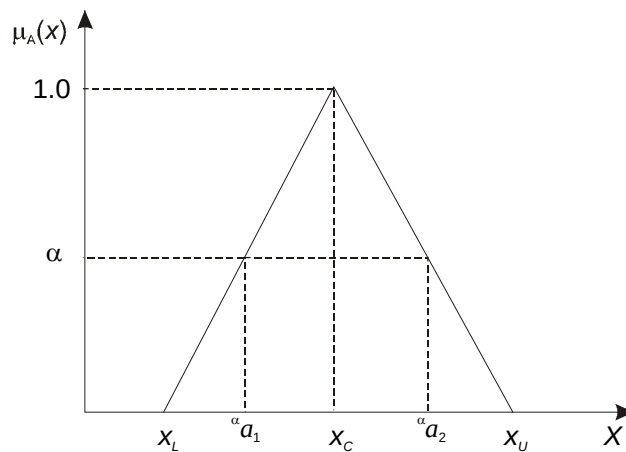


Figure 3.1: Triangular and symmetric membership function with α level cut.

Standard arithmetic and algebraic operations can be extended to fuzzy arithmetic and fuzzy algebraic functions by means of Zadeh's extension principle [Zadeh 1975]. The extension principle defines functions of fuzzy sets and can be used to define the degree of membership of one fuzzy subset in another through a known function. Let fuzzy sets $A_1, \dots, A_n$ be defined on the universe $X_1, \dots, X_n$, with membership function $\mu_1, \dots, \mu_n$ and let f be a function

$$f : X_1 \times X_2 \times \dots \times X_n \rightarrow X$$

The extension principle can be used to induce a fuzzy set B over X from $A_1, \dots, A_n$ through the function f , where $B = f(A_1, \dots, A_n)$. The membership function of B is given by

$$\mu_B(y) = \max_{y=f(x_1, \dots, x_n)} \{ \min[\mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n)] \}$$

The above equation is generally called Zadeh's extension principle [Ross 1994], and is given for a discrete-valued function, f . For the function which is a continuous-valued expression, the max operator is replaced by the sup (supremum) operator.

For the calculation of the membership function of B , several approaches have been proposed. One approach is to formulate the problem as a non-linear programming problem, which was proposed by Dong and Shah [Dong 1987] as:

$$\begin{aligned} &\text{maximise } \mu_B(y) \\ &\text{subject to} \\ &\quad \mu_B(y) \leq \mu_{A_i}(x_i) \text{ for } i = 1, \dots, n \\ &\quad f(A_1, \dots, A_n) = B \end{aligned}$$

Other approaches such as the fuzzy weighted average technique (FWA), which is based on interval analysis, and vertex method, which was developed to simplify the manipulations of the extension principle for continuous-valued functions can be found in [Dong 1987].

In fuzzy theory, assessment of uncertainties are often characterised by some phrases, such as *probable*, *likely*, *unlikely* and *approximately* 10^{-2} , for which a possibility distribution can be used to define these terms, showing that one value may be *more possible* than another. A formal definition of the possibility distribution is

given by Zadeh [Zadeh 1978] in which A is assigned as fuzzy set over a universe of discourse with a membership function $\mu_A(x)$. The proposition “ X is A ” associating the variable X with the concept represented by A , induces a possibility distribution Π_X on X which restricts the values which X may take, where Π_X is defined by

$$\Pi_X(x) = \mu_A(x) \quad \text{for all } x \in A$$

Several research studies have been carried out using fuzzy theory to represent the uncertainty in a structural reliability analysis. Ayyub and Lai [Ayyub 1992] studied structural reliability assessment by using fuzzy set theory to study the uncertainty associated with ambiguity. Venter [Venter 1998] applied fuzzy theory to model a composite laminated plate with internally terminated plies under uncertainty. Further literature on the application of fuzzy set can be obtained from [Ross 1994] [Klir 1995]. This literature suggests that fuzzy techniques are principally concerned with linguistic uncertainty, rather than uncertainty in which probability distribution function are explicitly known.

3.1.3 Probability Theory

In probability theory, uncertainties of a structural reliability analysis are defined by the probabilities associated with events which correspond to any of the possible states a system can assume [Stark 1986]. Tossing a coin, or experimental measurement of a material strength are some examples of the concept in probability theory. Thus, the probability of an event can be interpreted in terms of the frequency of occurrence of that event. For a large number of samples, for example, the probability of heads turning up is computed as the ratio of the number of times the coin turned head to the total number of times the coin is tossed.

In probability theory, uncertain parameters are represented by a random variable, which may be described by a single event probability value (for a logical random variable), a discrete set of probabilities (for a discrete variable) or a *Probability Density Function* (PDF) (for a continuous random variable). In most structural reliability problems, input variables are described as random variables and represented by a PDF since they are continuous random variables such as loads, dimensions or material parameters. Therefore, the first requirement of probability

theory is to assign a PDF to a random variable. There are many methods in the literature to obtain the PDF from a sample values of the random variable, some of which are parameter estimation [Ang 1975], maximum entropy [Cox 1961] and using ranks [Siddal 1983].

Although probability theory and fuzzy theory have been used to represent uncertainties in structural reliability analysis, they describe different aspects of uncertainty. The probability theory deals with randomness of the uncertain parameters while fuzzy theory deals with vagueness or fuzziness of the uncertain parameters expressed linguistically. Several studies have compared fuzzy set and probabilistic methods in analysis of the safety of systems under uncertainty [Wood 1994]. More recently Chen *et al.* [Chen 1999] compared two methods in terms of design rather than analysis for which two methods were used to obtain the design with a given amount of resources and a given amount of data about uncertainty. They concluded that the probabilistic approach is superior to fuzzy theory when full statistical information is available because of a more reasonable evaluation of multiple components as well as multiple failure modes. On the other hand, fuzzy theory appears to be more suitable for qualitative reasoning with limited information [Chen 1999].

Conventional methods based on probability theory for the evaluation of uncertainty, such as First Order Second Moment methods or Monte Carlo simulation are generally computationally expensive and time consuming, as are described in the following sections.

3.2 Overview of Structural Reliability Methods

A general review of structural reliability methods is given in the following paragraphs, based on the computational means used in the methods, which are broadly classified into two groups:

- Analytical Methods: the First-Order Reliability Method (FORM) and the Second-Order Reliability Method (SORM)
- Numerical Methods: Crude Monte Carlo simulation and Importance Sampling

The FORM and SORM methods are developed by using an analytical approach

explained in the later section of this chapter. However, simulation methods are

simpler to apply to structural reliability problems than the FOSM methods, and more accurate results can be obtained, but with a performance penalty, as is outlined in the Section 3.5.

3.2.1 General Review of First-Order Second Moment Methods

The necessity to incorporate uncertainty in engineering design analyses has been recognised in the last five decades, and many methods have been also suggested for this purpose since then. One of the first studies on structural reliability analysis was published by Freudenthal [Freudenthal 1956] in which the *design point* concept, which is the one of the most important concepts in structural reliability analysis, was first introduced by the author as a “checking point that has the critical condition which has the highest probability of failure”. In a later study, Freudenthal *et al.* [Freudenthal 1966] proposed that the probability of failure is a function of the shortest distance, assigned as β in the literature, between the limit state and the origin of the transformed space which is explained in Section 3.4. This definition was later named the *safety index* and is the most important term of structural reliability analysis, since the POF can easily be obtained for random variables distributed normally, for which a detailed explanation is given later in this chapter. However, this approach only dealt with two Gaussian random variables, which are the resistance strength, R , and the corresponding load, S , so the limit state, defining the safe or failure state of the problem, is a function of R and S , and the response of the problem was assumed to be linear. Therefore, there is no need to linearize the limit state to find the safety index. This approach only works for linear limit state functions like the stress-strength model given in Section 3.3. Obviously, most structural engineering problems have more than two random variables, and they are too complicated to be represented by a simple approach like the stress-strength model proposed by Freudenthal [Freudenthal 1956]. For a more general approach, Cornell [Cornell 1969] suggested a method in which a general limit state function is linearised by means of a Taylor series expansion at the mean value of random variables, which is known as the *Mean Value First-Order Second-Moment Method* (MVFSOM). Thus, the safety index was expressed as the

ratio of the mean values and the standard deviation of the performance function. Although this approach simplifies the original complex probability problem into a simple problem represented in linear terms, linearising highly non-linear limit state functions can result in a poor approximation, in which the results can simply ignore the nonlinearity of the response of the problem. The point on the limit state function determined by the mean values of random variables might be far from the maximum likelihood point; i.e. design point. So, the POF result based on this approach only reflects the POF of the limit state function at the mean value of random variables. In order to overcome the deficiencies of the MVFSOM, Hasofer and Lind [Hasofer 1974] proposed to linearise the general limit state function at the design point by means of Taylor series. This method is called *First Order Reliability Method* or FORM for short, and includes an approach to find the design point on the limit state function. In the FORM, all random variables are transformed into reduced space (transformed space, *U*-space) where they are standardised normal variables, and the minimum distance, which shows the reliability index, is obtained from this reduced system. Since the most important part of the FORM is to find the reliability index, several different methods have been proposed in the literature [Rackwitz 1978] [Ditlevsen 1981] [Schinozuka 1983]. One of the algorithms used to find the reliability index was suggested by Rackwitz and Fiessler [Rackwitz 1978] in which the initial design point is assumed to be at the mean values of the random variables, and an iterative procedure is carried out until the convergence of the value of β is reached. Ditlevsen [Ditlevsen 1981] proposed a different iterative approach in which the principle of normal tail approximation is used.

The computation of the reliability index can also be formalised as an optimisation problem for which several different optimisation algorithms have been proposed to get the minimum distance in the reduced space. Schinozuka [Schinozuka 1983] used the method of Lagrange multipliers to obtain the reliability index. Since most design engineers in structural engineering use a finite element code, Maymon [Maymon 1994] suggested a combination of one of the commercially available finite element codes such as ADINA, ANSYS or similar with probabilistic analysis. In his method, the *Modified Joint Probability Density Function* (MJPDF) is formulated, and the design point is obtained by finding a maximum value to this function in the

normal space, for which a built-in optimisation routine such as is incorporated in most commercial finite element packages, is used as a tool for numerical simulation. However, the limit state function must be defined explicitly, which is very difficult for most structural reliability problems. Borri and Speranzini [Borri 1997] proposed a more general method that can be applied to most structural reliability problems. In their study, to overcome the problem that transformation of random variables to normal space may result in negative values for variables that must have positive values at all times (such as material properties), a transformation is proposed that enables the reduced variables to have always positive values, in which constant values are added to random variables in a way that transformed values of the random variable always assume positive values.

A more recent study to develop a method to find the design point was proposed by Wang and Grandhi [Wang 1996] in which an adaptive safety index algorithm is developed. In this method, the limit state in U-space is approximated by an adaptive nonlinear approximation in which new variables are defined by means of the transformation proposed by Wang and Grandhi [Wang 1996]. The transformation of the random variables is based on the following expression:

$$y_i = \left(u_i + \frac{\bar{x}_i}{\sigma_i} s \right)^r \quad i = 1, 2, \dots, n \quad (3.1)$$

in which y_i is the new transformed variable, u_i shows the standard normal variable, $\bar{x}_i$ and σ_i are the mean value and the standard deviation of the random variable, respectively. The nonlinearity index is denoted as r which controls the functional nonlinearity, and the coefficient of co-ordinates shift is given as s which preserves the positiveness of normal co-ordinates. By this method, the accuracy of the approximation can be improved and in addition nonlinearities of the original performance function can be captured.

3.2.2 Review for the SORM

In order to reduce the errors caused by the linearizing of the performance function, a method called the Second Order Reliability Method, SORM for short, was suggested by Fiessler *et al.* [Fiessler 1979], in which second-order Taylor series of the

performance function as well as curvature-fitted second-order surfaces were used to approximate the performance function around the design point. A different approach for the SORM was proposed by Breitung [Breitung 1983], in which the problem of deriving approximations for multinormal integrals by using the results of asymptotic analysis was investigated. In his method, first order and second order Taylor expansion of the performance function around the design point were used to create the linear and quadratic approximations to the performance function. Hence, the comparison of those approximations with the exact probability integrals were carried out by means of their asymptotic behaviour. A more accurate three-term formulation of Breitung's method was developed by Tvedt [Tvedt 1988]. In his later studies [Tvedt 1988] [Tvedt 1990], an improved method was proposed in which the exact result for the probability expression is obtained from the full second-order Taylor expansion of the performance function at the design point by the inversion of the characteristic function. A simple method for the approximation to the failure surface by a quadratic equation was proposed by Der Kiureghian *et al.* [Der Kiureghian 1987]. In this study, a paraboloid is fitted to the failure surface at discrete points around the design point.

One of the recent studies on the SORM was carried out by Zhao and Ono [Zhao 1997] who introduced an empirical second order reliability index by which the SORM can be easily applied without rotational transformation and eigenvalue analysis of Hessian matrixes. In their latest study, they proposed [Zhao 1999a] an alternative point-fitting method for the second-order reliability approximation, in which the performance function is directly point-fitted using the response surface method proposed by Bucher and Bourgand [Bucher 1990].

Although the SORM improves the results obtained from the FORM, it can suffer the same deficiencies of the FORM in terms of finding the design point. Furthermore, there is no readily available algorithm to apply the SORM to the reliability problems. Instead of using these analytical methods, there is another alternative approach to obtain reliability data, such as the design point or the POF, based on simulation of the reliability problems. One of the most widely used simulation methods is the Monte Carlo method outlined in the following section.

3.2.3 Review of the Monte Carlo Methods

Monte Carlo simulation was first mentioned in the paper published by Metropolis and Ulam [Metropolis 1949] in a project which was one of the most important parts of the work to design thermonuclear weapons (the H-bomb). They suggested that random sampling could be used to simulate the flight paths of neutrons. The usefulness was commonly recognised in solving integrals and differential equations that led to a great deal of research on simulation [Meyer 1956]. Although the earlier applications of Monte Carlo simulation were carried out mostly in the field of physics and mathematics, one of the first applications of Monte Carlo simulation to structural reliability problems was proposed by Fiessler *et al.* [Fiessler 1979] in which the exact POF of a quadratic limit state function was calculated by means of Monte Carlo simulation. As the main effort is to reduce the variance of the POF that indicates the accuracy of the simulation, several methods have been suggested in the literature. These methods are generally called variance reduction techniques, some of them can be named as the importance sampling method, the stratified sampling method and the Latin hypercube sampling method [Rubinstein 1981] [Sundararajan 1994]. For more specific applications of Monte Carlo simulation to structural reliability analysis, effort has been spent on the development of methods that can require a smaller number of samples without an accuracy penalty. Therefore, Harbitz [Harbitz 1986] suggested a sampling method in which the sampling domain is restricted to values outside an area called the β -sphere whose radius is equal to the reliability index. Thus, a remarkable increase in efficiency was claimed to be obtained with regard to the conventional Monte Carlo methods. A further study on this method was proposed by Sweeting and Finn [Sweeting 1992], in which the approximation to the failure state involves two different approaches according to the type of the failure state. A simple hyperplane tangential at the design point is used to approximate the failure surface if the surface is relatively smooth, whereas a hyperparabolic approximation to the surface is suggested with the SORM method in the case that the shape of the failure surface consists of a significant curvature.

Of the most widely used variance reduction techniques is the importance sampling method first studied by Schinozuka [Schinozuka 1983]. In this method, the basic variables are sampled from a different distribution that gives more sampling from the

vicinity of the design points, forming the region of interest or importance region. Thus, more samples can be taken from the region contributing most to the calculation of the POF. Although the importance sampling method reduces the variance of sampling, the main problem in this method still remains, which is the selection of an appropriate distribution function for the importance sampling function. Harbitz [Harbitz 1983] suggested having the same type of distribution function for the importance sampling distribution function as for the random variables, but shifted to the design point. As Melchers [Melchers 1990] pointed out, once the design point is found there is a little justification for using a Monte Carlo method to compute the POF. Therefore, he proposed an algorithm called search-based importance sampling in which the points of maximum likelihood can be found through search and iteration coupled with efficient importance sampling. A similar method for updating the sampling function was suggested by Bucher [Bucher 1988], based on the notion that the knowledge about the failure domain increases as the simulation progresses. Hence, after starting the simulation by using crude Monte Carlo simulation, the sampling density is updated for the later stages. In order to make a decision among the existing Importance Sampling methods, Engelund and Rackwitz [Engelund 1993] carried out a benchmark study in which the methods were compared according to the criteria the authors proposed, which are robustness (against multiple critical points and noisy failure boundaries), capabilities to handle equalities, unions and interactions, efficiency and accuracy (convergence properties). Instead of reaching a decision for the best Monte Carlo method, they summarised the advantages and disadvantages of the methods based on these criteria, and concluded that intelligent use of either method should be made and that no prior preference for a particular method is justified.

3.3 Structural Reliability Methods: Detailed Description

In this section, further details will be given for the structural reliability methods introduced in Section 3.3, and in particular mathematical formulation will be described in detail. All the structural reliability methods given in the literature can be broadly classified into two groups, based on the computation methods used:

- Numerical Integration

- Analytical Methods

Since numerical integration is only efficient for problems where a small number of variables are involved, it is not efficient to use it to resolve the multidimensional integral representing the probability of failure given in Eq.(3.6). In this chapter, the analytical stress-strength interference method is given first since it is the basis of the most widely used analytical methods such as the FORM and the SORM for which more detailed information are given later in this chapter.

3.3.1 Stress-Strength Interference Method

Of the oldest methods of structural reliability analysis, the stress-strength method has been widely used in many engineering applications because it is simple and economical to use. In addition, it also plays a very important role in explaining more advanced structural reliability methods such as the FORM and the SORM. Thus, this section begins with a brief description and derivation of the stress-strength interference equation.

The name of the stress-strength interference method implies that structural reliability is computed from the stress distribution acting on the component and its strength distribution. However, this concept can be broadened if the term *stress* is

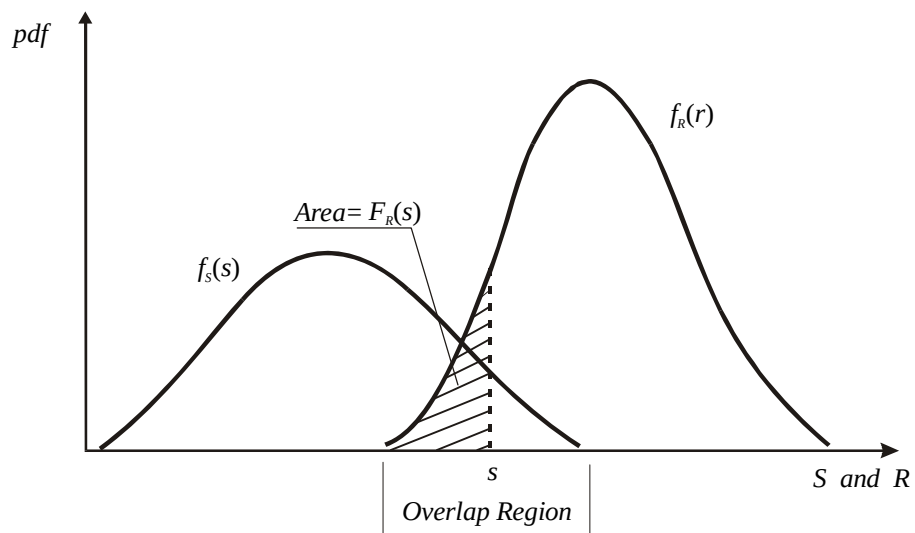


Figure 3.2: Illustration of Probability of failure for Stress-Strength.

exchanged with any applied load, such as force, moment, deflection, and so on. Similarly, the term *strength* can be interpreted as the capacity of the component to withstand the applied load, for which some examples are yield stress, ultimate stress, allowable deflection, and so on. The stress-strength interference equation is derived by describing the failure criterion as

$$g(X_1, X_2, \dots, X_n) < 0 \quad (3.2)$$

in which X_i is the basic random variables that can be load or stress, and $g(X)$ is called the *performance function*. A special value of $g(X)$ is of importance to describe the very significant term *limit state*, given as a special value of the performance function by

$$g(X_1, X_2, \dots, X_n) = 0 \quad (3.3)$$

The limit state can be described as “a state of the structure including its loads at which the structure is just on the point of not satisfying the requirement” [Ditlevsen 1996]. A more explicit representation of the limit state of the stress-strength interference model for failure cases can be presented as follows:

$$g(X_1, X_2, \dots, X_n) = [R(X_1, X_2, \dots, X_n) - S(X_1, X_2, \dots, X_n)] < 0 \quad (3.4)$$

where R and S are the vectors of the strength parameters and stress parameters, respectively. The probability of failure of Eq. (3.2) can be stated as

$$P_f = P[g(X_1, X_2, \dots, X_n) < 0] \quad (3.5)$$

A more general representation of the failure probability is

$$P_f = \int \dots \int_{g(X) \leq 0} f_X(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n \quad (3.6)$$

In order to describe Eq. (3.6), R and S can be used as follows:

According to the definition given in Eq. (3.2), the probability of failure can be given as:

$$P_f = P(R < S) = \int_0^\infty F_R(s) f_S(s) ds \quad (3.7)$$

If R and S are independent variables, the formula will be as follows:

$$P(R < S | S = s) = P(R < S) \quad (3.8)$$

Assume the probability distributions of R and S are available, and they are normally distributed, that is, $F_R(r)$ and $f_R(r)$ and $F_S(s)$ and $f_S(s)$ that are cumulative distribution function and probability density functions respectively are known and the probabilities may then be formulated for continuous random variables as:

$$P_f = \int_0^{\infty} F_R(s) f_S(s) ds \quad (3.9)$$

which is the convolution with respect to s and may be explained with reference to Figure 3.2 as follows: If $S = s$, the conditional probability of failure would be $F_R(s)$, but since $S = s$ is associated with probability $f_S(s)$, integration over all of S yields Eq. (3.9). The overlapping of the curves $f_R(r)$ and $f_S(s)$ represents a qualitative measure of the failure probability P_f . It indicates that a material with a lower strength value will encounter a high applied stress and this situation might cause failure. In deterministic analysis, in order to avoid the failure, the distance between the mean values of the parameters is increased by safety factor that is the ratio of the mean strength and the mean applied stress. Similarly, reliability analysis makes use of the safety margin concept which is used to determine safe mode and failure mode of a structure by introducing a new random variable, M derived by means of normally distributed random variables R and S

$$M = R - S \quad (3.10)$$

where M becomes a normally distributed random variable since the result of adding or subtracting two normally distributed is also normal random variable. The safe state and failure state of the limit state are described as

$$M \geq 0 \quad \text{safe state}$$

$$M < 0 \quad \text{failure state}$$

The calculation of probability of failure of M can be given as in Eq.(3.11)

$$P_f = \int_{-\infty}^0 f_M(m) dm \quad (3.11)$$

in which $f_M(m)$ is a density function of M with a normal distribution given as

$$f_M(m) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{m-\mu_M}{\sigma_M}\right)^2} \quad (3.12)$$

The integral given in Eq.(3.11) cannot be calculated by means of any analytical methods, therefore, the normal density function is transformed to a standard normal distribution, which is normally distributed with mean value equal to 0 and standard deviation equal to 1, by introducing the transformation as below:

$$Y = \frac{X - \mu_X}{\sigma_X} \quad (3.13)$$

For the boundaries of the integration given in Eq. (3.11), its equivalent value in the reduced space is given as

$$T = \frac{0 - \mu_M}{\sigma_M} = -\frac{\mu_M}{\sigma_M} \quad (3.14)$$

in which T shows the transformed variable. So, the probability integral becomes

$$P_f = \int_{-\infty}^{-\frac{\mu_M}{\sigma_M}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt \quad (3.15)$$

This integral includes the integration of the standard normal distribution which can be represented as

$$P_f = \Phi\left(-\frac{\mu_M}{\sigma_M}\right) \quad (3.16)$$

It can be observed from Eq. (3.16) that P_f is a function of the ratio of the mean value of the safety margin to its standard deviation, which is called the reliability index or safety index. This is denoted by β , for which more detailed information is given in later sections of this chapter.

Although the stress-strength approach can be easily used for reliability analysis, it has some drawbacks that must be taken into account. First of all, the joint probability density distribution is often unavailable or difficult to obtain in practice. Even where the required distributions are available, the limit state function cannot be represented

in linear form as in Eq. (3.10) for all structural reliability problems. This equation also requires explicit classification of random variables in two groups; i.e. stress and strength, and it may be very difficult to get this classification for many engineering analyses. Therefore, some analytical approximation methods have been developed, which make the computation of Eq. (3.6) simpler and applicable to general structural reliability problems. One of the most widely used approximation methods is the First-Order Second-Moment (FOSM), for which more detailed information is given in the following section.

3.4 The First-Order Second-Moment Code Formulation (FOSM)

As outlined in the previous section, the requirement of having full probability distribution functions for random variables is always difficult to obtain in practice. The available information and data may be sufficient only to evaluate the first and second moments; namely, the mean values and variance of the random variables. For these reasons, a method called the First-Order Second-Moment (FOSM) has been developed, based on first-order Taylor series approximation of the limit state, and makes use of only second-moment statistics (mean and variance) of the random variables. With the second moment approach, the reliability is approximated with a function of the first and second moments of design variables. Therefore, structural reliability analysis must be restricted to functions of these two variables, for which Cornell [Cornell 1969] suggested the first application to be put in practice. For simplicity, this approach can be explained by using the simple two-variable approach as is used in the stress-strength interference model, and the application of this method to more general structural reliability problems including multiple random variables is given later in this section. In this approach, the limit state equation can be given as

$$M = X - Y$$

By assuming that X and Y are statistically independent normally distributed random variables, the variable M is also normally distributed. Therefore, the mean value and the variance of the limit state can be easily obtained as

$$\mu_M = \mu_X - \mu_Y \quad \text{and} \quad \sigma_M^2 = \sigma_X^2 + \sigma_Y^2$$

Thus, the probability of failure is given by

$$P_f = P[M < 0] = \Phi\left(-\frac{\mu_M}{\sigma_M}\right) \quad (3.17)$$

From Eq. (3.17), the probability of failure can be expressed as the ratio of the mean value of M to its standard deviation, named as the *safety index* or *reliability index* and denoted as β

$$\beta = \frac{\mu_M}{\sigma_M}$$

$$P_f = \Phi(-\beta) \quad (3.18)$$

Although this approach can simplify the original complex probability problem into a simple problem represented in linear terms, it must be noted that the FOSM method is only used if all the variables are statistically independent normally distributed random variables, or can be transformed into such. The limit state is defined as a simple additive or multiplicative function of these random variables. If this is not the case, the point on the limit state function determined by the mean values of random variables might be far from the maximum likelihood point called the design point. So, the POF results based on this approach only reflect the POF of the limit state function at the mean value of random variables. In order to overcome the deficiencies of the FSOM, Hasofer and Lind [Hasofer 1974] proposed to linearize the general limit state function at the design point, outlined in the following section.

3.4.1 The Hasofer-Lind Method

Hasofer and Lind [Hasofer 1974] first defined the reduced variables in a similar way of the transformation random variables as given in Eq. (3.14) by means of

$$X'_i = \frac{X_i - \mu_{X_i}}{\sigma_{X_i}} \quad (3.19)$$

where X'_i is a random variable with mean zero and unit standard deviation. The original limit state $g(X) = 0$ is transformed into the reduced limit state $g(X') = 0$ by replacing X_i with X'_i in Eq. (3.19). This method can be explained by taking a limit

state in the following form

$$M = X - Y$$

If the reduced variables are used in place of X and Y , the equation given above will be

$$\sigma_X X' - \sigma_Y Y' + \mu_X - \mu_Y = 0 \quad (3.20)$$

This new form of the limit state equation is a linear expression, by which the reliability index can be associated with the position of the limit state to the origin, and the distance can be obtained as a closed form expression from analytical geometry as

$$\beta = \frac{\mu_X - \mu_Y}{\sqrt{\sigma_X^2 + \sigma_Y^2}} \quad (3.21)$$

This explanation must be generalised in order to apply the Hasofer-Lind method to more general reliability problems in which more than two random variables are involved, as well as for the performance function behaving nonlinearly. This generalisation can be explained by classifying the type of the performance function into two i.e. linear performance function and general type of performance function.

3.4.2 Linear Performance Function

For linear performance functions, the exact solution of the POF is easily computed from Eq.(3.18), since there is only one point on the limit state having the minimum distance from the origin of the reduced space to the limit state. Although a more detailed explanation on the computation of the probability of failure is given in the literature [Wu 1987], the basic steps are summarised here as well because this approach forms the basis of the developed response surface method, explained in Chapter 5.

Assume that a linear performance function is represented by a simple linear polynomial:

$$g(\mathbf{X}) = a_0 + \sum_i a_i X_i \quad (3.22)$$

where a_0 and a_i are the coefficients of the function to be determined. As the value of the performance function on the boundary between the safe state and the failure state is $g(\mathbf{X}) = 0$, so the limit state equation will be:

$$g(X) = a_0 + \sum_i a_i X_i = 0$$

This equation can be given in terms of the reduced variables as below:

$$a_0 + \sum_i a_i (\sigma_{X_i} X'_i + \mu_{X_i}) = 0$$

The distance of the limit state to the origin of the reduced varieties X' , namely β , is given as

$$\beta = \frac{a_0 + \sum_i a_i \mu_{X_i}}{\sqrt{\sum_i (a_i \sigma_{X_i})^2}} \quad (3.23)$$

After obtaining the reliability index, the probability of failure can be easily computed from Eq. (3.18). It should be noted that this method is only applied to those reliability problems having an explicit linear performance function. However, structural reliability problems are mostly implicitly defined, or the performance functions are mostly in a non-linear form, for which the following approach can be utilised to resolve the problem.

3.4.3 Non-linear Performance Function

In the case that the performance function is non-linear, the reliability index cannot be obtained as easily as in the linear case, and some approximation techniques to the limit state must be utilised in order to find out the reliability index. There are two basic approximation methods in structural reliability analysis, which are the *First Order Reliability Method* or briefly the FORM and the *Second Order Reliability Method* or briefly the SORM, respectively. The definition of these terms can be given as:

the FORM or the SORM is used to approximate the failure surface; that is, the limit surface ($g(X)=0$), by a single or a set of, first/second order surfaces and then to calculate the failure probability using these approximations [Karamchandani 1992].

In the FORM, the performance function is expanded in first-order Taylor series at the design point where the SORM makes use of second order (quadratic) hypersurfaces.

Although the SORM gives better results than the FORM in terms of the reliability index, the theory and the implementation of the SORM is more difficult. Thus, a few studies have been carried to apply this method to reliability problems in engineering design so far [Breitung 1983], [Der Kiureghian 1987], [Fiessler 1979], [Zhao 1997], [Zhao 1999a], for which a more detailed explanation for both methods is given in the following sections.

3.4.4 First-Order Reliability Method (FORM)

In this method, the design point is used as the point where a tangent plane is formed to approximate the limit state, then the required reliability index can be evaluated as in the linear limit state case as given in Eq. (3.18). The minimum point can be determined by resolving the equation defined as an optimisation problem as:

$$\begin{aligned} &\text{minimise } \beta = \sqrt{\mathbf{X}'^T \mathbf{X}'} \\ &\text{subject to } g(\mathbf{X}) = 0 \end{aligned} \quad (3.24)$$

where $(\mathbf{X}'^T \mathbf{X}')$ indicates the distance from the limit state to the origin of the transformed space while the performance function is given in normal space. The above optimisation problem can be solved by applying any constraint-based optimisation method, one of which is the Lagrangian multipliers [Ang 1984]. A simple iterative method was proposed by Rackwitz [Rackwitz 1976]. Having found

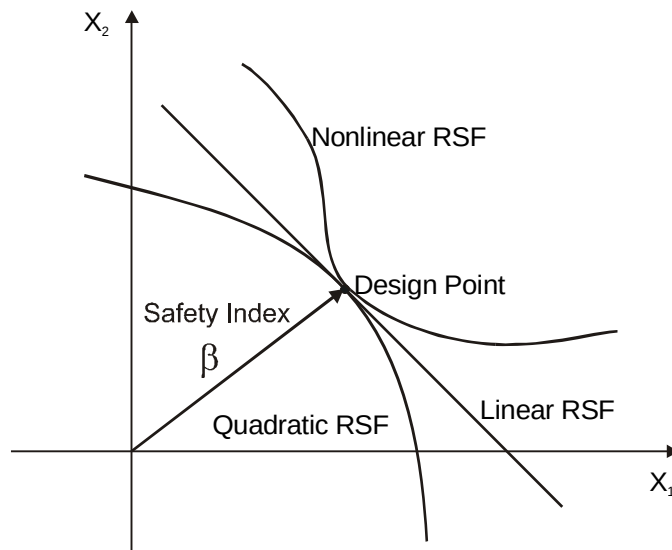


Figure 3.3: Different limit states having the same design point.

the reliability index, the POF can be easily computed by utilising Eq. (3.18). Although this approach is widely used to resolve structural reliability problems because of its simplicity and the availability of algorithms that can be easily applied to the problems,

it does not result in the same reliability index for different equivalent formulations of the same limit state function [Sundararajan 1994]. Ditlevsen [Ditlevsen 1979] also showed that the design point on the failure surface could be the same for the different types of performance functions, illustrated in Figure 3.3. Although these three limit states yield different values for the POF in MCS, they all have the same safety index resulting in the same POF in the FORM.

Another deficiency related to the FORM is the convergence problem as pointed out by Wang and Grandhi [Wang 1996]. This deficiency may be encountered for problems having a highly non-linear performance function since the Hasofer–Lind approach constructs a linear approximation to the limit state at every search point and finds the safety index for these points. This search algorithm may not result in a correct design point because of the linearisation of the given performance function. Therefore, a more advanced method utilising the basic principle of the FORM was suggested by Fiesler [Fiessler 1979], called the second-order reliability method (the SORM), explained below.

3.4.5 Second-Order Reliability Method (SORM)

In order to improve the accuracy of the FORM in terms of the deficiencies of the linear approximation to the performance function, an improved method, called Second Order Reliability Method, the SORM, can be used. The SORM utilises a second-order Taylor expansion of a performance function in a standard space $g(\mathbf{U})$ at the design point $\mathbf{U}^*$ as given below:

$$g(\mathbf{U}) = \beta_F - \boldsymbol{\alpha}^T \mathbf{U} + \frac{1}{2} (\mathbf{U} - \mathbf{U}^*)^T \mathbf{B} (\mathbf{U} - \mathbf{U}^*) \quad (3.25)$$

where

$$\boldsymbol{\alpha} = \frac{\nabla g(\mathbf{U}^*)}{|\nabla g(\mathbf{U}^*)|}, \quad \mathbf{B} = \frac{\nabla^2 g(\mathbf{U}^*)}{|\nabla g(\mathbf{U}^*)|}, \quad \beta_F = \boldsymbol{\alpha}^T \mathbf{U}^*$$

β_F indicates the reliability index from the FORM. The final form of Eq. (3.25) is obtained by using a rotated space of $\mathbf{U}$ in which the axis of the rotated co-ordinate system coincides with the perpendicular from the origin to the tangent hyperplane at the design point [Der Kiureghian 1987], which is achieved by an orthogonal transformation as

$$\mathbf{Y}' = \mathbf{R}\mathbf{Y}$$

where $\mathbf{R}$ is obtained by means of a standard Gram-Schmidt algorithm [Sundararajan 1994]. A second-order surface in this rotated space can be given as:

$$y'_n = \beta + \frac{1}{2} \mathbf{y}'^T \mathbf{A} \mathbf{y}'$$

in which $\mathbf{A}$ is a $(n-1) \times (n-1)$ second-derivative matrix. The elements of $\mathbf{A}$ are obtained from

$$a_{ij} = \frac{(\mathbf{R}\mathbf{D}\mathbf{R}^T)_{ij}}{|\nabla g(\mathbf{y}^*)|} \quad (i, j = 1, 2, \dots, n-1) \quad (3.26)$$

where a_{ij} shows the principal curvature, $\mathbf{D}$ is the $n \times n$ second-derivative matrix of the limit state surface in the standard normal space evaluated at the design point.

In the literature, several methods have been proposed to obtain the POF from the quadratic approximation to the limit state [Fiessler 1979] [Breitung 1983] [Tvedt 1988] [Koyluoglu 1994], for which a review of those approximations was given by Zhao and Ono [Zhao 1999a]. The POF expression based on the SORM was also proposed by Breitung [Breitung 1983] as:

$$P_f \approx \Phi(-\beta) \prod_{i=1}^{n-1} (1 + \beta \kappa_i)^{-1/2} \quad (3.27)$$

where κ_i denotes the i th main curvature of the limit state at the design point, which is obtained from the eigenvalues of $\mathbf{A}'$ given as

$$|\mathbf{A}' - \kappa \mathbf{I}| = 0$$

Although the SORM improves the results obtained from the FORM, it can suffer the same deficiencies of the FORM in terms of finding the design point. In order to obtain

the principal curvatures κ_i , two matrix rotations are required in which the Hessian matrix is obtained. These rotations can be very costly if the number of random variables is large, and the performance function requires the performance of a complicated and time-consuming analysis. Furthermore, there is no readily available algorithm to apply SORM to the reliability problems though a few studies have been carried out quite recently [Zhao 1999a] [Zhao 1999b].

It should be noted that all the methods mentioned in this chapter take into account the random variables as uncorrelated normally distributed random variables. However, most engineering applications involve more complicated random variables such as correlated types or those having different distribution types. Therefore, several methods have been proposed to transform random variables to uncorrelated standardised random variables such as Rosenblatt transformation [Rosenblatt 1969] in which independent normal variables are obtained by means of the inverse of the CDF of non-normal variables.

Instead of using these analytical methods, there is another alternative method that can be used to obtain reliability data such as the probability of failure, design point, so on, based on simulation of the reliability problem. One of the most widely used simulation methods is the Monte Carlo method outlined in the following.

3.5 Monte Carlo Simulation

As indicated in the previous section, several analytical methods, such as the FORM and the SORM in structural reliability analysis, have been developed so far to find a solution to the problems of structural reliability analysis. These analytical methods, however, need to take into account the entire integral given in Eq. (3.6) that defines the POF of the given structural reliability problem. As pointed out previously, it is very difficult to solve this integral and for some problems it is almost impossible to even form the integral. *Monte Carlo Simulation* (MCS) methods can be utilised to find the solution to the integral. Alternatively, the exact result with the actual performance function can be repeatedly evaluated in a MCS and the distribution of the resultant sample compared with that obtained using the FORM/SORM in order to verify the validity of using the FORM/SORM [Wang 1996]. In this approach, simulation is utilised directly in the system without the need to establish the

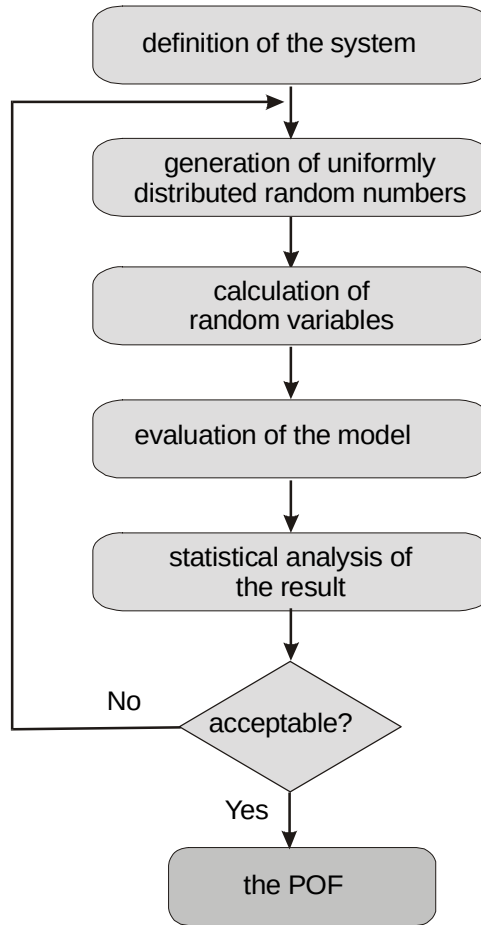


Figure 3.4: The main steps of Monte Carlo simulation.

differential equations that describe the behaviour of the system. The only requirement is that the elements affecting the system must have a known PDF. Once the PDF is known, MCS can proceed by random sampling from the PDF of the elements affecting the system, and repeated evaluation of the model. A result sample is thus constructed. Convergence can be checked by calculating a statistic of the results sample, such as the mean or the estimate of the POF, and observing the gradual reduction in the variance of this statistic as the sample size increases. If the variance of the statistic is not acceptable, another sample of input values is generated, and the results of evaluating the model on the new input sample are combined with the existing results sample to give an increased results sample size.

To obtain the POF from MCS, the failure equation (3.6) can be rewritten in the form that an indicator function $I[g_i(\mathbf{X}) \leq 0, i = 1, \dots, m]$ is defined, such that $I[\] = 1$ if $\mathbf{X}$ is in the failure domain, D , and $I[\] = 0$ otherwise as:

$$P_f = \int_{\text{all } X} \dots \int I[] f_X(x) dx \quad (3.28)$$

an unbiased estimation of P_f can be given by

$$P_f \approx \frac{1}{N} \sum_{j=1}^N I[g_i(x_j) \leq 0, i = 1, \dots, m] \quad (3.29)$$

in which x_j represents the j th vector of random observations from the distribution function $f_X(\cdot)$. The steps needed to obtain the value of POF from Eq. (3.29) are illustrated in Figure 3.4. As can be seen from the figure, MCS involves generating sequences of two types of numbers; *uniformly distributed random numbers* and *random variables*, and a statistical analysis of the results of evaluating the model using the random variable samples as inputs, as explained in the following sections.

3.5.1 Generation of Numbers Required for MCS

Two different sets of numbers are required in order to carry out the simulation, one of which is a series of random numbers with uniform distribution; i.e. rectangular distribution, and the other is a set of samples of random variables in accordance with the PDF of the system inputs. The random variable samples are calculated from the series of random numbers, and the relationship between the random numbers and the samples of random variables can be explained as follows:

Suppose a random variable X has a cumulative distribution function (CDF) $F_X(x)$. For a given CDF $F_X(x) = u$, the value of x can be easily obtained by taking the inverse of F_X :

$$x = F_X^{-1}(u) \quad (3.30)$$

Suppose that u is a value of U , the corresponding value of random variable X obtained from Eq. (3.30) will have a cumulative probability,

$$\begin{aligned} P(X \leq x) &= P[F_X^{-1}(U) \leq x] \\ \text{since } P[F_X^{-1}(U) \leq x] &= P[U \leq F_X(x)] \\ &= F_U[F_X(x)] = F_X(x) \end{aligned}$$

which indicates that if $(u_1, u_2, \dots, u_n)$ is a set of values from U , then a set of

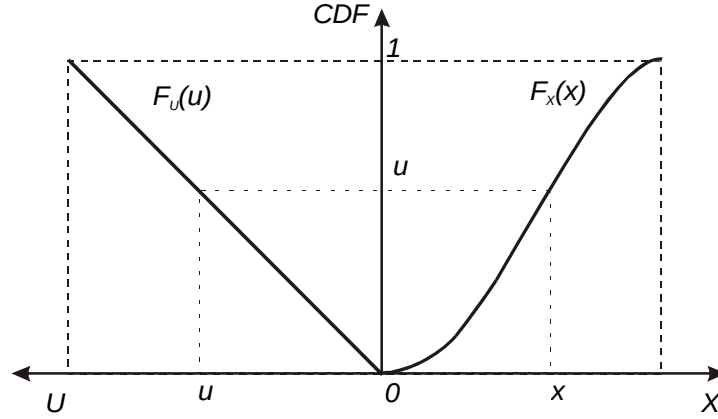


Figure 3.5: The relationship between random numbers U and random variables X .

values for X with the desired CDF $F_X(x)$ is obtained by means of Eq. (3.30) as depicted in Figure 3.5.

3.5.1.1 Generation of Uniformly Distributed Random Numbers

In order to generate samples of random variables for the simulation, a pseudo-random number generator is used to obtain a series of random numbers between 0 and 1 with uniform PDF. There are many algorithms that have been developed to generate random numbers as was outlined by Law and Kelton [Law 1991]. Since the subject of getting a reliable algorithm for producing random numbers with a uniform distribution is more related to computer science, only the method to generate random variables in this study will be focused on. The simulation for structural reliability problems is carried out here using a program written in the C programming language. In the program, a C library function, *drand48()* using the well known linear congruential algorithm [Lehmer 1951] and 48-bit integer arithmetic is utilised to generate random numbers that are distributed uniformly between 0 and 1. Before the first call to *drand48()*, *srand48()* or *seed48()* must be called to initialise a random “seed”. In the present thesis, the following commands are used to generate random numbers:

```
srand48(time(NULL));
```

which generates an initial value for the pseudo-random number sequence. *srand48()* sets the seed for these random number generators. It is important to have

different seeds when we call the functions otherwise the same set of pseudo-random numbers will be generated. *time(NULL)* always provides a unique seed since it returns the time since 1 January 1970, measured in seconds [Curry 1996]. Then, a random number, x , is generated as follows:

$$x = \text{drand48}();$$

Different algorithms and methods can be easily obtained from various sources especially on the Internet [Virtual 2000] [NCSA 2000].

3.5.1.2 Generation of Random Variables

Samples of random variables are generated by utilising one of the methods explained in this section once the random numbers have been obtained. The selection of a proper method depends on the type of CDF of the random variables. For a CDF whose inverse can be obtained analytically, the inverse method given by Eq. (3.30) can be easily used to generate samples of random variables. However, there are many important probability distributions used in structural reliability analysis such as Normal, Lognormal, etc., for which the inverse of the CDF of random variables can not be obtained analytically. Therefore, more advanced methods which are capable of generating samples for random variables are needed.

The methods to generate samples of random variables can be classified into two groups according to the methods applied:

1. General approaches to generating random variables
2. Particular approaches to generating random variables.

The general approaches to generate samples of random variables are utilised regardless of the type of probability distribution function since they are based on general analytical methods. In this approach, samples of random variables are generated by applying one of the methods such as inverse transform, composition, convolution and acceptance rejection method. However, the methods for generating random numbers which are mentioned above are so general that, for some applications, using these methods may not be effective in terms of the computational time spent and the accuracy of the results. Apart from these general approaches, more accurate and straightforward methods which are formed according to the probability

density functions have been considered, and the algorithm for most widely used PDF can also be found in any simulation book [Law 1991] [Rubinstein 1981]. One of these methods will be explained in detail for normally distributed random variables since normal distribution is most widely used PDF type in structural reliability analysis.

3.5.2 A Particular Approach to Generating Normally Distributed Random Variables

If a random variable has a normal density function, the algorithm to generate random variables according to Box and Muller's method [Box 1958] can be summarised as follows :

- Random numbers u_1 and u_2 are generated according to the standard uniform random variable U
- $x_1 = \sqrt{-2 \ln u_1} \cos 2\pi u_2$ and $x_2 = \sqrt{-2 \ln u_1} \sin 2\pi u_2$ are set.

Then, x_1 and x_2 are samples of independent random variables X_1 and X_2 , which have a normal distribution with mean=0 and variance=1. In this method, the random variables are generated in pairs so x_1 could be returned immediately, with x_2 being returned in the next call to the simulation program. Using only a single stream linear congruential generator for producing uniformly distributed random numbers may result in having random variables X_1 and X_2 for which the generated samples x_1 and x_2 samples fall on a spiral in (X_1, X_2) space, rather than independently normally distributed [Law 1991]. In order to avoid the difficulties arising from the algorithm which generates random numbers, Marsaglia and Bray [Marsaglia 1964] proposed the polar method which also generates standard normal random variables, with mean=0 and variance=1. The algorithm describing the polar method is summarised in Figure 3.6.

Another key problem in Monte Carlo simulation is to decide what the sample size must be. The required sample size can be determined by evaluating the variance of the output of the system, that is, the smaller the standard deviation is the more accuracy can be obtained. The variance of estimated POF can be calculated approximately as

$$\text{Var}(P_f) \cong \frac{(1 - P_f)P_f}{N} \quad (3.31)$$

the coefficient of variation of the POF can also be used to assess the statistical accuracy of Eq. (3.29) as

$$COV(P_f) \cong \frac{\sqrt{\frac{(1-P_f)P_f}{N}}}{P_f} \quad (3.32)$$

When N approaches infinity, $Var(P_f)$ and $COV(P_f)$ approach zero, indicating that a large number of simulations must be used to obtain a reasonably small value for $Var(P_f)$. Therefore several research studies have been done to reduce the sample size of simulation, some of which are outlined in the next section.

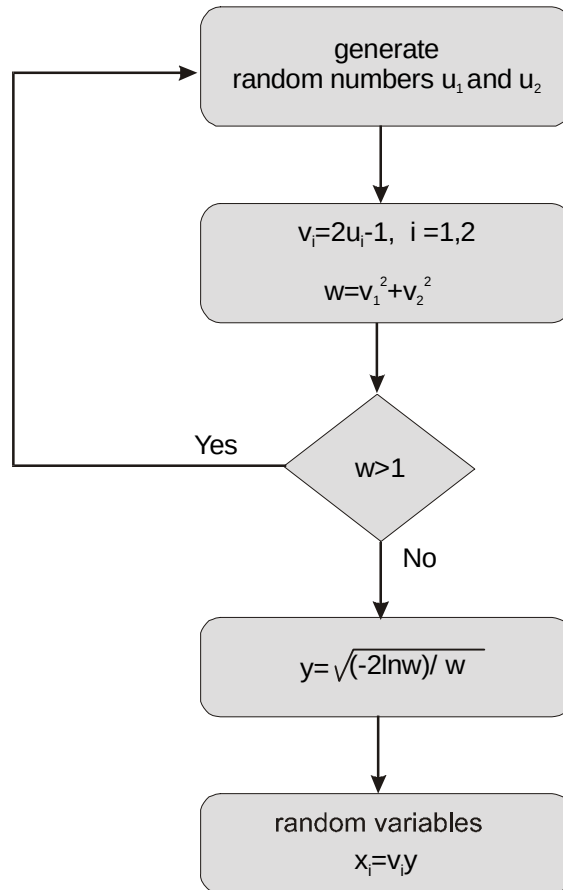


Figure 3.6: Polar method of generating random variables.

3.6 Variance Reduction Techniques

The results from a simulation process are subject to sampling errors that decrease while the sample size is increased. However, the variance in the estimates of the output distributions' parameter due to a finite sample size can be reduced without increasing the sample size of simulation performed by using what are known as *Variance Reduction Techniques* (VRT). Some of the most widely used VRTs are Antitethic Varieties, Correlated Sampling, and Control Varieties, for which more detailed information can be obtained from the literature [Ang 1984] [Law 1991] [Rubinstein 1981]. On the other hand, in structural reliability analysis, more specific techniques, such as Adaptive Sampling and Importance Sampling, have been applied in accordance with the features of the simulation in structural reliability analysis. Importance Sampling has gained more interest in the application to structural reliability problems since the theory is simple, and the sample size is far less than for crude MCS [Melchers 1990] [Engelund 1993] [Song 1997], the basic concepts are outlined in the following section.

3.6.1 Importance Sampling (IS)

As can be seen from Eq.(3.29), for low values of P_f , only the sample vectors $\mathbf{x}_j$ leading to values of $g_i(\mathbf{x})$ equal or less than zero will contribute to the probability calculation. Hence a very large sample size is required for the computation which makes the computational effort for a reasonable estimate of P_f excessive. In order to reduce the computational burden, several importance sampling methods have been suggested by many researchers [Hohenbichler 1981] [Melchers 1990] [Ibrahim 1991] in which the probability equation is rewritten as:

$$P_f = \int \dots \int I \left[\frac{f_{\mathbf{x}}(\mathbf{v})}{h_{\mathbf{v}}(\mathbf{v})} \right] h_{\mathbf{v}}(\mathbf{v}) d\mathbf{v} \quad (3.33)$$

where $h_{\mathbf{v}}$ is the Importance Sampling probability density function, and the P_f is computed by rewriting the above equation as

$$P_f \approx \frac{1}{N} \sum_{j=1}^N \left\{ I \left[g_i(\mathbf{v}_j) \leq 0 \right] \frac{f_{\mathbf{x}}(\mathbf{v}_j)}{h_{\mathbf{v}}(\mathbf{v}_j)} \right\} \quad (3.34)$$

where v_j is a vector of sample values taken from the Importance Sampling function h_v . The main aim in Importance Sampling is to choose an appropriate function type for the Importance Sampling function and hence take the sample in the vicinity of the point of maximum likelihood of $f_X(\cdot)$ which corresponds to the point called the design point X^* in the FOSM. Several suggestions have been proposed which are classified in the reference [Sundararajan 1994].

3.7 Comparison of the FORM Results with That of MCS

Approximating to the limit state by means of a linear function in the FORM can result in a large penalty in terms of the POF as well as the design point that can be used in the later stages of structural reliability analysis. In that case, obtaining an accurate value for the design point sometimes becomes crucial to having a good result in the subsequent stages. Therefore, this section shows how the results from the FORM can be different to the accurate result as well as proposing an alternative approach to overcome the deficiencies of the FORM.

In order to show the deficiencies of the FORM, an example from [Wang 1996] is taken, in which the performance function and the problem parameters are given in Table 3.1.

Table 3.1: Problem parameters of the example taken from [Wang 1996].

Performance		$g(X_1, X_2) = x_1^4 + 2x_2^4 - 20$	
Random Variables	Type of PDF	Mean Values	Standard deviation
X_1	<i>normal</i>	10	5
X_2	<i>normal</i>	10	5

The results given in Table 3.2 are obtained from COMREL [Comrel 1999]. In Table 3.2, the reliability results for the two analytical methods, the FORM and the SORM, are given along with the results that are obtained from MCS, shown in a shaded area, by which the other methods can be evaluated. By the comparison of the POF for the FORM with that of MCS, it can be pointed out that for the reliability results of a nonlinear performance function, as is in this example, the FORM can give

Table 3.2: The results for three methods compared.

<i>Methods Used</i>	<i>FORM</i>	<i>SORM</i>	<i>MCS</i>
<i>Design Point</i>			
<i>X1</i>	1.860	0.1468	1.870
<i>X2</i>	1.504	-0.280	1.275
<i>β</i>	2.360	2.848	2.921
<i>the POF</i>	0.009137	0.002199	0.001744
<i>$g(X)$</i>	2.202	-19.987	-2.48

an erroneous results because of the linearisation of the performance function as given in Figure 3.3. Although the condition given in Eq. (3.24) for the value of the performance function, which is required to be as close to zero as possible, is satisfied, the rest of the reliability data such as the reliability index and the POF is not accurate. The reason why the design point value for the FORM is almost the same as that of MCS can be explained as the FORM searches the minimum point on the limit state in the reduced space, which can be the same value that the exact solution has. However, using this design point along with the reliability index cannot give correct results because the FORM assumes the design point on a linear limit state. By using the SORM, better results are obtained for the reliability index and the POF, but it gives a poor result for the design point since the SORM prefers to have some penalty on the value of the design point in order to get correct results for the reliability index and the value of the POF.

From the above considerations, a different approach is used in this thesis in order to obtain the design point. In this approach, the design point does not represent the minimum distance from the limit state to the origin of the reduced space as illustrated in Figure 3.3. However, the design point is determined as a point equivalent to the POF obtained from MCS for which a more detailed explanation and its application to structural reliability problems will be given in Chapter 5.

3.8 Conclusion

In this chapter, background information and a basic theoretical review of the most commonly used structural reliability methods have been explored. Firstly, some theoretical background of the analytical methods has been given by explaining a structural reliability problem in terms of stress-strength interference model since this model forms the basis of the more improved methods such as the FORM and the SORM. According to the type of performance function, a different approach must be used as outlined in Section 3.4. Besides these analytical methods, a different approach based on simulation of the structural reliability problem has been also given, by which the exact solution can be obtained provided that the sampling of random variables is carefully designed. As the main problem of any simulation method such as MCS is to carry out a number of calls to the performance function, which can be very time consuming and impractical for some problems requiring time consuming analyses, some of the variance reduction techniques have also been given in this chapter. Finally, it has been shown that the results obtained from the FORM should in general not be used alone since the FORM approximates to the performance function by a linear function for which the result of either the design point or the POF is not correct for nonlinear performance functions. Therefore, another approach that can give better results on the reliability data is used in the thesis as will be explained in detail in Chapter 5.

In order to use MCS in a time consuming structural reliability analysis, an approximation method, called the Response Surface Method, can be utilised by which the performance function is replaced with a polynomial that can be either as simple as linear function or a higher polynomial function, depending on the response of the performance function.

Chapter 4

Response Surface Methods

This chapter defines the necessity of using approximation methods for structural problems for which structural reliability methods cannot be effectively used owing to an implicit performance function or the requirement for time consuming repetitive analyses. In the first section, an overview of an approximation technique called the response surface approach is given. The second section contains more specific information on the use of the response surface method when applied to structural reliability problems along with explanation on the most widely used response surface method for structural reliability analysis. In the third section, existing methods for modelling error assessment and model choice for a response surface function are given. Some deficiencies of the existing response surface methods for structural reliability analysis are described, along with the necessity of using a more advanced approach that can alleviate those deficiencies. Finally, another system definition method, neural network, is explored.

4.1 Overview of the Response Surface Methods

The *Response Surface Method* (RSM) can be described as a collection of statistical tools and methods for forming and exploring an approximate functional relationship between a response variable and a set of design variables [Yao 1996]. The most widely used form of the functional relationship is a low order polynomial that is referred to as a *Response Surface Function* (RSF). In structural analysis, the RSM normally/generally specifies the relationship between the response of a structure and the variables that affect the response. The relationship is expressed by forming a low order polynomial as follows:

$$\hat{y} = f(\mathbf{x}) + \varepsilon \quad (4.1)$$

in which $\hat{y}$ is the response vector (response matrix) and $\mathbf{x}$ specifies a vector including the experimental points, for which more detailed information is given in Section 4.2. The RSM was originally developed to analyse the results of physical experiments and to form empirically-based smooth models of the observed response values, thus any noise that is associated with the experiments is filtered out [Box 1951]. Therefore, it has been used especially for chemical and physical experiments. However, its general usefulness in approximation processes was realised, and the application field of the

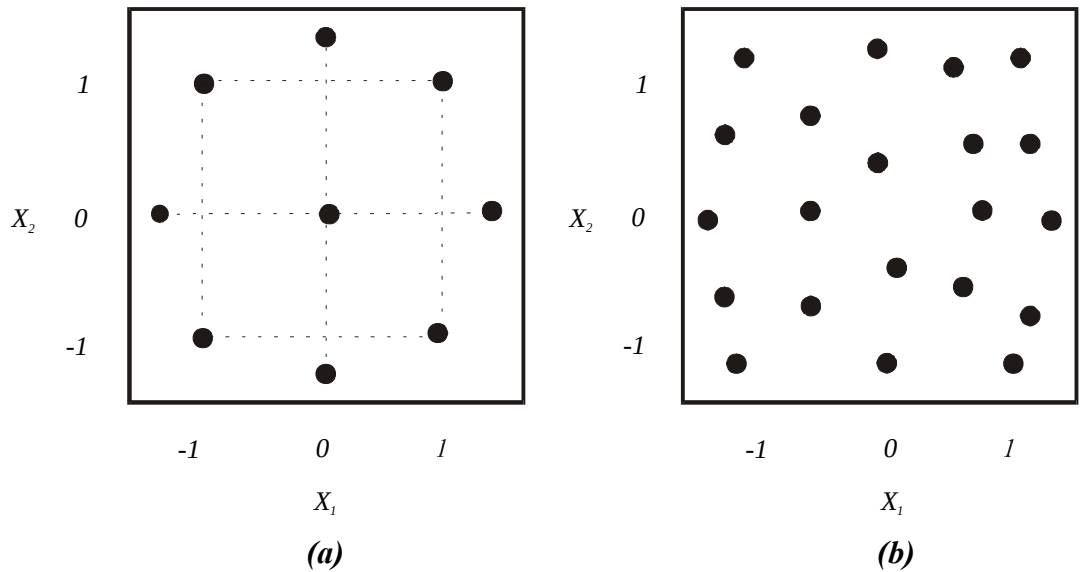


Figure 4.1: Regularly shaped design CCD (a), and irregularly shaped design (b).

method has been spread into many engineering areas, especially in structural optimisation. Due to the similarities between detailed numerical simulations in optimisation problems and in physical experiments, the RSMs have been successfully applied to structural optimisation problems [Schmit 1974] [Barthelemy 1993]. Although the RSMs differ according to the fields in which they are used, a general RSM can be classified into three main steps:

- Choosing the experimental points (design of experiment)
- Choosing the type of model function (model choice)
- Evaluation of the modelling error (model error evaluation)

Design of Experiments (DoE) is used to select a small set of data points to construct the RSF [Box 1987]. Generally, statistical DoE is performed using a minimum-variance criterion in which all errors are assumed to be random errors or variances. A classification of DoE can be given according to whether experimental points are selected in a regular pattern such as a rectangular shape or in an irregular pattern by spreading the sample points out in the design space, where the former is called a regularly shaped design space, and the latter is an irregularly shaped design space. In a regularly shaped space, the allowable range of the variable is arranged as equally-spaced levels as is shown in Figure 4.1(a). By contrast, the experimental points in an irregularly shaped space are spread out to fill the design space as illustrated in Figure 4.1(b). For a regularly shaped design space, the most widely used DoE approaches are *Central Composite Design* (CCD) [Box 1951] and the Box-Behnken method [Box 1960]. For an irregularly shaped design space, a computer-generated design is used, of which the most popular design is D-optimal design for which a more detailed explanation can be found in [Myers 1995].

There is another approximation method, based on minimising the bias error of the resulting RSM, which is known as minimum-bias design. In order to explain what bias error means, errors associated with a response model can be considered to be classified as either random errors due to variance in the output parameters or modelling errors due to the use of inadequate model type. So a minimum bias design is one that tries to minimise modelling error. A clearer representation of the errors can

be written as follows in terms of the *Mean Squared Error* (MSE):

$$E[\hat{y}(\mathbf{x}) - f(\mathbf{x})]^2 = \underbrace{E[\hat{y}(\mathbf{x}) - E\hat{y}(\mathbf{x})]^2}_{\text{Variance}} + \underbrace{E[E\hat{y}(\mathbf{x}) - f(\mathbf{x})]^2}_{\text{SquaredBias}} \quad (4.2)$$

where E refers to the expected or mean value, $\hat{y}(\mathbf{x})$ represents the RSF, and $f(\mathbf{x})$ is the true model to be approximated by $\hat{y}(\mathbf{x})$. As can be seen from Eq.(4.2), bias error is associated with the use of an inappropriate type of RSF. Large bias error is easily introduced since low order polynomials are generally used for the approximation to the true model.

In order to find minimum-bias designs, there are several methods developed that differ depending on whether the design space is regular or irregular. For regularly shaped designs, a closed form solution for finding the minimum-bias design points is explained in detail [Myers 1995]. For an irregularly shaped design space (as shown in Figure 4.1(b)), there is no closed form solution and the minimum-bias data points must be obtained numerically [Venter 1997].

There is an ongoing discussion in the literature on which type of design is most appropriate for use in a computer-based experiment [Sacks 1989] [Simpson 1998]. As most DoE are developed for physical experiments, they include some features that are questionable to use in a computer-based design, which would normally be the case in a structural reliability analysis, owing to the absence of random errors in the computer model. Unlike physical experiments, running the code with the same inputs will always produce the same output in computer-based experiments. Therefore, the use of repeated experimental points that is widely used in physical experiments becomes meaningless for computer-based experiments. Therefore, the central runs in any DoE methods become meaningless because the same response are obtained in each run for the same experimental points. Thus, for computer based experiments, some alternative methods have been suggested, one of which is *Design and Analysis of Computer Experiments* (DACE) [Welch 1989], which is based on a minimum-bias design from which the RSF is obtained based on the experimental points having the minimum-squared bias value given in Eq. (4.2). This approach was used to generate RSFs for structural optimisation problems [Venter 1997] [Simpson 1998].

Some research has been done to compare DoE and minimum-bias design

approaches, to find the most appropriate design methodology for computer based experiments [Giunta 1998] [Venter 1998]. In this research, the authors reached a conclusion that using minimum-bias designs does not improve the predictive capabilities of the RSM. It was also indicated that implementing a minimum-bias design for an irregularly shaped design space is very difficult. This might be the reason why DACE models have found limited applications in engineering design. Other reasons can be given as the lack of readily available examples, the added complexity of fitting a DACE model, and the non-availability of a closed form polynomial equation which can be easily used for an irregularly shaped design which requires using a numerical method or an optimisation method such as genetic algorithm to find the design points having the minimum squared bias value [Simpson 1997].

The above comments on RSMs apply to their use in deterministic structural design, especially in the field of structural optimisation. However, the application of the RSM in structural reliability analysis differs because of the nature of structural reliability analysis, and in this case the additional remarks below apply.

For a structural reliability analysis, Bucher [Bucher 1989] first applied a modified version of the traditional RSM in which experimental points are chosen around the mean values of the random variables, which form a matrix called the *design matrix*. As a RSF, a second-degree (quadratic) polynomial is used to fit the experimental points. The number of experiments is also equal to the number of unknown coefficients in the RSF. Bucher's RSM includes two stages to form the final RSF to be used in place of performance function. In the first stage, a second degree polynomial RSF is formed from the design matrix explained above, and the design point is obtained by applying the FORM. For the second stage a new design matrix is established based on the design point from the first stage. This design matrix is used to form another second degree RSF, which is the final RSF, with an improved fit to the performance function. Although this method of updating the polynomial RSF is assumed to result in a sufficiently accurate approximation to the limit state function after two iterations, as Rajashekhar and Ellingwood [Rajashekhar 1993] pointed out, the accuracy of the reliability results from application of Bucher's RSM depends on the characteristics of the limit state. Therefore, one cycle of updating may not be

sufficient. Thus, a criterion to carry out further updating was proposed by the authors, based on an observation of the distance between the design point and the centre points obtained based on Eq. (4.9). The iteration process is repeated until the distance is acceptably small. In their improved RSM, the authors also studied the effect of the selection of the experimental points from one side of the distribution function of random variables in order to choose the experimental points to be as close to the region of interest as possible. Another criterion on whether mixed terms such as x_1x_2 should be added or left out from the RSF was also proposed, in which the necessity of the terms is determined by comparing the result of the performance function at the centre point and the experimental point.

The selection of the experimental points is very important in terms of having a RSF close the limit state since it defines the failure state or safe state of the component for which more detailed information was given in Chapter 3. Therefore, an improved sequential RSM which is mostly based on the selection of experimental points close to the original failure surface by using the gradient projection method as was depicted in Figure 4.5 was proposed by Kim and Na [Kim 1997]. In this method, a linear RSF is used to get the design point and the reliability index rather than the commonly used quadratic RSF, by projecting the experimental points on the response surface obtained in the preceding iteration.

The application of Bucher's method has been carried out in many areas ranging from the aerospace industry to civil engineering [Liu 1994] [Zhao 1997]. In addition, the method has also been used to improve the accuracy of the SORM, as well as reducing the computational burden of getting the reliability information of the problem by introducing a simple RSM based on Bucher's design as a method for a point-fitted second-order surface. The RSF obtained from this approach is used in the SORM in place of the actual limit state [Zhao 1999b].

In this section, a general overview of the conventional RSM has been given. The application of the RSM to the structural reliability analysis was also described, and the most widely used methods were also explained. In the next section the theoretical background to the RSM will be explained in more detail.

4.2 Theoretical Background of the Response Surface Methods

In structural analysis, the RSM normally/generally specifies the relationship between the response of a structure and the variables that affect the response. The relationship is expressed by forming a low order polynomial as follows:

$$\hat{y} = f(\mathbf{x}) + \varepsilon \quad (4.3)$$

in which $\hat{y}$ is the response vector (response matrix) and $\mathbf{x}$ specifies a vector including the experimental points which is represented in a matrix notation called the design matrix, and $f(\mathbf{x})$ is the response of $\mathbf{x}$. The modelling error is represented by the vector ε , which is caused by the differences between the predicted and actual response. More clearly, Eq. (4.3) can be written in matrix notation as [Venter 1996]

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b} + \varepsilon \quad \left\{ \begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{matrix} \right\} = \left[\begin{matrix} 1 & x_{12} & x_{13} & \cdot & \cdot & x_{1p} \\ 1 & x_{22} & x_{23} & \cdot & \cdot & x_{2p} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_{n2} & x_{n3} & \cdot & \cdot & x_{np} \end{matrix} \right] \left\{ \begin{matrix} b_1 \\ b_2 \\ \cdot \\ b_n \end{matrix} \right\} + \left\{ \begin{matrix} \varepsilon_1 \\ \varepsilon_2 \\ \cdot \\ \varepsilon_n \end{matrix} \right\} \quad (4.4)$$

in which the vector $\mathbf{b}$ includes estimates of the unknown coefficients b_i of the response surface function $f(\mathbf{x})$, n denotes the number of experiments and p is the number of terms in the RSF. The unknown coefficients b_i of the RSF are generally estimated using the well-known least squares method [Kleinbaum 1987] [Kreyszig 1993]. This method fits a straight line to the given experimental points so that the sum of the squares of the distance of those points from the straight line is minimum, where the distance is measured in the vertical direction. An explicit formulation is as follows

$$\left. \frac{\partial L}{\partial b_i} \right|_{b_i} = -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X}\mathbf{b} = 0 \quad (4.5)$$

in which L shows the total distance between the experimental points and the line:

$$L = \sum \varepsilon_i^2 = \sum_{i=1}^n (y_i - f(\mathbf{x}))^2$$

To estimate the unknown coefficients, n analyses are required, where $n \geq p$. The

solution for finding the coefficients for a linear RSF can be obtained from the following straightforward expression.

$$\mathbf{b} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (4.6)$$

For a non-linear RSF, however, Eq. (4.5) cannot be used to get the coefficients of the RSF. Instead, one of the non-linear regression methods that are explained in detail in the literature [Ryan 1997] [Press 1992] should be used. After forming the RSF, the next step involves the assessment of the model and the estimated coefficients

As noted above, for reliability problems, Bucher [Bucher 1989] reformed the traditional RSM given above so that the response function can fit around a region that might include the design point in structural reliability analysis. By this approach, the RSF is formed in a sub space rather than taking into account the whole design space. Thus, this function can be used in place of the actual performance function for the calculation of the probability of failure, for which the main points of Bucher are described in the following sections.

4.2.1 Bucher's Response Surface Method

The aim of Bucher's RSM is to replace a performance function $g(\mathbf{x})$ by an equivalent response surface function $\bar{g}(\mathbf{x})$ which is mostly given as a second order polynomial as shown in Eq.(4.7) below:

$$\bar{g}(\mathbf{X}) = a + \sum_{i=1}^n b_i x_i + \sum_{i=1}^n c_i x_i^2 \quad (4.7)$$

in which X_i are the basic variables, $i = 1, 2, \dots, n$ shows the number of the basic variables, and the parameters a, b_i, c_i are to be determined. As the number of these free parameters is $2n+1$, only a few calculations are needed to obtain the RSF. In Bucher's RSM, the suggested way of obtaining these parameters is interpolation using the points along the axes x_i which are chosen to be of the form:

$$x_i = \bar{x}_i \pm f_i \sigma_i \quad (4.8)$$

in which $\bar{x}_i$ and σ_i are the mean value and standard deviation of X_i , respectively, and f_i is an arbitrary factor. The interpolation points are selected as indicated in Figure 4.2.

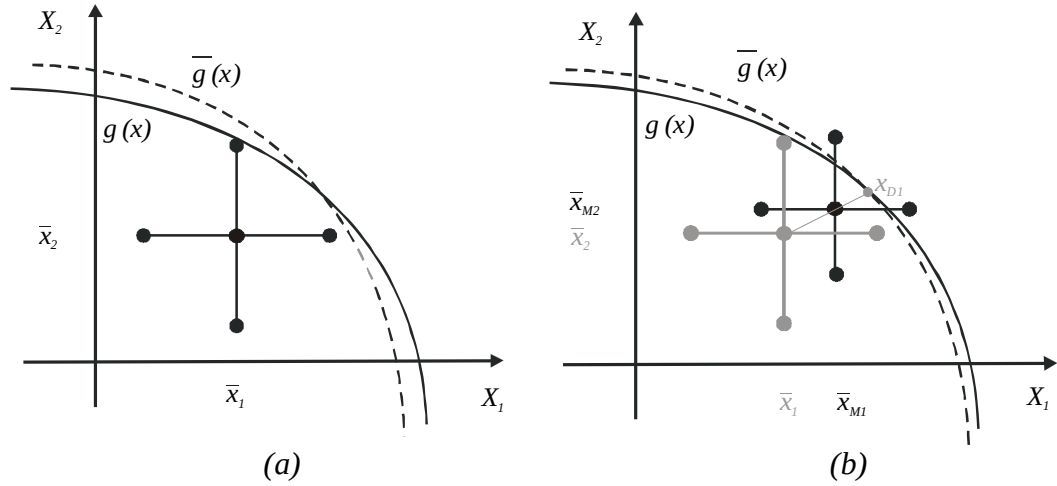


Figure 4.2: The initial points (a) and interpolation points (b) of Bucher's design.

The values of these points are then substituted into the performance function to get the response of $g(\mathbf{x})$, which forms the response matrix. After $\bar{g}(\mathbf{x})$ is formed, this RSF is used along with the information $\bar{x}_i$ and σ_i to obtain an estimate of the design point $\mathbf{x}_D$. One of the key steps is to find the design point because the next iteration will be based on this point which significantly affects the accuracy of the second and final RSF. After $\mathbf{x}_{D1}$ is found the new centre point $\mathbf{x}_M$ for interpolation is obtained on a straight line from $\bar{\mathbf{x}}$ to $\mathbf{x}_{D1}$ as shown in Figure 4.2(b), and an explicit formulation of $\mathbf{x}_M$ is given as

$$\mathbf{x}_M = \bar{\mathbf{x}} + (\mathbf{x}_{D1} - \bar{\mathbf{x}}) \frac{g(\bar{\mathbf{x}})}{g(\bar{\mathbf{x}}) - g(\mathbf{x}_{D1})} \quad (4.9)$$

Once $\mathbf{x}_M$ is found, the new interpolation points are established according to the new centre point $\mathbf{x}_M$ to find a new RSF which is supposed to be closer to the limit state.

A clearer definition of Bucher's method is also given in Figure 4.3. Although the proposed RSM is assumed to give very good results after two iterations, as Rajashekhar and Ellingwood [Rajashekhar 1993] have pointed out further iterations might be needed, as is discussed in more detail in the next section.

4.2.2 Rajashekhar's Response Surface Method

In Bucher's design, only two iterations are carried out to get the RSF which is assumed to be very close to the real performance function after those iterations. However, Rajashekhar and Ellingwood [Rajashekhar 1993] showed that further

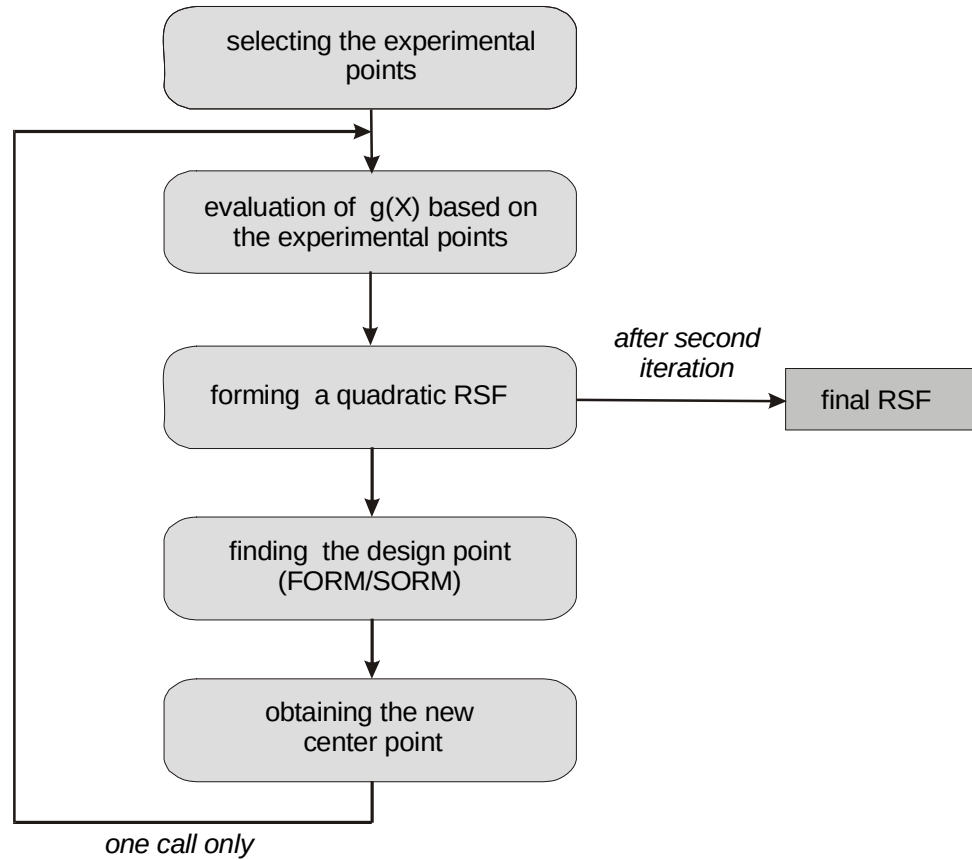


Figure 4.3: Basic steps of Bucher's Method.

iterations might be needed to get a well-approximated RSF. In order to find out how many iterations might be needed, the authors proposed a criterion based on determining the distance between the design point $\mathbf{x}_D$ and the centre point $\mathbf{x}_M$ that is denoted as δ_i , as shown in Figure 4.4. The iteration continues until the criterion set for δ_i is satisfied, which is generally selected close to zero, depending on the required accuracy of the results. As the tail regions of the probability distributions of random variables are most important in the estimation of POF, the authors also studied the effect of the selection of experimental points from the tails of the probability distribution functions of random variables, for which three cases were taken into account.

In the first approach taken, the width of the sampling region is chosen arbitrarily from one side of the distribution function, and the number of the experimental points for each variable is the same as the number of experiments. After the first iteration,

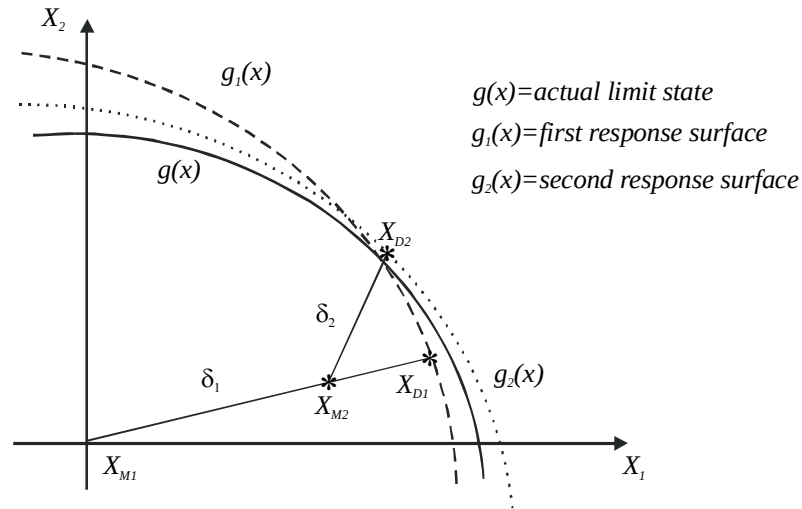


Figure 4.4: Convergence to the design point by reducing δ_i .

the width of the sampling region is reduced around the design point.

In the second approach, the selection of the centre point for the first iteration is carried out by considering the fractions of the distribution function that are determined as the 95% percentile of load or 5% percentile of resistance. Two other points for each variable are chosen on either side of the centre point. After finding the design point, the selection of the experimental points for the successive iteration stages is carried around that point.

In the final approach, the selection of two other points for each variable in the second approach is carried out with respect to their standard deviations as in the Bucher RSM, but the arbitrary function f_i is chosen to be less than that of the preceding iteration, which also reduces the sampling region.

Although more accurate RSFs can be generated from this approach, the authors concluded from the examples given in [Rajashekhar 1993] that sampling in the tail regions of the random variables instead of the entire region does not lead to significant improvements in the response surfaces or reliability estimates.

The effect of additional terms on a quadratic RSF was also studied to determine which term(s) should be added to the RSF. As pointed out, neglecting the mixed terms may be inappropriate as there are situations where cross terms are more important to the actual RSM than the squared terms of the individual variables. In order to

determine whether the cross terms are important in the RSF to be formed, a criterion given in Eq. (4.10) is suggested in which g_c is the response computed at the centre point, g_i is the response computed at the i th experimental point where a new term is added and ε is an arbitrary error value.

$$\left(\frac{g_i - g_c}{g_c} \right)^2 < \varepsilon \quad (4.10)$$

In Eq.(4.10), ε is assigned a low value, given as 0.01 in the paper, and square term is computed for the i th experimental point. If it is larger than ε , the corresponding term is neglected.

4.2.3 Kim's Response Surface Method

In their paper [Kim 1997], Kim and Na propose an improved sequential RSM which is mostly based on the selection of experimental points close to the original failure surface by using the gradient projection method. As the type of the response surface function, a linear one is used to get the design point or the reliability index rather than a more commonly used quadratic RSF.

The authors first point out that selection of experimental points in traditional methods, which are selected along the co-ordinate axis centres at the mean values of random variables, takes no account in positioning of the actual shape of the performance function. In the paper, the authors utilised the gradient projection technique so the experimental points used in response surface formulation are positioned close to the performance function as depicted in Figure 4.5. The main steps of Kim's proposed algorithm are as follows:

- The same first step of Bucher's design is utilised with a linear type of RSF, and the design point is found by applying the Rackwitz-Fiessler algorithm [Rackwitz 1978].

A new set of sampling points which is equivalent to new centre points in Bucher's design is selected by using the vector projection by which the conventional sampling points are projected on the RSF obtained in the preceding stage. The basic steps of computing these sampling points is explained in the following, which is also depicted in Figure 4.6.

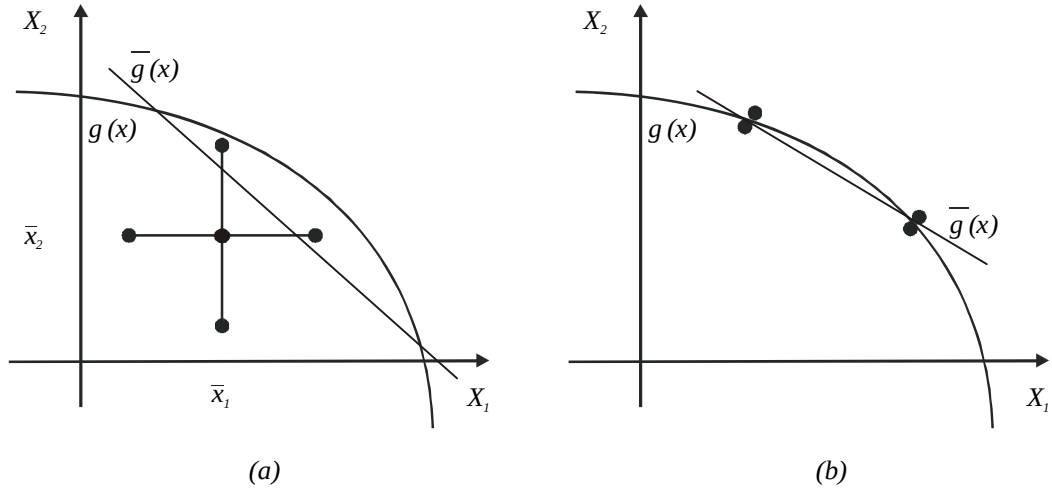


Figure 4.5: Experimental points of Bucher's design (a) and Kim's design (b).

The points for the second iteration are selected, based on the following equation:

$$\mathbf{x}_s^i = \mathbf{x}_D \pm f_i \sigma_i \sqrt{n-1} \boldsymbol{\varepsilon}^{iT} \boldsymbol{\delta}^i \quad (4.11)$$

in which $\mathbf{x}_D$ represents the design point obtained in the preceding iteration, f_i is an arbitrary factor as in Bucher's design, σ_i is the standard deviation of the random variable, n shows the number of the random variables, $\boldsymbol{\varepsilon}^{iT}$ represents a parameter for which the following values are proposed by the authors:

$$\boldsymbol{\varepsilon}^{iT} = [\varepsilon_1 \dots \varepsilon_n] \begin{cases} \varepsilon_k = 1.0 & \text{for } k = i \\ \varepsilon_k = 0.9 & \text{for } k \neq i \end{cases} \quad (4.12)$$

The vector given in Eq.(4.12) above positions the sampling points slightly from the failure surface. The projected unit vector, $\boldsymbol{\delta}^i$, is given as

$$\boldsymbol{\delta}_j^i = \frac{h_j^i}{\sqrt{\sum_{k=1}^n (h_k^i)^2}} \quad (4.13)$$

An explicit formulation for the parameters given in Eq. (4.13) is as follows:

for i th random variable ($j=1..n$)

$$\mathbf{h}^i = \mathbf{u}^i - \alpha \mathbf{u}^i$$

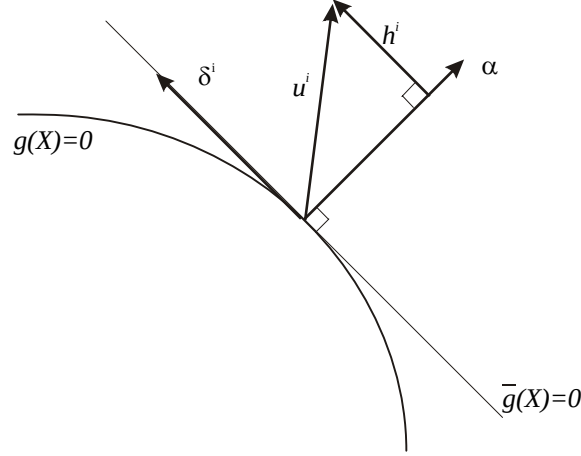


Figure 4.6: Vector projection technique used by Kim and Na [Kim 1997].

where $\alpha = \frac{\nabla g}{|\nabla g|}$

$$\nabla g = \left[\frac{\partial \bar{g}}{\partial x_1} \dots \frac{\partial \bar{g}}{\partial x_n} \right] \text{ evaluated at } \mathbf{x}_D$$

$$\mathbf{u}^{iT} = [e_1 \dots e_n] \begin{cases} e_k = 1 & \text{for } k = i \\ e_k = 0 & \text{otherwise} \end{cases}$$

- The first two steps are repeated until a convergence criterion on the reliability index is satisfied which is $|\beta_k - \beta_{k-1}| < \varepsilon_\beta$, where ε_β is a predetermined convergence criterion.

The authors also contend that most limit states are not highly non-linear around the design point. Therefore, the proposed method evaluates the reliability index with two different values of f_i and the results are compared to find the non-linearity index Δ using

$$\Delta = \frac{|\beta_{1.0} - \beta_{1.5}|}{n-1} \quad (4.14)$$

where $\beta_{1.0}$ is the reliability evaluation with $f = 1$, $\beta_{1.5}$ is the reliability index evaluated by the value of $f = 1.5$, and n shows the number of random variables. The non-linearity index is used to determine the value of f for the linear RSF by which the region of interest will be covered.

4.3 Modelling Error Assessment and Model Choice for a RSF

One of the most important deficiencies of the existing RSMs for reliability analysis is that they do not include any method for the evaluation of the modelling error for the formed RSF. The only assessment given by Rajashekhar and Ellingwood [Rajashekhar 1993] is carried out after calling the reliability analysis and comparing the reliability index with its previous value obtained in the first stage of structural reliability analysis. This approach makes comparison based on the reliability analysis carried out rather than evaluating the RSF itself. Other RSMs simply assume that approximating the performance function with a second degree polynomial gives a satisfactory result for the subsequent analysis or simply carrying out RSM in two stages will guarantee that the formed RSF will have a good approximation to the actual performance function. Using a modelling error assessment not only gives information on the RSF to be used in reliability analysis but also enables the iterative procedure of the RSM to be stopped at an early stage in the event of a satisfactory result in terms of having a good approximation to the performance function. In addition, using a modelling error assessment also helps in determining redundant term(s) that increase the modelling error, and result in a poor approximation to the actual performance function [Myers 1995].

Several methods have been developed for the conventional RSM either to choose the best terms for the RSF or to remove redundant terms from the RSF. The most widely used methods can be given as all-possible regression and stepwise regression methods [Kleinbaum 1987] [Otto 1993]. In the all-possible regression method, all possible combinations of terms that can be obtained from the design matrix are generated, and the best subset of terms is chosen for the RSF, based on one of the statistical criteria such as C_p , the t-statistics or PRESS, for which detailed information can be found in most statistical textbooks [Ryan 1997]. This method requires forming of 2^p partial response surface approximations for p terms in the general response surface approximation, which can increase the computational burden as the number of the terms gets large.

An alternative approach to the all-possible regression method is the stepwise regression method, in which one term is added or removed at a time based on the

evaluation of the same statistical criteria used in the all-possible regression. This method can be classified as a forward, a backward, or a mixed stepwise regression method. The forward stepwise regression method assumes a RSF with only one parameter, the intercept. Parameters are then added one at a time by selecting the parameter that results in the largest increase in predictive capabilities of the existing RSF. The process of adding parameters is repeated until the criterion chosen for termination of the process is satisfied. However, the backward stepwise regression method starts with a RSF, called the full model, that includes all the parameters that can be formed from the design matrix. One parameter from the full RSF is deleted at a time, selecting the parameter with the minimum influence on the predictive capabilities of the existing RSF. In the mixed stepwise regression method, a combination of forward and backward stepwise regression is utilised, which can be summarised as follows:

1. Starting with the full RSF as the initial RSF.
2. Apply the backward stepwise regression method, hence the least significant parameters based on the criterion are deleted from the RSF.
3. Apply the forward stepwise regression method to all of the deleted parameters to evaluate if any of the deleted parameters has a significant influence on the goodness of the RSF.
4. Repeat from step 2 and until the criterion is satisfied.

Although there are many examples of the application of stepwise regression methods in the literature [Venter 1996] [Unal 1994], there are some limitations that might prevent its use in some engineering applications. First of all, none of the stepwise methods guarantees to form the best possible RSF because the methods may exit prematurely from the procedure as happens mostly in the forward stepwise regression method. The second and most important deficiency of the stepwise methods, especially for computer-based experiments such as applications in structural reliability analysis, is the lack of statistical information on the response because there is not any variation in the response of the performance function as the same result is obtained for the same input from the analysis carried out on computers. Therefore, the statistical criteria such as p-values of t-test or F-test used to choose the best term(s) in stepwise regression methods cannot be used for the RSM in reliability analysis [Sacks 1989]

[Simpson 1997]. However, there are a number of studies utilising the same statistical criteria for deterministic computer experiments [Unal 1994] [Venter 1998] [Kodiyalam 1998]. In ADAPRES, the statistical criteria suggested for computer based experiments [Simpson 1998] are explained in Chapter 5.

4.4 Review of Issues in Response Surface Methods

As was outlined in the previous sections of this chapter, all the RSMs utilise the first proposed Bucher RSM in one way or another. Therefore, in this section as well as in later sections of this chapter, the term Bucher RSM will be used to indicate RSMs that are generated from Bucher's proposed method, including the RSMs explained in the previous sections of this chapter.

The main deficiency of the Bucher RSM is that the experimental design method is a saturated design, the definition for which is the number of experimental points is equal to the number of unknown coefficients in the RSF. Since there are zero residual degrees of freedom for this kind of design, it is impossible to test model adequacy without having additional experimental points. Therefore, a saturated design is only advised if there is a great limitation on calling the actual function to get extra experimental point [Myers 1995]. As will be shown in the application of ADAPRES, for some reliability problems using modelling error evaluation can help forming a very good RSF at the first stage without carrying out further reliability analysis.

The second deficiency concerns the type of RSF suggested in the Bucher RSM which is either a quadratic or a quadratic with mixed term functions. However, for some reliability problems, using a quadratic RSF might include an unnecessary term(s) which can reduce the accuracy of the RSM as well as increasing the computation time and cost of analysis. Conversely, in other problems a quadratic formulation may be inadequate to model the problem with sufficient accuracy.

The third deficiency of the method is related to using an experimental design comprised of a fixed number of experiments to form the RSF for each step of the RSM by assuming that the RSF formed in that step is sufficient to use either in the first stage of the RSM or the subsequent step(s). The logic behind this approach goes back to the RSM applications in real physical experiments for which setting up a new experiment always requires effort and time. In computer based experiments, however,

carrying out another experiment to improve the design matrix from which a better RSF may be obtained does not need preparation other than an extra call to the analysis from which the output of the performance function is obtained. Although it increases the computation time, for some problems having highly non-linear performance functions, it can reduce the modelling error caused by using a low order polynomial that cannot approximate the real performance function well.

All the deficiencies given above are considered in the development of the proposed method, ADAPRES, for which a detailed explanation will be given in the next chapter.

4.5 Neural Networks

Artificial neural networks, often called neural networks for short are computer systems linking inputs with outputs in a network structure of nodes and arcs (*connections*). They are inspired by replicating portions of what is known about the way human brains function. In the human brain, neurons are connected in a complex network, with activity generated by impulses from one neuron to another [Sundararajan 1994]. Artificial neural networks offer an important alternative to traditional methods of data analysis and modelling. In conventional computing a model is expressed as a series of equations, but a neural network is much more flexible. Instead of having a precise definition of a relationship or model the neural network is trained by the repetitive use of sample data called a training set to represent the best relationships and processes that are implicit within the data. The basic structure consists of a number of simple processing elements (also known as neurons or nodes), which are arranged in a number of different layers, and joined together to form a network as shown in Figure 4.7 . The processing elements sum their inputs, and then transmit a single output to all processing elements in the next layer via the connections. In Figure 4.7, for example, data enters the network on the left and is then fed forward through successive layers to emerge on the right. This is called a feed-forward network because the flow of information is all in one direction going from input nodes (left) to output nodes (right). The outer layer where information is presented to the network is called the input layer. The layer on the far side, where processed information is retrieved, is called the output layer. All layers between the two outer ones are called hidden layers.

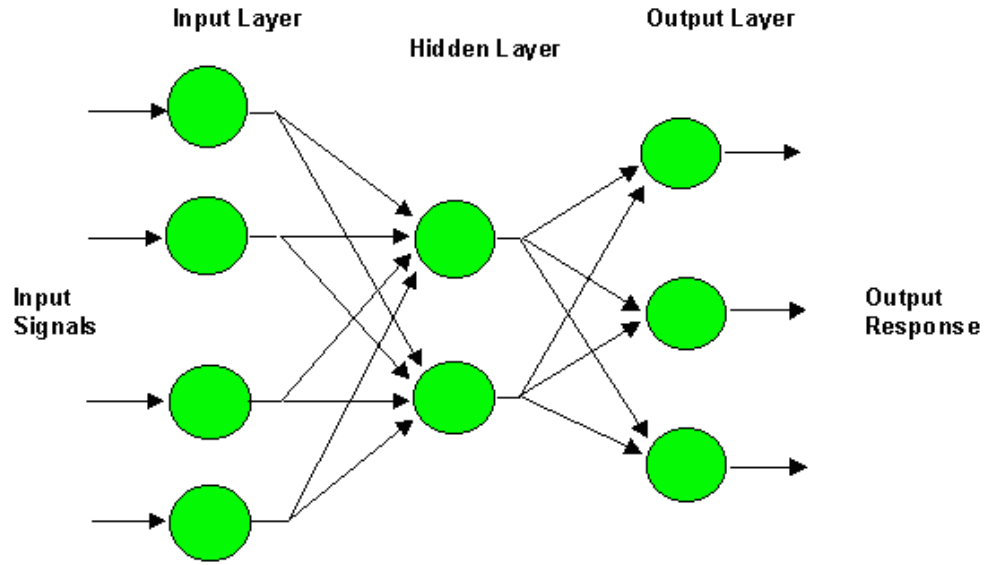


Figure 4.7: Basic configuration of a feedforward multi-layer neural network.

A brief mathematical description of a general neural network is given as follows [Papadrakakis 1996]:

Let input signals are represented as $I[1], I[2], \dots, I[N_{\text{inp}}]$ with weights $W[1,j], W[1,j], \dots, W[N_{\text{inp}}, j]$, where j is the identifier of a neuron, N_{inp} is the total number of the input signals. A signal within the neuron is obtained by the weighted sum combination rule as:

$$X[j] = \sum_{i=1}^{N_{\text{inp}}} I[i] W[i, j]$$

The sigmoid transfer function produces an output signal from the neuron as:

$$O[j] = 1 / [1 + \exp(-X[j])]$$

Then, a simple neural network including only one hidden (middle) layer is represented as follows:

The neurons in the middle layer ($j = 1$ to N_{mid}) are represented by

$$X_{\text{mid}}[j; n] = \sum_{i=1}^{N_{\text{inp}}} I[i; n] W_{\text{inp}}[i, j]$$

in which n refers to the training set and ranges from 1 to N_{trg} . The scaled input

from the middle layer is given by

$$M[j; n] = 1/[1 + \exp(-X_{mid}[j; n])]$$

The neurons in the output layer ($k=1$ to N_{out}) is represented as:

$$X_{out}[k; n] = \sum_{j=1}^{N_{mid}} M[j; n] W_{out}[j, k]$$

Then the output is given by

$$O[k; n] = 1/[1 + \exp(-X_{out}[k; n])]$$

In the learning process the total error over the whole training set is minimised by adjusting the various weights in the neural network. Initially all weights are chosen at random within the range -1 to $+1$. The inputs from the first member of the training set, $n=1$, are run through the network and the outputs $O[k; n]$ are obtained as described above. Comparison with the corresponding target $T[k; n]$, which are the equivalent values in the input training set, is carried out to obtain the error associated with the initial set of weights. The root-mean square error is computed over all outputs and all members of training set as

$$\text{Error} = \left[\sum_{n=1}^{N_{trg}} \sum_{k=1}^{N_{out}} \frac{(T[k; n] - O[k; n])^2}{N_{out} N_{trg}} \right] \quad (4.15)$$

The weights of the paths joining the nodes of the network are then adjusted by an algorithmic scheme to minimise the error given in Eq. (4.15). There are a number of schemes available, of which perhaps the best known is the back propagation approach developed by Rumelhart *et al.* [Rumelhart 1986]. This approach is invoked at the end of each forward pass through the network. For each training example n the algorithm uses the outputs $O[k; n]$ to produce the corrections the various weight. After each training set has been completed, the corrections are summarised and the weights adjusted before the next forward pass. After training has stopped, input data are then passed through the network in its non-training mode, where the data are transformed within the hidden layers to provide an appropriate output value or set of output values.

The use of neural network to predict analysis outputs has been studied in the context of optimal design of structural systems [Berke 1993] [Lu 1999]. Papadrakakis *et al.* [Papadrakakis 1996] applied neural networks to the reliability analysis of

complex structural systems in connection with Monte Carlo simulation. However, neural networks have some difficulties that might make their use inefficient [Lee 2001]. First of all, it is difficult to establish the architecture of the network as indicated by the number of hidden layers of neurons, the number of the neurons in each layer and the number of training patterns that must be presented to train the network. A process of trial and error is used to train the network, which is likely to perform inadequately for the problems where the behaviour of the functions varies significantly from one region to another. This results in either poor performance or the need for very large numbers of training samples.

4.6 Conclusion

In this chapter, a general review of the RSMs has been given, along with the more specific use of the RSM for structural reliability problems. The first section has contained the explanation of the conventional RSM, which is classified into three steps comprising DoE, model choice and model error evaluation. In the second section, the theoretical background of the conventional RSM has been explained, with the method used to compute the regression coefficients. As the RSM differs according to the applications in which it is used, a specific application of the RSM for the structural reliability analysis has been explored by giving detailed explanations on the most widely used methods in this domain. Modelling error assessment and model choice for a RSF were also given in this chapter. The deficiencies of these methods have been investigated, which give rise to the basics of the proposed RSM, called ADAPRES, which, is explained in detail in the next chapter, along with examples showing the comparison with the other methods mentioned in this chapter. Finally, another system identification method, neural network, has been investigated in this chapter.

Chapter 5

The Adaptive Response Surface Method

In this chapter, an alternative RSM is developed to overcome the deficiencies of the RSM for structural reliability analysis. This new method is called the Adaptive Response Surface method (ADAPRES). The first section of the chapter includes a basic explanation of the main steps of ADAPRES, divided into three stages, which are firstly the Design of Experiment, secondly the statistical criteria used in ADAPRES, and thirdly generating the terms in a RSF. The reliability methods used in ADAPRES are explained in the second section along with an improved approach to obtain the design point from Monte Carlo simulation. The third section contains a description of the method proposed in this study by which fewer iterations need be carried out than are required in traditional approaches to RSM. Finally, three examples are chosen to demonstrate both the applicability and the efficiency of ADAPRES.

5.1 Basic steps of ADAPRES

In the present work, methods for overcoming the limitations of the conventional RSMs, explained in Chapter 4, are proposed by developing a better RSM than the existing methods. For this purpose, an adaptive response surface method, called ADAPRES for short, has been proposed for which the basic steps are given in Figure 5.1. From this figure, it can be seen that the ADAPRES method can be broadly classified into three main stages.

The first ADAPRES stage, called the *initial stage* in Figure 5.1, involves forming an initial design matrix obtained from a design of experiment method (DoE) developed in this research. This stage also forms a matrix, called the response matrix, that contains the responses of the performance function, based on the experimental points in the design matrix. The final step of this stage is to form a linear RSF including the effect of all the random variables.

In the second ADAPRES stage, the linear RSF formed in the previous stage is improved by adding some extra term(s) if the linear RSF does not satisfy a performance criterion as will be explained in a later section of this chapter. If the improved RSF satisfies the criterion, it is used to compute the reliability data such as the design point and reliability index. Otherwise, extra experimental points are added to the design matrix, by which approach further higher terms can be added to the RSF. It also allows more information to be available about the design domain of the problem since more points are used to explore the design domain.

The final stage of ADAPRES, called *reliability assessment* in Figure 5.1, includes the assessment of the formed RSF according to the percentage difference between the performance function calculated for the mean values of the random variable and that calculated for the design point obtained by means of the formed RSF. The purpose of this criterion is to determine how close the response of the performance function is to the limit state where $g(\mathbf{X}) = 0$.

Although more detailed explanations of the steps of ADAPRES are given in the later sections of this chapter, some of the contributions of ADAPRES to the application of RSM in structural reliability analysis can be summarised as follows:

Generating the design matrix dynamically:

Instead of using a fixed DoE, a design matrix is formed by selecting the experimental points one at a time, based on their effect on the performance function.

Forming the design matrix with points from the region of interest:

The effect of random variables can be considered to determine which side of the distribution function is to be used to generate the experimental points, by which approach the selected points can be closer to the limit state.

Dynamic forming of the RSF:

After forming an initial RSF in a linear form, some higher term(s) are selected based on the modelling error assessment which enables a dynamic RSF that is determined according to the problem rather than having a fixed type of RSF for all kinds of structural reliability problems.

Introducing modelling error for RSF:

The selection of the best possible RSF by adding higher term(s) to the RSF can be achieved by utilising a modelling error assessment that indicates the importance of the added term(s) to the performance function.

Reducing the stages of the RSM:

For some problems having a linear response from the performance function, using a linear RSF in the initial stage of ADAPRES can result in the best possible RSF, so there is no need to carry out further iterations since the RSF represents the actual performance function sufficiently well.

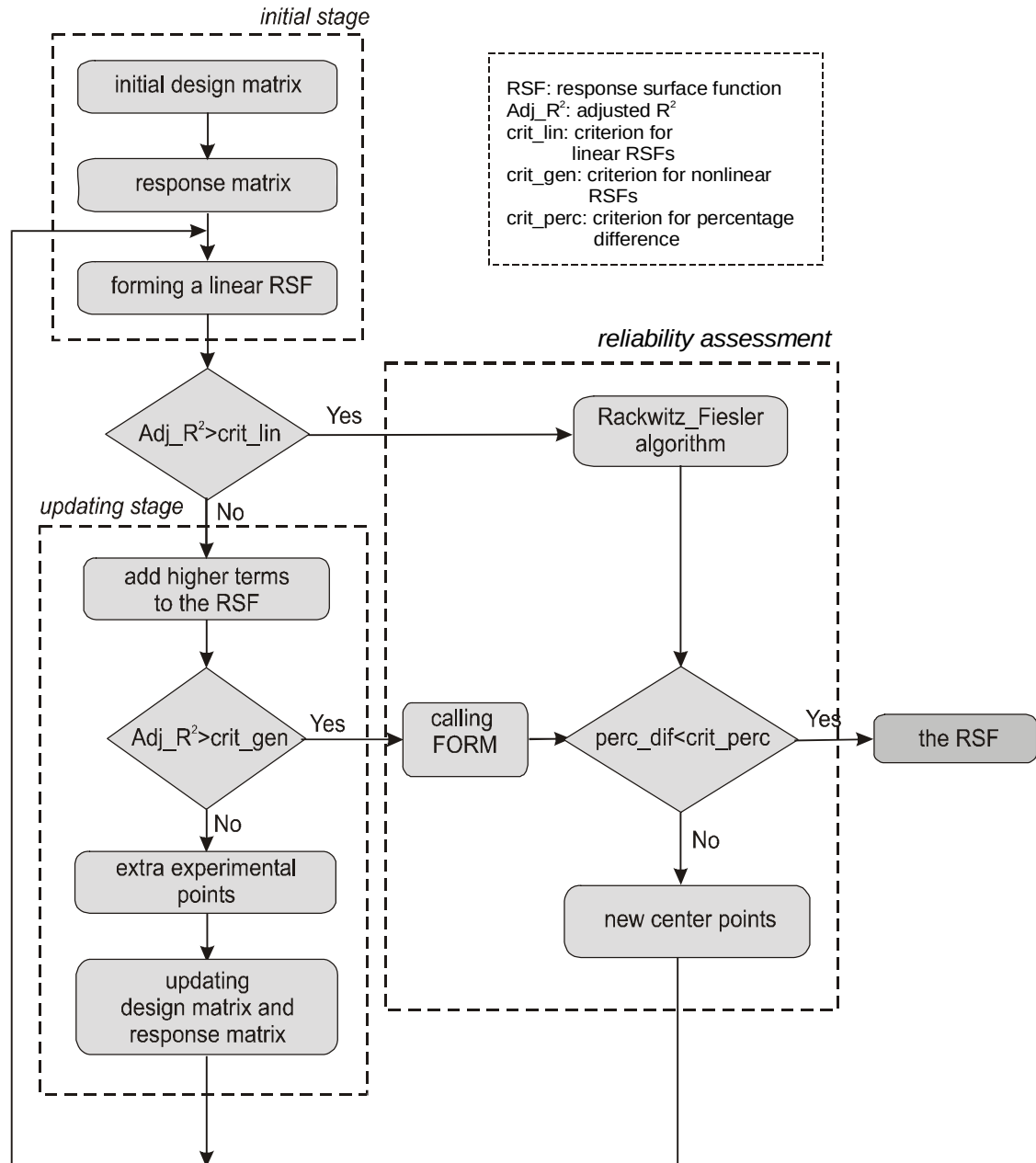


Figure 5.1: The basic steps of ADAPRES.

All the points given above are explained in the following sections in detail as well as comparing ADAPRES with the other RSMs. In addition, two experiments showing the improvement that results from use of a better RSF are also presented at the end of the chapter. The main steps are explained in the following sections, based on the general steps for a traditional RSM, which are experimental design, model choice and model error evaluation.

5.1.1 Design of Experiments (DoE)

Design of experiments (DoE) is used to select the value of the sampling points by which the design domain for structural reliability analysis is determined. In this design domain, structural reliability analysis is carried out to find the reliability data such as the design point, the probability of failure and the reliability index. Thus, it is very important to determine the design domain by means of the DoE employed. A general approach is to determine this domain by means of the mean value and standard deviation of the random variables. Bucher [Bucher 1989] proposed this approach using the following formula.

$$x_i = \bar{x}_i \pm f_i \sigma_i \quad (5.1)$$

where $\bar{x}_i$ and σ_i are the mean values and standard deviations of random variables, respectively. A more detailed explanation of the arbitrary function f_i in Eq.(5.1) was given in Chapter 4. By using Eq. (5.1), experimental points are selected from the design domain for which the upper and the lower boundaries are assigned as $+f_i \sigma_i$ and $-f_i \sigma_i$, respectively. When normally distributed random variables are considered, using Eq. (5.1) generates experimental points entirely around the mean value of random variables, which are also consistent with the distribution of the random variables since almost 68% of data is distributed between $\mu+\sigma$ and $\mu-\sigma$. Although Bucher's design might be convenient for this purpose, there is no criterion on how to select the arbitrary coefficient f_i in Eq. (5.1) that not only determines the location of the experimental points, but also indicates the minimum and maximum boundaries of the design domain. In the literature, various values are suggested, ranging from 1 to 3 [Bucher 1990] [Rajashekhar 1993] [Kim 1997]. Although some researchers use the same value for f_i for all iterations in the RSM, Rajashekhar and Ellingwood [Rajashekhar 1993] proposed to reduce the value of f_i in the later iterations which enables the RSM to search for the design point in a smaller design domain. Consequently, fewer iterations might be needed to be carried out to find the required reliability data. However, there is no clear definition on how to reduce the value of f_i for the later stages in the RSM.

The DoE employed in this study focuses mainly on two parameters of the design matrix, one of which is the determination of the value of the arbitrary function f_i by

which the boundaries of the design domain are determined, and the other is the selection of the experimental points from this design domain by considering the fractile of the distribution function of the random variable, based on the effect of the random variable on the performance function.

5.1.2 Determining the Value of the Arbitrary Function

The value of the arbitrary function f_i in ADAPRES can be chosen to be any value since the value is read from a data file described in Section 5.4, allowing the user to select or modify the boundaries of the design domain more flexibly. However, from the experiments carried out in this study, the value of f_i for the initial design is determined by considering the *choice of* reliability method in which the RSF is substituted for the performance function. In the case that FORM or SORM is used, the value of f_i can be reduced in subsequent iterations as these methods search only a single point, which is the design point, on the limit state. As long as the region where the design point exists is approximated well, the RSF obtained from that kind of design matrix gives a good result in terms of the POF, the reliability index for the subsequent stage of structural reliability analysis. If one of the Monte Carlo methods is used, the RSF should cover the sampling region of the simulation because the samples can be distributed over the entire design domain in order to have a good distribution of the samples, which is the one of the requirements of Monte Carlo simulation explained in Chapter 3. Therefore, the distribution of the random variables of the structural reliability problem becomes important to determine a good design domain. For normally distributed random variables, for instance, 99% of data are distributed between $\mu+2.58\sigma$ and $\mu-2.58\sigma$ as is illustrated in Figure 5.2. In order to form the initial design matrix by experimental points that span between minimum and maximum boundaries of the distribution function, an initial value for f_i of 2.5 can therefore be suggested for normally distributed random variables. Thus the design space from which the design matrix is formed is determined between -2.5σ and $+2.5\sigma$. The experimental points are selected from this design space. If second or further iterations are required to satisfy the criterion set for the RSF, however, the value for f_i is reduced to 1 in order to avoid the selection of experimental points from the region in which the POF very low. This also enables ADAPRES to reduce the design domain around the design point obtained in the preceding iteration, by which

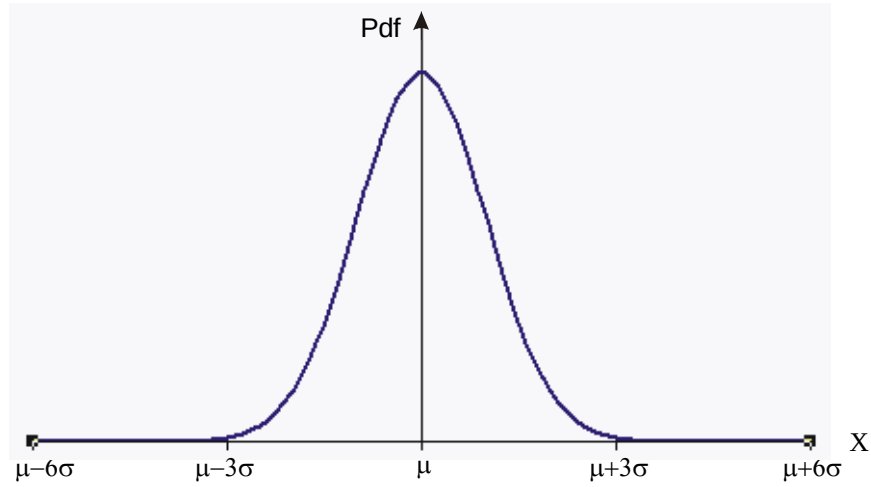


Figure 5.2: Illustration of normal distribution function.

more experimental points can be selected from the region in which the actual design point is likely to exist. Furthermore, reducing the value of the arbitrary coefficient also prevents the DoE from selecting experimental points outside of the boundaries of the distribution function of the random variables. For instance, if the first stage results in a design point in the lower section of the distribution function, the experimental points for the second stage will be placed around $\mu + 6\sigma_i$ or $\mu - 6\sigma_i$ if the same value for f_i is chosen for the second stage as is carried out using Bucher's design.

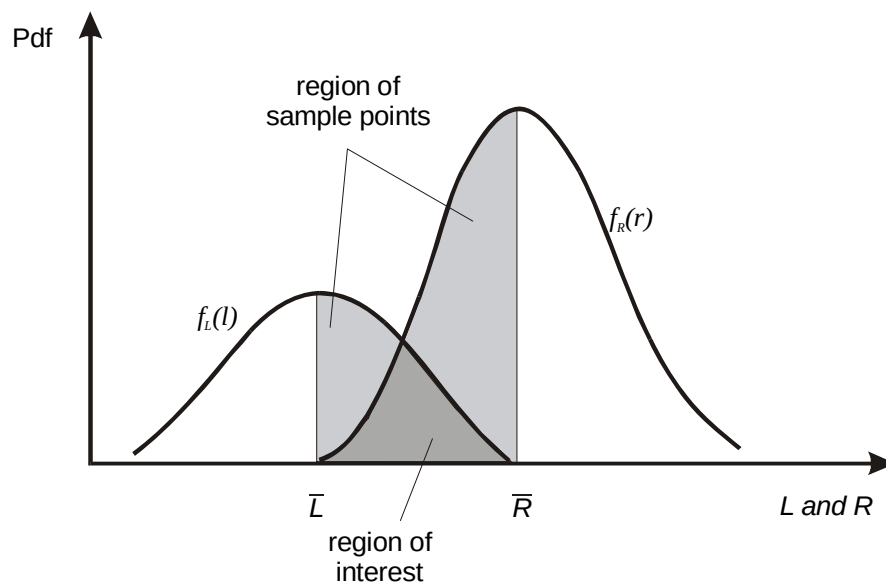


Figure 5.3: Distribution functions and the region of interest for L and R .

As can be seen from Figure 5.2, this will yield to selection of the experimental points from a region where the POF is less than 1×10^{-10} which can only be taken into account for extreme probability analysis. It is worth restating that the value of the arbitrary function is input as a parameter in ADAPRES so that the user can select any value, based on his/her experience.

5.1.3 The Effect of The Random Variables on The Performance Function

In order to select the experimental points from the region of interest or at least close to that region, ADAPRES utilises an approach in which the response of the performance function for the experimental points is observed in order to check the effect of the random variables on the performance function. This can be illustrated by means of an example in which the performance function is formed by a load variable and a resistance variable as given in Eq. (5.2).

$$g(x) = R - L \quad (5.2)$$

It is obvious that the region of interest will be formed by the lower tail of R and the upper tail of L as shown in Figure 5.3. On the other hand, it should be noted that the shaded area is not equal to the POF, which only shows the qualitative measure of the POF. Instead of taking into account the whole distribution area to form the design matrix, experimental points can be chosen from one side of the distribution to cover the region of interest since it includes the design point. A similar approach was proposed by Rajashekhar and Ellingwood [Rajashekhar 1993] where the experimental points are selected from the upper or lower fractile of the distribution function according to their effect on the performance function. However, this approach may give a very good estimate only if the design point falls in the very lower or upper tail of the distribution since the experimental points are selected from that region. On the other hand, it may give an erroneous result for problems whose design points are close to mean values since this region is not well covered by the experimental points chosen. For an explicit performance function as given in Eq.(5.2), the effect of the individual random variable can be easily determined by studying the performance function. For instance, one can easily say that the experimental points for the performance function given in Eq. (5.2) must be selected from the lower tail of the resistance variable while the upper tail is used to select the experimental points for the

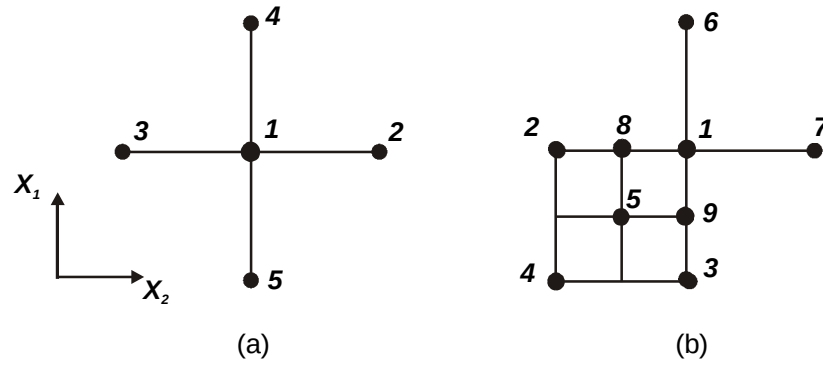


Figure 5.4: Bucher's experimental points (a), and ADAPRES' experimental points (b).

load variables. However, for an implicit performance function in which the random variable is not explicitly expressed, the effect of the variables can be determined by carrying out some analysis such as sensitivity analysis. In ADAPRES, however, a simple approach is used to determine the effect of the random variables, which is based on the observation of the response for the corresponding experimental point at the time of forming the design matrix. In order to explain the method employed by ADAPRES for the selection, the experimental points are illustrated in Figure 5.4, in which the numbers show the selection order for the experimental points. The value of the arbitrary function f_i is determined according to the approach given in the previous section. The selection procedure of the experimental points in ADAPRES is as follows:

1. A first experimental point at the mean value of the random variables, marked **1** in Figure 5.4(a), is selected unless there is *a priori* information on the problem.
2. For the second experimental point, ADAPRES selects the point that is formed by $x_i = \bar{x}_i - f_i \sigma_i$, corresponding the experimental point **2** in Figure 5.4.
3. If the absolute value of the response from the performance function for the experimental point **2** is lower than that of the experimental point **1**, this point is included in design matrix in ADAPRES while the other random variables are kept at their mean values.
4. Otherwise, the experimental point **7**, which is formed by $x_i = \bar{x}_i + f_i \sigma_i$ is chosen.

5. The same approach is utilised to select the experimental points for the rest of the random variables of the structural reliability problem, illustrated in Figure 5.4(b).
6. For the fifth experimental point in the figure, ADAPRES simply computes the mid point between the end points (2 and 4 in Figure 5.4) and the mean values.

The reason for searching the absolute lowest value of the experimental point for the random variable in step 3 is that the experimental points should be selected close to the limits state where $g(\mathbf{x}) = 0$. It should be noted that this selection only works for problems whose performance function behaves globally monotonically which means that the effect of the random variable on the performance function is monotonic overall. As will be shown in Example 1, the design matrix formed in ADAPRES results in a better RSF than the design matrix formed according to Bucher's design.

5.1.3.1 *Generating the Design Matrix Dynamically*

The number of the experiments in both deterministic applications of RSM and in reliability analysis is conventionally held constant. This approach has been mostly used successfully in physical experiments in which a RSM is used after finishing the experiments designed by using one of the DoE methods such as central composite design. Setting up another experiment for additional experimental points also requires time and physical effort. However, computer based experiments are quite different in terms of getting the output from the experiments for additional experimental points, which only increases the time spent to form the response matrix. In order to accomplish the first objective of the thesis, which is to find out the best possible RSF for the problem, a dynamically generated design matrix is used rather than a fixed design matrix. In a general RSM, once the design matrix is formed, it is used to find out the RSF. For some cases, the design matrix may not be sufficient to form a RSF that can be used either in the first stage or in the reliability analysis. Instead, ADAPRES forms a design matrix for a linear RSF, based on the approach given in the previous section. Then, it is checked if the RSF formed is satisfactory for the use in reliability analysis. The necessary criterion for this check is based on modelling error assessment, and is explained in the following sections of this chapter.

If the existing design matrix cannot result in an acceptable RSF to be used in

reliability analysis, the next issue is to select some additional points, given in Figure 5.1 as the *updating stage*. The criterion used in ADAPRES for the selection is chosen on the basis of the effect of random variables on a performance function whether the additional point should be selected from the right tail or left tail of the distribution function, based on the comparison of the results for the experimental points, which is explained in the previous section. This selection results in having the experimental points close to the limit state. For the place of the additional points for a random variable, the mid point between mean value and left or right boundary of the distribution function is chosen first, illustrated in Figure 5.4 as the points marked **8** and **9**, and for the later selections, the tail of the distribution function is simply divided evenly to get the necessary experimental points. As is shown in Example 3, adding two more experimental points to the design matrix increases the goodness of the formed RSF substantially. This procedure is carried out until the design matrix is able to satisfy the condition set for the RSF.

5.1.4 The Statistical Criteria Used in ADAPRES

ADAPRES makes use of a modelling error assessment based on the *Adjusted Coefficient of Determination* (Adj_R^2) that represents the proportion of the variability in the response data that is accounted for by the response surface approximation. After carrying out regression analysis, R^2 is computed as follows:

$$R^2 = \frac{SS_R}{S_{yy}} = 1 - \frac{SS_E}{S_{yy}} \quad (5.3)$$

in which SS_R denotes the regression sum of squares, SS_E the error or residual sum of squares, and S_{yy} denotes the total sum of squares, for which more explicit forms of the expression are as follows:

$$SS_R = \mathbf{b}^T \mathbf{X}^T \mathbf{y} - \frac{\left(\sum_{i=1}^n y_i \right)^2}{n}$$

$$SS_E = \mathbf{y}^T \mathbf{y} - \mathbf{b}^T \mathbf{X}^T \mathbf{y}$$

$$S_{yy} = \mathbf{y}^T \mathbf{y} - \frac{\left(\sum_{i=1}^n y_i \right)^2}{n}$$

in which n is the number of experiments, $\mathbf{b}$ is the vector for the coefficients in the RSF, $\mathbf{X}$ is the design matrix including the experimental points, and $\mathbf{y}$ is the vector showing the results of the performance function for the experimental points in the design matrix.

The value of R^2 is between 0 and 1, and a large value of R^2 may indicate that the selected model can result in a good approximation to the performance function. However, this measurement is not independent on the number of terms in the RSF because adding a variable to the model will always increase R^2 , regardless of whether it is significant or not. In order to overcome this deficiency related to the measurement, the adjusted R^2 approach can be used whose definition is stated below:

$$Adj_R^2 = 1 - \frac{SS_E / (n - p)}{S_{yy} / (n - 1)} = 1 - \left(\frac{n - 1}{n - p} \right) (1 - R^2) \quad (5.4)$$

where p shows the number of parameters in the RSF. Adj_R^2 indicates the goodness of the fit of the selected model to the performance function. A value of $Adj_R^2 = 1$ means that the RSF passes through every experimental point in the design matrix while a value of $Adj_R^2 = 0$ means that the RSF does not represent the experimental points any better than a horizontal line. Therefore, the required value for Adj_R^2 can be chosen between 0 and 1, depending on the trade off between the number of the calls to the performance function and the modelling error since a higher value of Adj_R^2 requires the RSF to pass very close the experimental points. A RSF having a higher value of Adj_R^2 means that the RSF is a good choice for the experimental point used to form that RSF. For practical purposes it is generally chosen very close to 1 such as 0.9 [Myers 1995]. In ADAPRES, the value of the criterion for a linear RSF, `crit_lin` and for a general RSF, `crit_gen` are parameters that can be set in the input data file by the user. From our experience, it can be suggested that `crit_lin` should be selected higher than that of a general RSF, `crit_gen`, in order to determine the linearity of the performance function correctly. This selection forces ADAPRES to have a linear RSF that can pass through almost all of the experimental points in the design matrix. Therefore, for the examples in this chapter, the value of `crit_lin` is selected as 0.95, by which the linear response of the performance function is well approximated by means of a linear RSF. In the case that the linear RSF is not adequate to approximate the performance function well enough to satisfy the criterion, ADAPRES starts adding

higher order term(s) to the linear RSF until the value of Adj_R^2 is equal or higher than $crit_gen$, based on the approach explained in the following section.

5.1.5 Generating the Terms in a RSF

In order to generate the terms of the RSF, the all-possible regression method is used to determine the best RSF according to its value of Adj_R^2 since the number of the variables in a structural reliability problem is generally low. To generate all the possible combinations of the terms from the design matrix, an algorithm was developed in this study, by which ADAPRES can dynamically generate terms for the RSF. Firstly, ADAPRES forms a linear RSF to check whether it is a sufficient model type to approximate the real performance function by checking the value of Adj_R^2 . The reason for starting with a linear RSF is that the limit state around the design point mostly behaves linearly for many structural reliability problems according to some researchers [Ditlevsen 1996] [Kim 1997]. Another advantage of this approach is that a linear RSF can approximate the response of the real performance function quite well if it behaves linearly. Thus, there is no need to carry out further experiments for the additional experimental points required for the RSF including higher terms such as a quadratic RSF. Therefore, the number of calls to the real performance function can be reduced. If the value of Adj_R^2 is less than $crit_lin$, i.e. a linear RSF is not satisfactory, then the next stage involves forming higher terms observing the criterion set by the maximum number of terms allowed for a RSF. This criterion is determined as one less than the number of experiments. This approach leaves at least one experimental point out to be used in modelling error evaluation. Although leaving one experimental point out from the forming process of the RSF might not be satisfactory for some problems, or may be the least condition to compute the modelling error, we note that the more points left out from the regression analysis, the more information can be obtained for modelling error evaluation. This requires use of a more advanced DoE including more experimental points than the traditional Bucher design method. In order to perform the approach explained above, a matrix called the indicator matrix is formed in ADAPRES, in which the combination of the terms is obtained by marking the corresponding cells as is given in Figure 5.5. The indicator matrix is arranged to have a distinct area for the terms having the same degree. The *least shaded* area in the matrix, for example, includes quadratic terms while *the darkest area* shown in Figure 5.5 can generate 6th

order terms for the RSF. Having checked the formed RSF against crit_lin , the first area, including only quadratic terms, is marked 1 to generate the following terms:

$$x_1x_1 \quad x_1x_2 \quad x_2x_2$$

The number of terms that is added to the RSF from the group as given above is determined according to the maximum number for the terms the RSF can have. If the maximum number for the terms is less than the number of the terms in the group, terms are added one by one, and the value of $\text{Adj_}R^2$ is compared for each added term. If the added terms increase the value of $\text{Adj_}R^2$ of the formed RSF, these terms are marked as 1 in the matrix while the terms decreasing the value of $\text{Adj_}R^2$ are assigned as -1 so ADAPRES does not add these terms to the RSF. This process is carried out until the total number of terms is equal to the maximum number for the term. If the search fails to form a RSF whose $\text{Adj_}R^2$ is more than the criterion set for a general RSF, called crit_gen in Figure 5.1, the RSF having the maximum $\text{Adj_}R^2$ is accepted to find out the effect of the individual random variables in order to add additional experimental points to the design matrix. By adding some experimental points, ADAPRES can form a higher degree RSF from which a better approximation may be obtained. In Figure 5.5, for example, the following type of RSF is obtained by searching the terms up to 4th order terms.

$$y = a + b_1x_1 + b_2x_2 + b_3x_1^2 + b_4x_2^2 + b_5x_1x_2^2$$

in which a and b_i are the coefficients of the RSF to be obtained. The advantages of using such a matrix can be summarised as follows:

Instead of inputting a fixed type of RSF in ADAPRES, requiring additional effort to test for other types of RSFs, this matrix enables ADAPRES to add polynomial term(s)

	x_1	x_2	x_1^2	x_2^2	x_1^3	x_2^3
x_1	1	-1	0	1	0	0
x_2	0	1	0	0	0	0
x_1^2	0	0	0	0	0	0
x_2^2	0	0	0	0	0	0
x_1^3	0	0	0	0	0	0
x_2^3	0	0	0	0	0	0

	Quadratic terms
	Cubic terms
	4 th order
	5 th order
	6 th order

Figure 5.5: Indicator matrix for two random variables.

dynamically, depending on both the modelling error assessment, and the maximum number for the terms.

As the RSF formed in ADAPRES is used in place of the real performance function in other programs or applications, it becomes very easy to regenerate the RSF since the matrix clearly indicates the terms to be added to the RSF while the value of the coefficients is kept in a file. Therefore, the applications or the programs just need to access the data file and the matrix

As the RSM considered in this study is mainly used for structural reliability analysis, ADAPRES includes some features from the structural reliability methods in order to generate a RSF for structural reliability problems. Therefore, the following section describes the use of ADAPRES in structural reliability methods.

5.2 Reliability Assessment in ADARES

One of the main differences between a conventional RSM used for optimisation problems and a RSM formed for structural reliability analysis is that the latter can be established in a restricted region in the design domain by means of a reliability assessment, while the former should simply cover all the design region. It is reasonable for the conventional RSM to distribute the sampling points in entire design domain unless prior information on the whereabouts of the optimum design values is available. Therefore, all the RSMs in structural reliability analysis use one of the structural reliability methods to reduce the design domain to a smaller region.

ADAPRES utilises two different structural reliability methods, based on the type of RSF generated in ADAPRES. In the case that ADAPRES yields a linear RSF either in the initial stage or in the updating stage, the Rackwitz-Fiesler [Rackwitz 1978] algorithm is employed to get the reliability index since the exact value of the reliability index is obtained from the closed form solution. A more detailed explanation of this method is given in Chapter 3. Although the reliability index can be used to compute the probability of failure, the Rackwitz-Fiesler algorithm does not provide the value of the design point that is needed for both the modelling error evaluation and computation of the new centre point in ADAPRES. Therefore, ADAPRES utilises the following optimisation expression to obtain the design point yielding the reliability index obtained from the Rackwitz-Fiesler algorithm:

$$\begin{aligned}
& \text{minimise } F(\mathbf{U}) \\
& \text{subject to} \\
& \quad g(\mathbf{X}) \leq 0 \\
& \quad \beta_{RF} - \sqrt{\mathbf{U}^T \mathbf{U}} = 0
\end{aligned} \tag{5.5}$$

in which $\mathbf{X}$ shows the random variables in normal space, $g(\mathbf{X})$ is the performance function, $\mathbf{U}$ indicates the reduced random variables which is obtained by the following expression

$$\mathbf{U} = \frac{\mathbf{X} - \mu_X}{\sigma_X}$$

where μ_X and σ_X are the mean value and standard deviation of $\mathbf{X}$. β_{RF} is the reliability index obtained from the closed-form solution of the Rackwitz-Fiesler algorithm. Other parameters in Eq. (5.5) are explained in detail in Chapter 3.

For a nonlinear RSF, two different approaches can be chosen in ADAPRES, one of which is based on using the FORM, and the other is to use one of the Monte Carlo methods combined with an optimisation method as is explained in the following section.

5.2.1 The FORM in ADADPRES

The reliability index and the value of the design point can be obtained by defining the problem as an optimisation problem as follows

$$\begin{aligned}
& \text{minimise } \beta = \sqrt{\mathbf{U}^T \mathbf{U}} \\
& \text{subject to} \\
& \quad g(\mathbf{X}) = 0
\end{aligned} \tag{5.6}$$

for which a brief description of the parameters for the above equation is given for Eq.(5.5) in the previous section, and a more detailed explanation is given in Chapter 3. Having defined this optimisation problem, any of the standard optimisation methods can be used to find the optimum values. However, in this study, ADAPRES is combined with an optimisation package, called CFSQP [CFSQP 1998] for which more detail was given in Chapter 2.

To use the FORM in CFSQP, a file is prepared that includes user-defined version of three functions relevant to the problem concerned. In the first function, besides the

program parameters, upper and lower bounds of the random variables are also provided as well as an initial guess for the value of the random variables. If these initial values are not feasible, CFSQP is able to generate feasible values to start optimisation iteration. As initial values for the random variables, ADAPRES provides mean values of the variables. In the second function, the objective function is given in transformed $\mathbf{U}$ -space as shown in Eq. (5.6). Finally, the third function includes the expression of the performance function.

5.2.2 MCS in ADAPRES

As mentioned in Chapter 3, using the FORM can generally give accurate results. For some cases, however, it can suffer a convergence problem due to the methods used in the FORM. Therefore, in ADAPRES, a Monte Carlo method that is modified to get the design point is utilised. By using crude Monte Carlo simulation, only the probability of failure can be obtained, without having the points of maximum likelihood, which is called the design point in the FORM/SORM approach. However, the design point is needed not only to check the RSF obtained to examine whether it is good enough to be used in place of the actual performance function, but also to obtain the new centre points by which a new design matrix is formed. Therefore, the following optimisation problem, which is similar to the expression given in Eq.(5.5), is used to get the value of the design point corresponding to the probability of failure obtained by Monte Carlo simulation.

$$\begin{aligned}
 &\text{minimise } F(\mathbf{U}) \\
 &\text{subject to} \\
 &\quad g(\mathbf{X}) = 0 \\
 &\quad \beta_{equiv} - \sqrt{\mathbf{U}^T \mathbf{U}} = 0
 \end{aligned} \tag{5.7}$$

in which $F(\mathbf{U})$ is the same objective function as in Eq. (5.6), $g(\mathbf{X})$ is the performance function in $\mathbf{X}$ -space, β_{equiv} is called the equivalent reliability index obtained from the following expression

$$\beta_{equiv} = -\Phi^{-1}(P_f) \tag{5.8}$$

where the probability of failure, P_f , is easily obtained from Monte Carlo simulation and $\Phi(\cdot)$ is the standardised normal distribution function. Eq. (5.7) yields a point on

the limit state, whose distance to the origin of the reduced space is equal to the equivalent reliability index obtained from Eq. (5.8). The point obtained from Eq. (5.7) cannot be the point having the minimum distance in the reduced space. This approach also alleviates the deficiencies of the FORM/SORM for highly nonlinear structural reliability problems, for which the theoretical review is given in Chapter 3.

The second constraint in Eq.(5.7) requires the computation of the inverse of the cumulative distribution function in Eq.(5.8). In ADAPRES, a statistical program, **dcdflib** from [decdfliib 1997], is used to obtain the inverse, which is a collection of C program functions, including algorithms for major probability distribution functions such as Gamma, Normal or Chi-square. These C functions are modified in such a way that they can be called inside ADAPRES without requiring a separate call to the main function of dcdflib.

5.3 Reducing the Stages of Forming RSF

Almost all of the RSM methods in the literature assume that forming a RSF for reliability analysis should be carried out at least in two stages. The first stage involves forming a general RSF from the experimental points centred at the mean values of the random variables, and in the second stage the final RSF is formed from the experimental points based on the design point obtained in the first stage. However, for reliability problems whose performance functions are well approximated by using the initial design matrix, especially for linear performance functions [Kim 1997], only one stage may be enough to find out reliability information such as the reliability index, probability of failure and design point as is shown in Example 1 in Section 5.4.1. In order to stop the iterative procedure from carrying out the second stage, the following approach is used in ADAPRES:

Since the RSF is formed to approximate to a limit state where $g(\mathbf{x}) = 0$, the obtained RSF is checked to examine how close it is to the limit state, based on a comparison between the outcome of $g(\bar{\mathbf{x}})$ using mean values of the random variables, and $g(\mathbf{x}_D)$ using values at the design point. If $g(\mathbf{x}_D)$ is acceptably close to the limit state, ADAPRES does not go to the next stage, and reliability analysis is carried out by using the RSF previously obtained. The criterion set in ADAPRES can be given more explicitly as

$$\text{perc_dif} = 100 - 100 \frac{g(\bar{x}) - g(x_D)}{g(\bar{x})} \quad (5.9)$$

and the RSF is assumed to be very close to the limit state if perc_dif is less than the criterion that can be determined according to how accurately the reliability result is required. The small values indicate that the RSF is close the limit state. Therefore, the criterion for the examples given in this thesis is selected as being lower than 5%, based on the experiments carried out with different values of perc_dif. However, a proper selection could be done by carrying out more experiments with different values of the criterion, then using the RSF in structural reliability analysis to observe the variation of the reliability results. It should be noted that the smaller the value for perc_dif, the more iteration is needed

The basic explanation on how to use ADAPRES for a structural reliability analysis is presented in Appendix B. In addition, in order to show the benefit of using the ADAPRES method, two examples are described below, using problem data take from various reference papers. In the first example, ADAPRES resulted in a linear RSF, whereas a more complex RSF was required for the third example as shown in the following section.

5.4 Examples

The method proposed here is tested by applying it to examples chosen from two reference papers using data given for examples in the papers. For some examples, the results obtained from ADAPRES are compared with the results of the method based on Bucher's RSM. Exact results are obtained from either the data given in related reference paper or from the COMREL program [Comrel 1999] by using the real performance function in a reliability analysis.

5.4.1 Example 1

This example was taken from [Kim 1997], in which the problem parameters are as follows:

$$\text{Limit state function: } g(\mathbf{X}) = x_1 - x_2 / x_3 \quad (5.10)$$

where X_1 , X_2 and X_3 are normally distributed with $N(600,30)$, $N(1000,33)$ and

$N(2,0.1)$, respectively.

As can be seen from Eq. (5.10), the response of the performance function is linear, so ADAPRES yields the following linear RSF at the initial stage.

$$y = -556.0364 + 0.999x_1 - 0.539x_2 + 300.111x_3$$

Although the Rackwitz-Fiesler method provides a closed form solution to the reliability index, β_{RF} , an optimisation problem given in (5.5) must be utilised to find the value of design point resulting in the same reliability index. The exact result for the design point and the reliability index were obtained from COMREL [Comrel 1999] by using the exact performance function in Eq. (5.10) directly in the program. Also, the result for Bucher's design was calculated from ADAPRES by choosing the type of the RSM as Bucher, and then carrying out the FORM to get the value of the design point and the reliability index. The results from Kim's response surface method are also given in Table 5.1, the values in which are obtained from [Kim 1997]. In the reference, only the reliability index and the POF were given. Therefore, Table 5.1 includes the comparison of those results with the results of ADAPRES. As can be seen from Table 5.1, even though the Bucher RSM requires 15 calls to the performance function to form a quadratic RSF, the results from this RSF are not better than a simple linear RSF obtained from ADAPRES, since the response of the performance function is linear. The absolute percentage difference that is calculated from (5.9) is quite low. Therefore, the design points obtained from ADAPRES are very close to the actual design points even though the procedure stops at the first stage in which the first design matrix is used to obtain the RSF. Since Kim's response surface method was developed especially for linear type performance functions as described in Chapter 4, it yields slightly better results than that of ADAPRES. However, ADAPRES requires less calls to the performance function than that of Kim's response surface method, which is important for the application of RSM in an optimisation problem.

Table 5.1: Results from exact solution and from ADAPRES.

	<i>Exact</i>	<i>Bucher RSM</i>	<i>Kim RSM</i>	<i>ADAPRES</i>
No. of calls to $g(x)$	15		15	9
Design Points				
X_1	555.917	551.439		556.566
X_2	1028.80	1031.310		1028.282
X_3	1.853	1.874		1.856
Reliability Index	2.254	2.145	2.232	2.213
POF	0.0121	0.01597	0.01281	0.0134
Absolute perc_dif : 2.630%				

5.4.2 Example 2

The second example shows the comparison of the results obtained from Rajashekhar's response surface method and ADAPRES, for which the case problem was taken from [Rajashekhar 1993]:

$$\text{Limit state function: } g(\mathbf{X}) = 18.46 - 7.4769 \times 10^4 x_1 / x_2^3 \quad (5.11)$$

where X_1 , X_2 are normally distributed with $N(1000, 200)$, $N(250, 37.5)$ and $N(2, 0.1)$, respectively.

As given in Table 5.2, although the POF obtained from Rajashekhar's RSM is slightly better than the results of ADAPRES, ADAPRES gives better results for the design point and the reliability index. In addition, ADAPRES requires many fewer calls to the performance function than Rajashekhar's RSM, which shows the effectiveness

Table 5.2: Comparisons of the results from Rajashekhar's RSM with that of ADAPRES.

	<i>Exact</i>	<i>Rajashekhar RSM</i>	<i>ADAPRES</i>
No. of calls to $g(x)$		35	15
Design Points			
X_1	1093.4	1072.38	1120.38
X_2	165.0625	163.15	166.13
Reliability Index	2.341	2.234	2.315
POF	0.00957	0.00954	0.01035
Absolute perc_dif : 1.145%			

of ADAPRES in terms of computational time spent for structural reliability analysis for an optimisation problem as given in the case study in Chapter 7.

5.4.3 Example 3

In order to show the validity of ADAPRES for problems having more complex performance functions, an example from Wang and Grandhi's [Wang 1996] paper was taken, for which the performance function is given as:

$$\text{Limit state function: } g(\mathbf{x}) = x_1^3 + x_1^2 x_2 + x_2^3 - 18 \quad (5.12)$$

where X_1 and X_2 are distributed normally with $N(10, 5)$ and $N(9.9, 5)$, respectively. The problem parameters for ADAPRES are shown in Table 5.3. The first design matrix formed in both Bucher's RSM and ADAPRES is given in Table 5.4. As seen from the table, ADAPRES selects experimental points around the limit state while Bucher's RSM simply selects the experimental points around the mean value in the design domain, according to the approach given Eq. (5.1), some of which result in very poor approximation to the limit state such as the result for the experimental points of the third experiment. Such experimental points are avoided in ADAPRES by means of the comparison of the response of the performance function for the experimental points as is explained in Section 5.1.3. For ADAPRES, the first design matrix, formed by only five experimental points given in Table 5.4 was inadequate to form a RSF to satisfy the criteria given in Table 5.3 since the value of Adj_R^2 is less than the criteria set for

Table 5.3: Problem Parameters of Example 3 for ADAPRES.

<i>random variables</i>	X_1	X_2
<i>Mean values</i>	10	9.9
<i>Standard deviations</i>	5	5
<i>f_{min} and f_{max}</i>	2.55	2.5
<i>Type of design</i>	ADAPRES	
<i>crit_lin</i>	0.95	
<i>crit_gen</i>	0.9	
<i>Perc_dif</i>	1	
<i>Number of extra points</i>	1	
<i>Type of RSM</i>	ADAPRES	
<i>Method for design point</i>	FORM	

Table 5.4: Design Matrix of both RSMs.

<i>Bucher's Design Matrix</i>				<i>ADAPRES' Design Matrix</i>		
<i>No. of Experiment</i>	X_1	X_2	$g(X)$	X_1	X_2	$g(X)$
1	10	9.9	2942.298	10	9.9	2942.298
2	-5	9.9	1074.798	-2.5	9.9	998.548
3	25	9.9	22764.798	10	-2.6	704.423
4	10	-5.1	339.348	-2.5	-2.6	-67.451
5	10	24.9	18910.248	3.75	3.65	134.689
6				3.75	-2.6	-19.404
7				-2.5	3.65	37.814

both the linear RSF and the general RSF. Therefore, one additional experimental point for each random variable is selected at the mid-point between the limits, whose values are indicated in the shaded area of Table 5.4. The RSF created from this design matrix is a higher order polynomial as follows:

$$y = -204.474 - 9.4453x_1 + 5.9470x_2 + 8.584x_1^2 + 1.026x_2^2 + 13.449x_1x_2 \quad (5.13)$$

The results are given for the first design matrix formed in the first stage, from which ADAPRES yields the high order polynomial function shown in Eq.(5.13). These shows that it also gives better results in terms of the design point and the POF than the traditional RSF which is of a quadratic type, as illustrated in Figure 5.6 and 5.7. However, if the POF values of the exact solution and ADAPRES are compared, the result from ADAPRES is not satisfactory, though the design point obtained by ADAPRES is very close to that of the exact solution as indicated in Table 5.5. This indicates that the shape of the limit state around the design point is non-linear, so the POF computation from the method based on FORM is not adequate as is explained in Chapter 3. One of the approaches to solve that kind of problem in structural reliability analysis is to carry out the iterations more than once. In order to force further iterations, the value of perc_dif is chosen to be in very low such as, 0.1 in this example. The results for this selection are given in Table 5.6. Although the design point values of the exact solution are different, the POF obtained from ADAPRES with FORM is closer than that of the previous approach. As an alternative approach to

overcome this deficiency, ADAPRES utilises a method in which MCS is used along with the optimisation approach given in Eq.(5.7), for which the results obtained are given in Table 5.7.

Table 5.5: Reliability results for the third example.

	<i>Exact (MCS)</i>	<i>Bucher RSM</i>	<i>ADAPRES</i>
Design Points			
X_1	2.021	5.863	2.321
X_2	1.700	6.32	1.297
Reliability Index	2.550	1.093	2.306
POF	0.00539	0.137	0.030415
Absolute Perc_dif : 0.125%			

Table 5.6: The results in the second stage for ADAPRES with FORM.

	<i>Exact (COMREL)</i>	<i>ADAPRES</i>
The Design Point		
X_1	2.021	0.414
X_2	1.700	2.625
$g(X)$ for the design point	0.001973	0.494071
Reliability Index	2.550	2.406
POF	0.00539	0.00439
Absolute Perc_dif : 0.0210%		

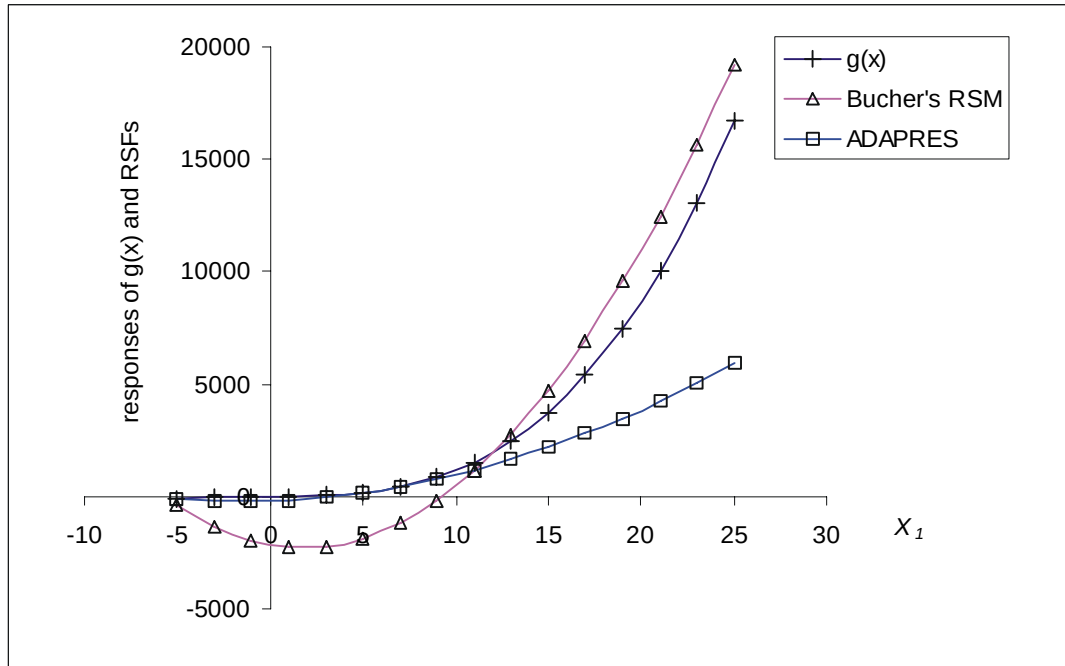


Figure 5.6: The comparison of Bucher's RSM and ADAPRES for the first stage for X_1 while the value of X_2 is kept at 1.7.

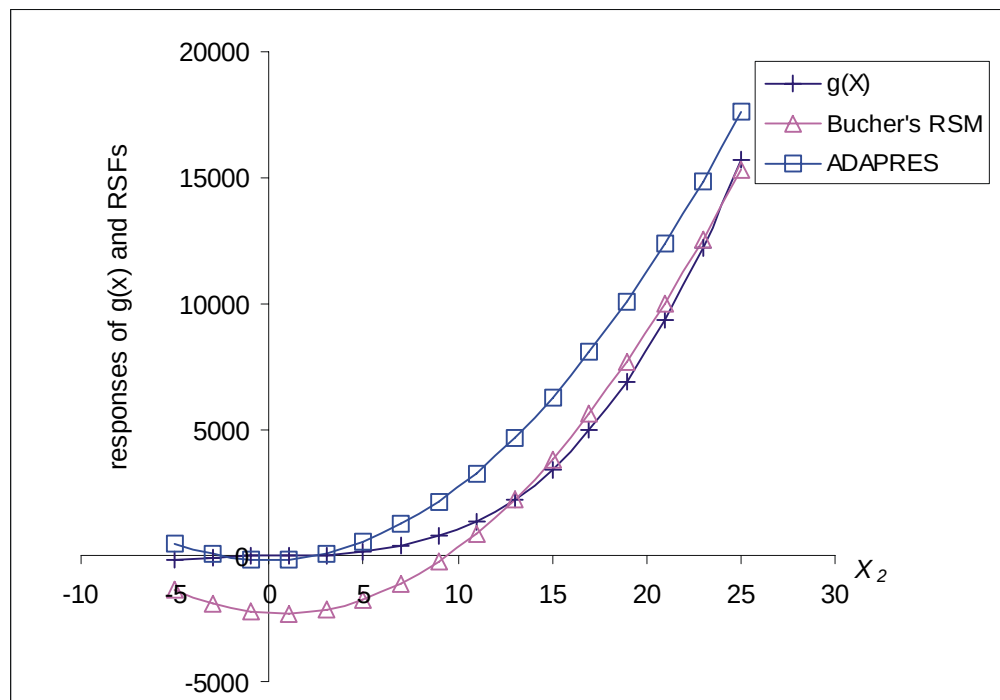


Figure 5.7: The comparison of Bucher's RSM and ADAPRES for the first stage for X_2 while the value of X_1 is kept at 2.021.

In the first stage, ADAPRES with MCS forms the same RSF given in Eq.(5.13).

Although the POF values for both approaches are the same, the reliability index and the value of the design point differ because ADAPRES with MCS makes use of Eq. (5.7) to get these values. As is explained in Section 5.2.2, the results from ADAPRES with MCS are more accurate than that of ADAPRES with FORM since the former approach searches the design point value that should satisfy the POF obtained from MCS. Therefore, in the second stage, ADAPRES with MCS gives a better POF that is very close to the value obtained by the exact solution.

Table 5.7: The results for the first and second stage of ADAPRES with MCS.

	Exact (COMREL)	Results from the first stage	Results from the second stage
The Design Point			
X_1	2.021	4.146	1.985
X_2	1.700	2.586	1.830
$g(X)$ for the design point	0.001973	116.331	2.597
Reliability Index	2.550	1.873	2.519
POF	0.00539	0.030	0.00587
Absolute Perc_dif : 0.107%			

5.5 Conclusion

In this chapter an improved response surface method, ADAPRES, which includes a modeling error assessment, was explained in detail. By this method, the importance of term(s) can be determined without carrying out reliability analysis, which enables ADAPRES to form a RSF which might include cross terms or higher order terms, depending on the modeling error evaluation. ADAPRES also makes use of a design of experiments approach by which the experimental points can be replaced close to the limit state. Two examples were given in the chapter, showing the application of ADAPRES in which better results were obtained than by using a standard method. In Example 1, it is shown that by applying a linear RSF, an adequate result is obtained. Example 2 indicates that ADAPRES is an efficient and computationally inexpensive RSM. For Example 3, a higher order polynomial RSF was generated by ADAPRES as the best option for the given design matrix to approximate the performance function.

Chapter 6

A Tool Management System: ADAPRES_NET

This chapter includes a description of explanation on the probabilistic design system which is called ADAPRES_NET in which a structural reliability problem is resolved by applying the structural reliability methods along with ADAPRES. In this chapter, the first section explains the necessity of using a distributed network application in structural reliability analysis along with some information on structural reliability analysis for a structural system. In the second section, a basic structure for distributed networking is given in which the fundamental concepts of network programming for structural reliability analysis are explained. Finally, the third section consists of the basic architecture of the program, ADAPRES_NET, by which a probabilistic design system can be modelled for a structural reliability problem.

6.1 Introduction

In Chapter 2, design integration by means of distributed networked programming was explained briefly in the context of Petri nets. As was shown in that chapter, there are two ways in which networks assist in computing of structural reliability problems. Firstly, many structural analysis tasks require a combination of analyses, the interactions and data flows between which can be described using a network model. Secondly, where there are computational tasks that can be carried out in parallel (either because there are different computing applications that can execute in parallel, or because the same application may execute repeatedly for different input values) then the time required for the computation can be reduced by distributing computational jobs around a physical computing network. The authors' research group has developed systems, which has been called "Tool Management Systems" to support both the control and sequencing of computing processes using a network model and the execution of distributed processes on a computing network [McMahon 1996]. In this present study, the adaptive response surface method has been incorporated into these tools, and other enhancements have been incorporated in a computing application called ADAPRES_NET. In this program, evaluation of multiple performance functions of a structural reliability problem using response surface functions or direct calls to the performance function can be distributed over the workstations on a network. In addition, the evaluation of the response surfaces approximating the different performance functions can also be carried out concurrently on the network, which can reduce the time spent on the evaluation of the multiple performance functions of the structural reliability problem.

6.2 General Structure of ADAPRES_NET

In the previous chapters, structural reliability analysis has been presented for a single failure mode, defined by a single performance function. Most structural engineering problems, however, often have multiple failure modes; namely, there might be many potential modes of failure, in which the occurrence of any of the potential failure modes will cause failure or non-performance of the system or component. A connecting rod, for example, is subjected to different loading conditions, which are modelled by using different analyses such as dynamic analysis,

stress analysis, fatigue analysis, buckling analysis. For these analyses, a different limit state formulation for each analysis must be established in order to take into account their effects in structural reliability analysis. For a structural reliability problem requiring multiple analyses, a probabilistic design system called ADAPRES_NET has been developed in the present study. In ADAPRES_NET, the interaction and dependency among the processes is modelled by using Petri nets which are explained in Chapter 2. The processes could be data file preparation for the analyses, stress analysis, fatigue analysis, formation of a response surface function for a performance function and so on which are represented by transitions, and each transition can in principal take place on a separate workstation. The sequential or parallel execution of the transitions as well as data transfer among the transitions are enabled by a central controller program in which dependencies within the Petri net of the process and the state of the Petri net are represented by a matrix notation as given in Appendix A.

As given in Chapter 5, ADAPRES can be used in place of the time consuming performance function in a structural reliability analysis. In the case that the structural reliability problem includes multiple performance functions, a RSF obtained from ADAPRES can be used for each performance function by fitting ADAPRES into the Petri net as a transition. In the case study that will be given Chapter 7, for example, the connecting rod is subjected to two analyses: stress analysis and buckling analysis. Since the probabilistic evaluation of stress analysis and buckling analysis is a time consuming process, ADAPRES replaces each analysis in the Petri net as will be shown in Chapter 7.

ADAPRES_NET can also be used to assist in building a response surface function in ADAPRES by distributing each call to the real performance function to a workstation, and feeding the result to ADAPRES. In order to provide this functionality, the system must have the capability to execute programs on different computers under central control, to send data between processes, to call the required programs remotely and so on, and this is discussed in the following section.

6.3 Fundamentals of Networking

ADAPRES_NET uses the network programming concept to distribute computing processes and information flows over a local network, based on UNIX network

programming methods. With the development of the Internet, networking has become so popular that it might be unnecessary to give an explanation of the benefit of using a network environment to distribute engineering applications. However, a very short description of the basics of network programming techniques could not be avoided in this study to define the terms which are used to explain the network programming of ADAPRES.NET. Therefore, the following sections begin with a description of the Client/Server concept in networking, and continue with the explanation of socket programming used to establish a connection between Server and Client.

6.3.1 Client/Server Concepts

The Client/Server concept divides a network application broadly into two sides: the client side and server side. The client side of a network connection requests information or services from the server side of the network connection. The server side of a network connection responds to a client's requests. In the Client/Server program model, a network application performs two separate functions: requesting information and responding to requests for information. The program that requests information function is called a client program, while the program that responds to such requests is called a server program [Jamsa 1995] as is illustrated in Figure 6.1.

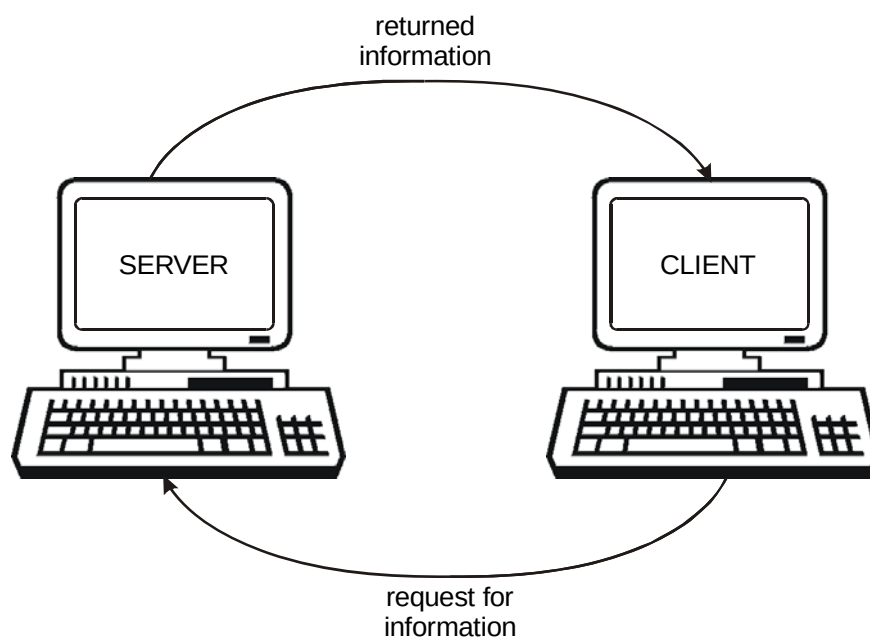


Figure 6.1: A simple representation of Client/Server networking concept.

The client/server model enables the selection and location of computing resources according to the needs of individuals and work groups. One of the prime benefits of a client/server system is to increase the productivity from the individual workstation to a network that might include many workstations, resulting in a better access to information as well as distributing the resources on a network. Additional benefits of the client/server approach can be summarised in the following [Orfali 1997]:

- Many client processes can share the resources provided by a single server, thus the non-critical data and functions can be moved to workstations, while the server can be left free for critical processing needs.
- Client applications can concentrate on users' requests, and then they ask for the information requested from the server. Having obtained the information, they can analyse and present the data, using the display capabilities of the client workstation.
- Client applications can be designed with no dependence on the physical location of the data. If the data is moved or distributed to other database servers, the application continues to function with little or no modification.
- If multitasking and shared-memory facilities are enabled on the operating system, the Client/Server networking concept delivers the highest possible degree of concurrency, data integrity, and performance to its client applications.
- Client workstations or terminals can be optimised for the presentation of data (for example, by providing graphics and mouse support) and the server can be optimised for the processing and storage of data (for example, by having large amounts of memory and disk space).
- In networked environments, shared data is stored on the servers, rather than on all computers in the system. This makes it easier and more efficient to manage concurrent access.
- In networked environments, inexpensive workstations can be used both to access the remote data of the server effectively, and to place a remote call functions or programs on the server.

- A Client/Server system provides independence between application components, as well as reducing maintenance costs.

In ADAPRES.NET, Client/Server programming on the UNIX operating system is programmed by using the socket programming techniques, which is explained in the next section.

6.3.2 Socket Programming

Unix programs do any sort of input/output by reading or writing to a file descriptor. A file descriptor is simply an integer associated with an open file. Therefore, when a communication with another program over the network is required to be established, a file descriptor for the connection must be initiated. This is done by calling the `socket( )` system routine. It returns a **socket descriptor**, and communication is enabled through it using the specialised `send( )` and `recv( )` socket calls. Although these functions form the core of the socket programming, only a brief explanation is given in this chapter, because they are intensively related to computing science, for which more detailed explanations and application can be found in any network programming reference [Curry 1996] [Stevens 1990].

There are mainly two types of network sockets used in the UNIX system: Stream Sockets and Datagram Sockets. Stream Sockets provides reliable two-way connected communication streams by which the data sent over the network arrive in the order sent. Internet browsers such as Internet Explorer and Netscape use the HTTP protocol which uses stream sockets to get the requested pages. Stream sockets achieve data transmission by using a network protocol called the “Transmission Control Protocol/Internet Protocol”, TCP/IP for short, which makes sure data arrives sequentially and error-free [Hahn 1996]. Although this term, TCP/IP, became very common with the use of the Internet, TCP/IP is also used for most UNIX-based local area networking applications such as remote login and network file service. Thus, the TCP/IP based socket programming method is used to establish the communication in ADAPRES.NET as well as enabling transmission of data across the network in which the evaluations of the multiple performance functions are distributed.

6.3.2.1 Client/Server Based Network Programming

In order to establish the server side of networking, a collection of the functions of socket programming needs to be called in a sequence as is shown in Figure 6.2. For Server-side networking, first of all, a `Socket()` system function is called to get a socket descriptor by which the communication with other machines on the network is established. This function call returns a descriptor similar to a file descriptor that is used to open or close a data file in any programming language such as C, C++. The socket descriptor differs according to the domain in which the Client/Server networking exists. The first is the `AF_UNIX` domain, in which addresses of both client and server are ordinary UNIX pathnames, while the second one, `AF_INET`, describes a domain in which internet addresses like Internet Protocol (IP) address are used. Having set up a socket call, the next step is to bind this descriptor to one of the port numbers the operating system has generated, which is a small integer used to identify different services on the operating system. This is achieved by the `bind()` system call. Thus, client requests can access the server machine. To notify the operating system that the server is ready to accept calls from clients, `listen()` system calls are initiated in which the requests from more than one client are queued until the

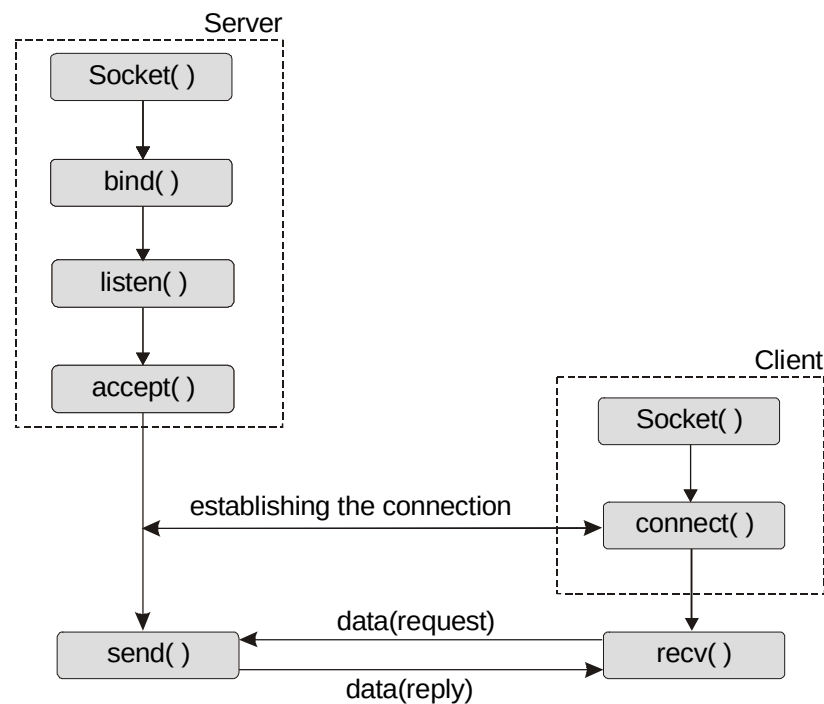


Figure 6.2: Main functions used in Client/Server networking.

server is ready to respond. The server uses the `accept()` system function call to actually accept a connection. When a connection request arrives on the socket, `accept()` returns a new socket descriptor by which the server can communicate with the client. After accepting the connection request, the requested data is sent by issuing a `send()` system function call.

Similarly, the Client-side of networking also requires establishment of a `socket()` call to get a socket descriptor by which a client program can communicate to a server. However, the client-side needs to establish a connection by calling the `connect()` system function call in which the socket obtained from `socket()` is connected to the server by issuing the server IP address. After establishing a successful connection, the client-side program issues a `recv()` system function call to receive the data sent from the server by the `send()` function call.

From the above explanations, both the client-side program and the server-side program must be invoked by the user simultaneously. This might cause a practical problem if Client/Server networking is used to receive/send data into an iteration loop such as in an optimisation problem for which the evaluation of the objective function and constraint functions are distributed to different machines. In such cases, user needs to invoke the related Client/Server programs on different machines separately for each iteration, which is practically impossible. Therefore, a remote shell invocation method has been implemented in ADAPRES.NET by which the required data transfer is carried out by only invoking one of the Client/Server programs. Thus, the program establishes the network connection and calls the program on the client or the server machine as is explained in the following section.

6.3.3 Remote Shell Invocation

One of the important capabilities of networking is to login on to a remote machine from the host machine, providing access to programs and data on the remote machine. Therefore, the UNIX system provides two system functions for remote access to hosts: `rlogin()` and `rsh()`. Although both system commands can be used for this purpose, there are some differences in the application of these commands. The `rlogin()` system call only enables the user to login on the remote host. After login, the user needs to issue the command he/she wants to execute in the command line. The

rsh() system call, however, not only enables the user to login on to the remote computer, but also to execute any command in the shell created in the remote computer. Therefore, the rsh() call can be used to remotely look at the status of other devices without connecting to the target machine, to execute commands, and to disconnect from the other machine without issuing any command in the target machine.

ADAPRES_NET requires access to remote hosts to distribute jobs around the network. If only one access is needed to invoke a program remotely on the remote hosts, it could easily be carried out by just logging in on the remote host where the requested program exists. However, as is explained in the later sections of this chapter, some structural reliability applications such as reliability based structural optimisation require a number of executions of the remote program. Thus, remote shell invocation is used in two places in ADAPRES_NET, one of which is concerned with data transformation across the network, and the other is resource sharing such as using a FEA package which is not installed on all the workstations on a network. Both are explained in the section dedicated to the explanation of the features of ADAPRES_NET.

6.3.4 Concurrent Process Calls

One of the most powerful features of the UNIX operating system is that it is a multi-user, multitasking operating system which means that more than one user can use the system at the same time to get work done and also that the system can do more than one thing at a time [Coffin 1990]. However, this does not mean physically the computer can carry out more than one thing a time unless it has multiple processors installed on it. Most computers have only one processor that can carry out only one task a time. The UNIX system, in common with many operating systems, creates the illusion that the computer is carrying out several tasks at once by timesharing the processor, which means that the single processor is resource that is shared by all users and processes. If a machine is shared by multiple users, first of all each user's shell is a process, and each process may request access to the processor at once. The processor spends a few fractions of a second doing one task for one user, and then switches to another process created by another user. Since microseconds are too short for the user to notice that the system is being used by other people or processes, it

appears that all the tasks are taking place simultaneously. This scheme works quite well since, when some processes are waiting for an input such as user input, the UNIX system can process other tasks required by other users or programs.

The UNIX system uses processes to split work into tasks by means of creating a separate process for each task. This is achieved by issuing the `fork()` system call in the UNIX system, for which the basic algorithm is illustrated in Figure 6.3 [Kernighan 1984]. This system call causes the current process to be split into two processes - a parent process, and a child process. All of the memory pages used by the original process get duplicated during the `fork()` call, so both child processes have all the features of the parent process. The only distinction is when the call returns. When it returns in the parent process, its return value is the process ID (PID) of the child process. When it returns inside the child process, its return value is '0'. If for some reason this call fails (not enough memory, too many processes, etc.), no new process is created, and the return value of the call is -1. In case that the process is created

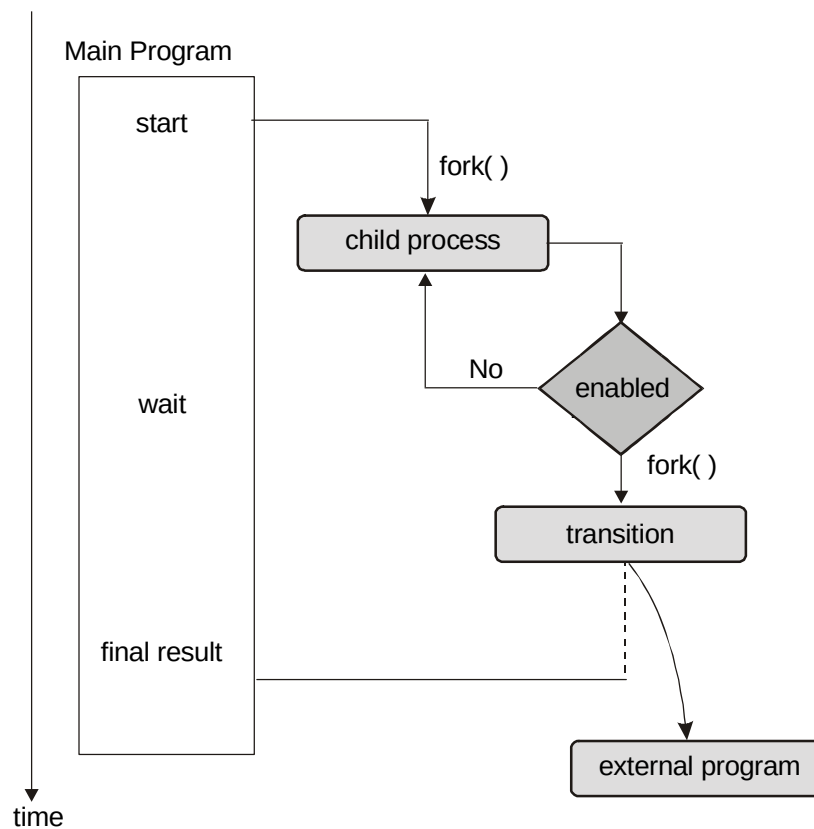


Figure 6.3: Concurrency by `fork()` system call.

successfully, both child process and parent process continue from the same place in the code where the fork () call is enabled. There is no guarantee that the child process will run before the parent process or vice-versa. This may cause some problems if each process is waiting for the other to do something, or each has some dependency on the other process. In order to avoid such a situation that yields a deadlock condition [Douba 1999], the second fork() call in Figure 6.3 is created in ADAPRES_NET to call the external programs to enable the Petri net controller, which is explained in Chapter 2, to control the execution of the child processes according to Petri nets diagram formed based on the problem involved.

Having given a brief introduction to the major techniques of the network programming, their special use in ADAPRES_NET is explained in the next section, based on the needs of structural reliability analysis.

6.4 Features of ADAPRES_NET

In order to accomplish the proposed probabilistic design system, a collection of programs have been developed and integrated together known as ADAPRES_NET. The main task that ADAPRES_NET carries out is to distribute the computation of the performance functions of structural reliability problems around a network as well as establishing concurrency over the network, based on the methods of network programming explained in the previous sections. Therefore, the preparation of the necessary programs and data files of ADAPRES_NET can be classified into three main areas as:

1. Data files:

The data files regarding both the structural reliability problem and the information of the server and the workstations on the network.

2. Server-side functions:

A number of programs designed to carry out the tasks on the server-side of the network.

3. Client-side functions:

A number of programs that are necessary to carry out the tasks on the client-side of the network.

An example of the Client/Server network approach designed for ADAPRES_NET on a local network environment is illustrated in Figure 6.4 for a general structural reliability problem having two performance functions in this case. In the local network, three workstations are used to model the structural reliability problem in the network. These workstations have the names Xenon, Brunel and Athena, and are named MAIN_CONTROLLER, WORKSTATION_1 and WORKSTATION_2 respectively in this chapter. These three machines carry out different tasks according to the structural reliability problem being addressed. MAIN_CONTROLLER, for instance, controls the flow of data among the other two machines as well as receiving the results obtained from the workstations. Two performance functions in this case are distributed to two workstations, and each invokes ADAPRES to find a RSF for the performance function. Another task in each workstation is to call the MCS program, MONTE_1, after having

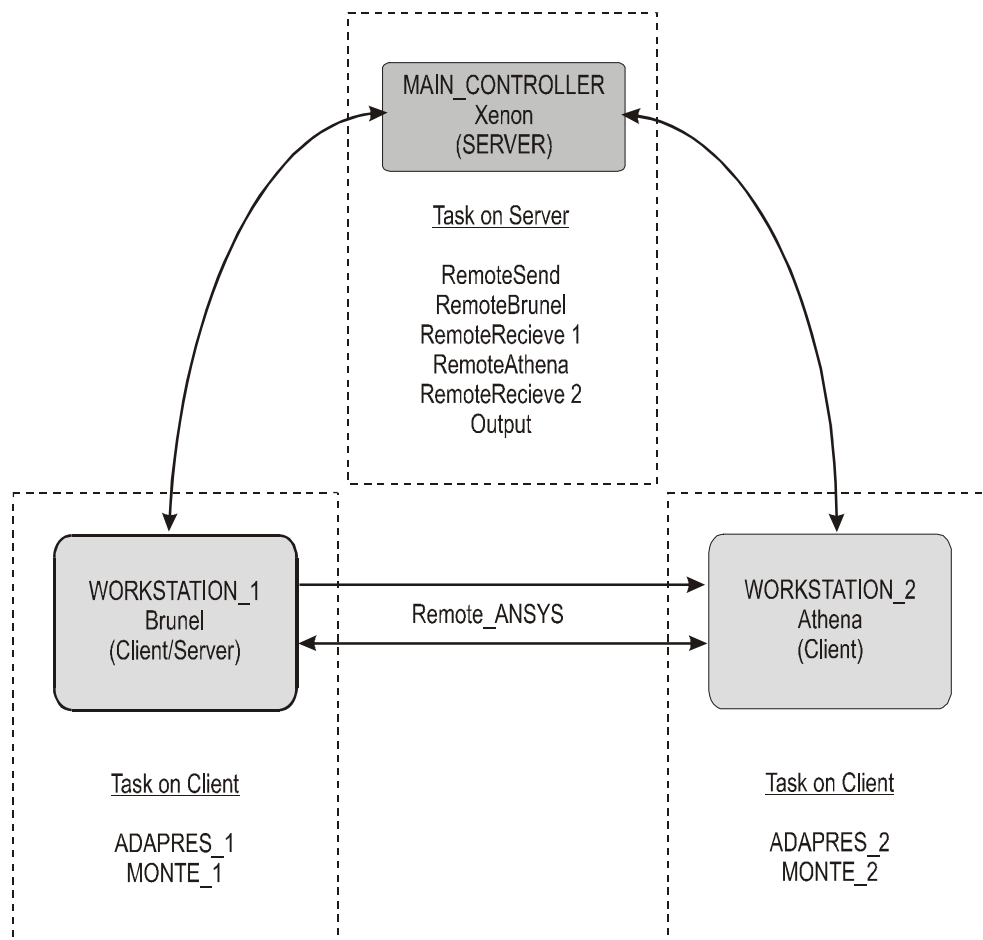


Figure 6.4: Representation of ADAPRES_NET for a structural reliability problem having two performance functions.

formed a RSF. The communication between WORKSTATION_1 and WORKSTATION_2 is established if any analysis on Athena requires to call programs such as ANSYS in this case, since ANSYS is only available on Brunel. This communication can be broadened to any type of information exchange between workstations in this local network.

6.4.1 Preparing the Data Files

One of the powerful features of the Client/Server networking is that all the data files can be kept in one place, and other programs on the network can access data by means of programs based on the network. This approach not only gives easy and fast access to the required data, but also provides security of the data. In ADAPRES_NET, the same approach is utilised to keep the data on the server so that all the workstations can access the required data, which also prevents duplication of the same data files. If the programs on the workstations require a data file that is unique to those programs or analyses, such files are kept on the related workstation. Keeping all data files in one place also enables the user to find the necessary data file easily without searching the file structure.

As can be seen from Figure 6.5, the file structure of ADAPRES_NET has many directories spread in one within the other. This might cause a problem if a new

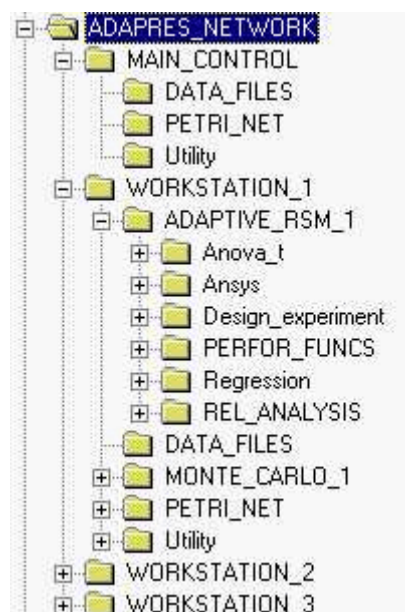


Figure 6.5: The file structure of ADAPRES_NET.

program or analysis is not invoked in a working directory of ADAPRES_NET, and they require to access the DATA_FILES directory, which results in an error if data files are not present or accessible. In order for the programs to access the directory where all the data files are kept, a header file was prepared in which the place of the directory is given. Thus, all programs can access this directory by giving reference to the header file in the main C program of the related programs.

The main data files requested by the server-side programs are given below along with a brief explanation of each data file.

prob_param.dat:

The main data file which includes the data relating to the structural reliability problem. This file contains information on the number of random variables and the mean values and standard deviations of the random variables together with the value of the arbitrary function, f_i . In addition to this problem dependent data, this file also includes the data by which ADAPRES_NET can choose the methods appropriate to the type of RSM and the reliability methods to find the design point.

Main_TMS.dat:

This file includes information on the number of transitions and places of the Petri net graph as in the example given in Figure 6.6.

Init_matrix:

This file comprises the Petri net matrix that shows the dependencies and the relationships between the places and the transitions in the Petri net graph.

Name_address_port.dat:

This file includes information on the number of workstations being used and the port numbers and IP numbers of both the server and the workstations.

The data files given above are the main files shared by both the server-side and Client-side programs. However, a number of data files are required when programs on either the server side or the client side are executed such as RemoteSend in which two data files are used, one of which is for the data to be sent to the workstations, and the second one is for the data sent by the workstations.

The data files on the Client-side are mostly problem dependent, and differ according to the problem involved. If the Client-side has a local Petri net graph, the data files regarding the graph are kept local since they are only required by the programs in the Client-side networking.

6.4.2 Server Side Functions of ADAPRES_NET

The main tasks of the distributed network are invoked from the server-side of the network. The server-side establishes a connection to send and to receive data as well as remotely invoking the necessary programs on other workstations. Thus, the server-side includes a program called MAIN_CONTROLLER, controlling the data flow as well as executing the required programs both on the host machine and the remote machines, designed by ADAPRES_NET. In order to explain the structure of the server-side networking in ADAPRES_NET, a simple scenario is given in terms of a Petri net design model, shown in Figure 6.6. In the figure, two workstations, named as WORKSTATION_1 (Brunel), and WORKSTATION_2 (Athena) respectively, which are the names of the Silicon Graphics machines used during the study, are connected to the network via MAIN_CONTROLLER, and the Petri net graph illustrates the interactions and communications between the Server and the Client programs. In addition, Figure 6.6 also presents the concurrent calls to the workstations, which carry out different tasks in context of the structural reliability problem. After preparing the necessary data files mentioned above, the MAIN_CONTROLLER begins executing the

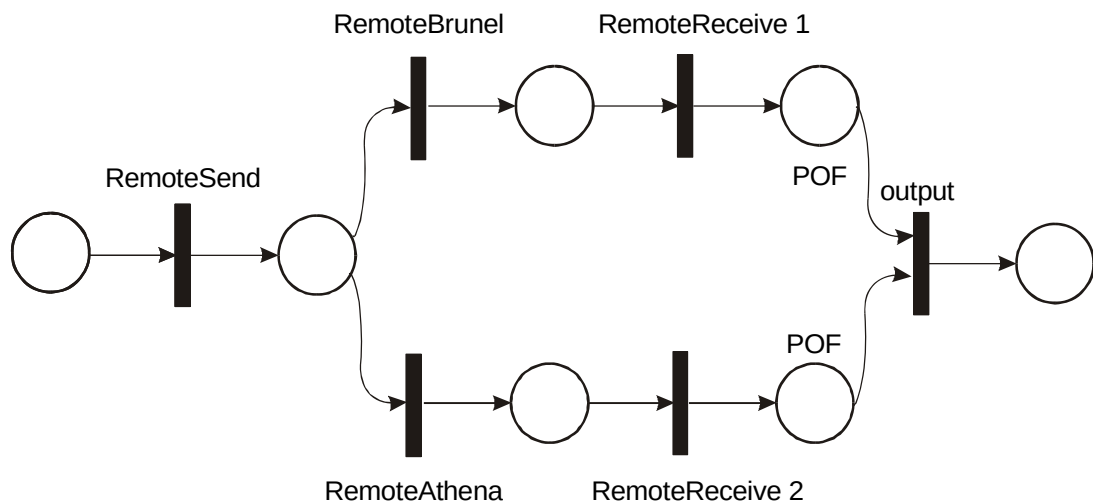


Figure 6.6: Petri net for server side of ADAPRES_NET.

following programs.

RemoteSend:

Sending the necessary data files to the workstations

RemoteBrunel and RemoteAthena:

Calling the programs on the remote workstations from the Server

RemoteReceive:

Receiving the outputs from the workstations

Each program call is explained in detail in the following sections since they form the core of ADAPRES_NET. Additionally, an application of ADAPRES_NET to an engineering problem is also given in Chapter 7.

6.4.3 RemoteSend

The server side of ADAPRES_NET keeps the data file including the problem parameters such as the value of random variables, material properties, geometry of the component. So, the data file is sent to workstations by invoking the program RemoteSend in which the necessary programs for data transfer across the network are invoked concurrently. On the server side, RemoteSend is called first, in which three main tasks are carried out within the program:

1. Invoking the program, SendData in the server side, which sends data over to the workstations requiring this data file.
2. Invoking the remote program, ReceiveData on the workstation, which receives data from over the network.

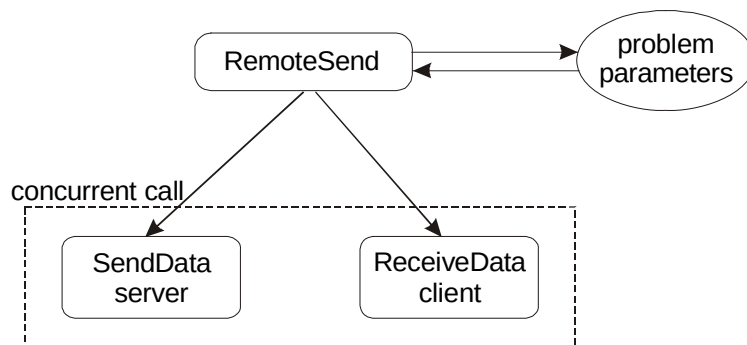


Figure 6.7: A schematic representation of RemoteSend.

3. Establishing the concurrency by means of the Petri net control mechanism so that both programs can communicate each other.

In order to send data over the network two programs are needed as shown in Figure 6.7. One is on the server side that can send the data, and the other on the client side that can receive data from the network. These programs must be invoked on both sides simultaneously so that the necessary connection between Client and Server can be established. Therefore, only RemoteSend is invoked in ADAPRES.NET, in which SendData on the server side and ReceiveData on the client side are called concurrently. In order to invoke the latter program, a remote shell invocation is established to the workstation. Hence, the user does not have to invoke remote program on the client side by logging on the workstation and calling the required program.

6.4.4 Remote Invocation of the Programs on the Network

The main purpose of developing ADAPRES.NET is to alleviate the computational burden of a structural reliability problem having more than one performance function by distributing and evaluating multiple performance functions concurrently over the machines on the local network. For the example illustrated in Figure 6.6, the programs required on the client side, which are RemoteBrunel and RemoteAthena, are invoked from the server side by means of a remote shell invocation as explained in Section 6.3.3. This procedure plays a crucial role in the distributed computing over the network. In an optimisation problem, for instance, the remote programs on workstations might be needed in an optimisation loop, and they might be called a great number of times. In such cases, the remote shell invocation automates the calls to the programs, making possible to invoke the distributed analysis on the remote machines.

6.4.5 RemoteReceive

After remote invocation of the program on a workstation has completed, a program, RemoteReceiver on the server side is called to receive the result from the workstation. As given in Figure 6.6, for each remote program call replaced on the workstations, there is a separate call for RemoteReceive since the computation time or

dependency of the transitions in the Petri net might be different. The structure of this

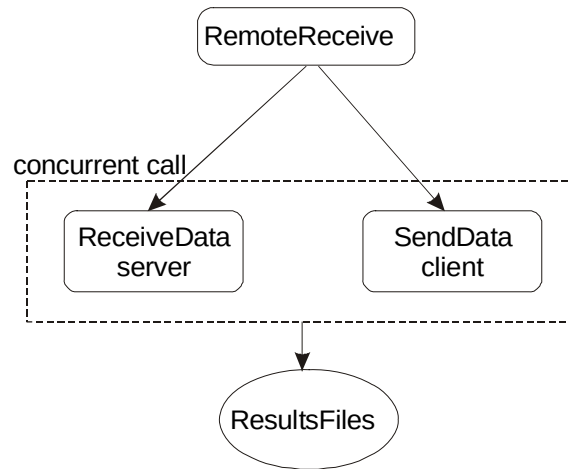


Figure 6.8: A schematic representation of RemoteReceive.

procedure is similar to the RemoteSend, but the programs on the server side and the client side are the other way around as illustrated in Figure 6.8 since the client sends the data while the server is waiting to receive the data from the server.

6.4.6 Client Side Functions of ADAPRES_NET

As explained in the previous section, the server side of ADAPRES_NET includes the overall control programs that manage the send/receive data to/from the workstation on the network, system calls that remotely calls the programs on the workstations. In the client side of ADAPRES_NET, however, the programs required can mainly be divided into two groups:

1. Programs required for sending/receiving data to/from the server side of ADAPRES_NET, such as SendData, ReceiveData.
2. Programs required for the evaluation of the performance function such as ADAPRES to form a response surface function.

Beside those main programs, the client side might also include some external program such as ANSYS, I-DEAS to get the response from the performance function. It must be recalled that those external programs are called from ADAPRES, and ADAPRES itself is invoked by a server side function such as RemoteAthena to enable the automation of the evaluation of the performance function.

6.5 Conclusion

In this chapter, the network programming has been explained in the context of developing a system in which the evaluation of performance functions required by structural reliability analysis can be distributed over the network. In addition, concurrency can be established in order to run the necessary programs on different workstations in parallel, so reducing the time spent on the evaluation of each performance function. Although ADAPRES_NET is based on a “Tool Management Systems” developed in author’s research group, further improvements have been done in this work mainly in two areas:

1. Data send/receive has been automated so that only the MAIN_CONTROLLER invokes the necessary programs such as SendData or ReceiveData in both the Server side and Client side of the local network, as explained in Section 6.4.3
2. The programs on the workstations can be remotely invoked by MAIN_CONTROLLER which enables automating the calls to those programs on the workstations, as given in Section 6.4.4.

The explanation of the developed system, ADAPRES_NET, has been given in two main sections, one of which is on the methods of network programming used in ADAPRES_NET, and the second one is more specific explanation of these method in terms of their use in ADAPRES_NET.

The use and benefit of ADAPRES_NET is presented in the next chapter, in which ADAPRES_NET is applied to a structural reliability problem.

Chapter 7

Case Study: An application of ADAPRES_NET to an Engineering Problem

This chapter describes a case study involving a reliability based optimisation of a connecting rod for which two probabilistic constraints are set, one of which is based on the maximum Von Mises stress, and the other is the critical load for a linear buckling analysis. In the first section of the chapter, a brief discussion on Reliability Based structural Optimisation (RBSO) is given in the context of its application in ADAPRES_NET. In the second section, the modelling of a connecting rod in ANSYS is presented in detail. The combination of the analyses applied in RBSO, stress analysis and buckling analysis are explained in the third section. The distribution of the tasks by ADAPRES_NET for the case problem is explained in the fourth section of the chapter while the use of ADAPRES is given in the fifth section. In the final section the results obtained from the RBSO are presented.

7.1 Problem Description

To illustrate the application of ADAPRES_NETWORK, the optimisation of a well-known mechanical part, a connecting rod, has been chosen. The connecting rod is illustrated in Figure 7.1. It is a highly loaded part of a car engine which transfers the firing pressure in the combustion chamber from the piston into the crankshaft. The representation used in this study is an idealisation of a connecting rod, as will be explained below.

The uncertainty evaluation of the connecting rod is carried out in a reliability based structural optimisation (RBSO) problem in which the probabilistic constraint includes two different structural analyses, stress analysis and linear buckling analysis. The RBSO is established as follows:

$$\begin{aligned}
 &\min \quad \text{volume} \\
 &\text{subject to} \\
 &\quad P_f(g_{\text{stress}}(\mathbf{X})) \leq P_{\text{stress}} \\
 &\quad P_f(g_{\text{buckling}}(\mathbf{X})) \leq P_{\text{buckling}} \\
 &\quad \mathbf{X} \geq 0
 \end{aligned} \tag{7.1}$$

where $\mathbf{X}$ shows the vector for the design variables, $g_{\text{stress}}(\mathbf{X})$ represents the performance function for the stress analysis and $g_{\text{buckling}}(\mathbf{X})$ shows the performance

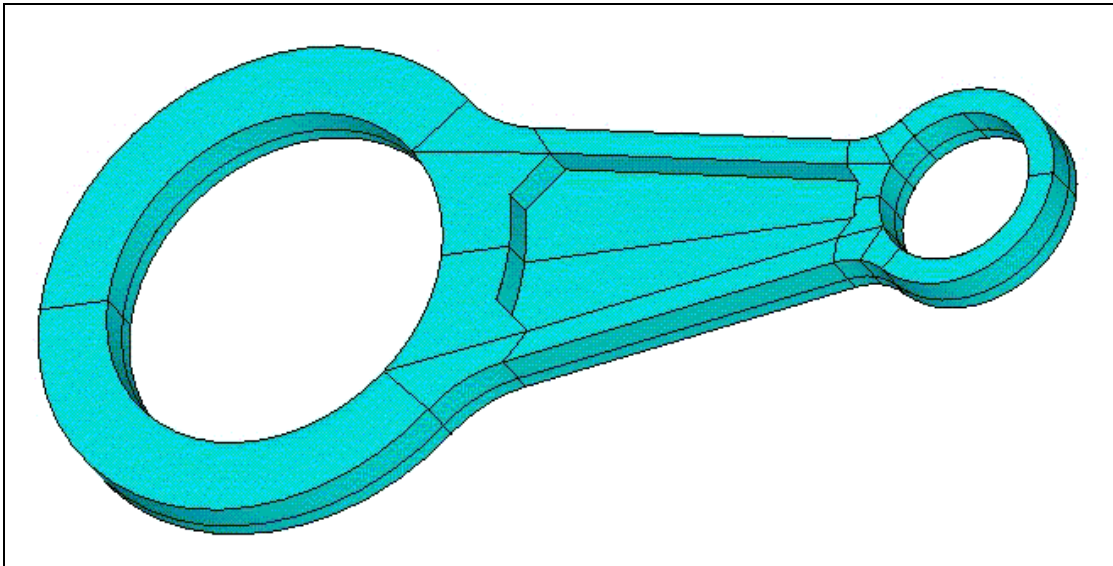


Figure 7.1: 3D representation of the connecting rod.

function for buckling analysis. P_{stress} and $P_{buckling}$ are the allowable probability of failure for the maximum stress and for the critical load in linear buckling analysis, respectively. A More explicit representation of the performance functions given in Eq. (7.1) is as follows:

$$\begin{aligned} g_{stress}(X) &= \sigma_{max} - \sigma_{yield} \\ g_{buckling}(X) &= F - F_{cr} \end{aligned} \quad (7.2)$$

where σ_{max} is the maximum Von Mises stress, σ_{yield} is the yield stress for the material, F shows the force that arises from the pressure acting on the connecting rod, and F_{cr} is the critical buckling load.

Three workstations have been used to carry out the tasks that are represented in Section 7.5 in detail, and the probabilistic analyses of the constraints are distributed to two of the workstations. For the constraint functions, response surface functions were formed for each generated set of design variables of the optimisation problem by ADAPRES. Both distributing the jobs to the workstations and using the RSF can reduce the evaluation time of the probabilistic constraints as shown in Section 7.6.

7.2 Parametric Modelling of the Connecting Rod

In RBSO, the values of the design variables are generated until the constraints set for obtaining the objective given in the objective function are satisfied. Therefore, for each generated value of the design variables, the connecting rod is updated. The geometric model of connecting rod given in Figure 7.2 is parameterised in order to make the model's changes possible without interaction from the user. The parametric

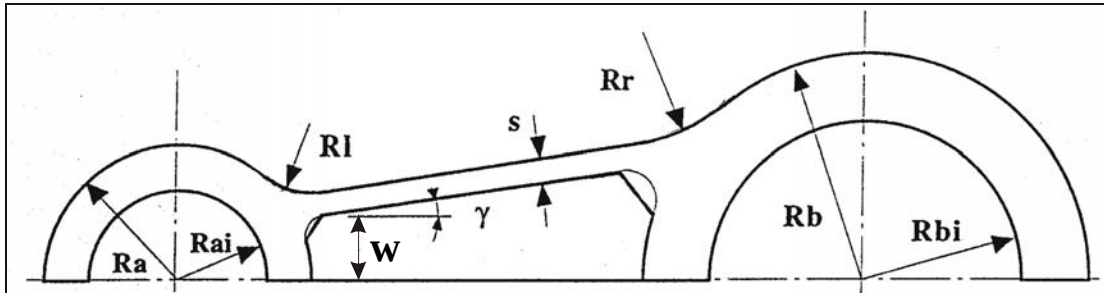


Figure 7.2: Parameters and geometry of the symmetric connecting rod.

modelling also enables reduction in the time for modifying the model. For the stress analysis, the Parametric modelling technique in ANSYS for the actual model in 2D was adapted from [Wendt 1997], which is shown in Figure 7.2.

Due to the fact that a common connecting rod is made of forged steel and has therefore a rather complex shape, the following main simplifications for the model were considered at the time of analyses of the connecting rod:

- The two bolts on the big end were left out from the model
- The big end split was omitted and the component was regarded as one unit
- Tapered sides with forging draft were changed into parallel ones

7.2.1 FEA Model of the Connecting Rod

Two models are generated for the analysis of the connecting rod due to the differences in the applied analyses; i.e. stress analysis and buckling analysis. Instead of taking into account the whole geometric model of the connecting rod for the stress analysis, one quarter of the whole model is taken into account to reduce the computational expense in terms of the number of the elements in the FE meshes. The model is meshed by using the free mesh feature of the ANSYS FE code, which has no restrictions in terms of element shapes, and has no specified pattern applied to it. Although mapped meshing of the connecting rod has some benefits such as reducing the computational time and ease in specifying the region where more refined meshing is needed such as in the area of the fillets, it has also some drawbacks for use in an optimisation loop since mapped meshing is restricted in terms of the element shapes it contains and the pattern of the mesh. These restrictions cause problems if the parameters are not carefully assigned, which might lead to some element shapes not satisfy the shape conditions of elements for mapped meshing. Free meshing is also very much easier to automate through a command script. Therefore, free meshing with triangular elements is selected for the meshing of the connecting rod. Commands for the construction of the geometric model, for meshing it and then applying loads are incorporated in a command file using the ANSYS Parametric design Language (APDL) incorporated in the ANSYS finite element suite used for the work. Variations in the model are produced by altering the numeric parameters in an APDL file. The general approach to modelling of the connecting rod using APDL is

based on the standard approach to FE modelling, and is as follows:

1. A solid model of the part is constructed, using dimensions entered as variable parameters.
2. Boundary conditions and loads are applied to the solid model.
3. The model is meshed.
4. The model is solved.
5. The required parameter (stress value etc.) is extracted from the results value
6. 1-5 are repeated for all values of the input parameters.

Appendix C includes annotated APDL files for each type of model used here.

7.2.2 Applied Loads

Since the main aim of the present thesis is to represent the integration of different analyses under the probabilistic problem domain, some simplifications on the analysis as well as the model itself were done as explained in the previous section for the model. In reality the gap between the pin and the hole of the small end of the connecting rod as well as the big end of the connecting rod and crankshaft is filled with oil, increasing the complexity of the modelling the forces acting on the connecting rod due to non-linear affect of the pressure [Lang 1979]. In this study, however, a linear distribution is assigned to forces acting the connecting rod by which non-linear loads or surfaces are approximated by subdividing the areas/lines.

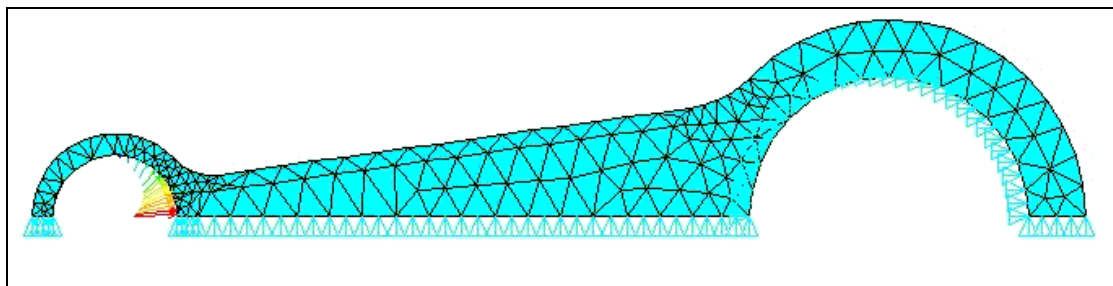


Figure 7.3: Meshing and applying the surface load to the connecting rod.

7.3 Applied Analyses: Stress Analysis and Buckling Analysis

In order to model the connecting rod under the uncertain problem parameters, two analyses have been chosen: stress analysis and buckling analysis. Although the main concern of designing a connecting rod is to design according to fatigue analysis [Rabb 1996] [Yi 1999], the aim in this study is to show the application of ADAPRES and ADAPRES_NET. Thus, the main effort has been spent on the probabilistic evaluation of the given analysis and the distribution of their evaluation around the network rather than concentrating on true design of the connecting rod that require much more rigorous analyses such as combined nonlinear stress analysis, fatigue analysis and dynamic analysis.

7.3.1 Stress Analysis

Yielding is considered as a failure mechanism, thus the Von Mises stress is obtained from the stress analysis from ANSYS to establish the failure criteria. The Von Mises yield criterion states that yielding occurs when the Von Mises stress exceeds the yield strength in tension [Boresi 1993].

The quarter of the actual connecting rod geometry is used for stress analysis due to symmetry by which the computational time can be reduced dramatically, especially for meshing the model. A typical Von Mises stress distribution is shown in Figure 7.4.

As the constraint in Eq. (7.1) only requires the maximum value for the structural reliability analysis, only the required results must be extracted from the result file since most commercial software like ANSYS record very detailed information on the result file. To avoid searching for a single value in the result file, the following APDL

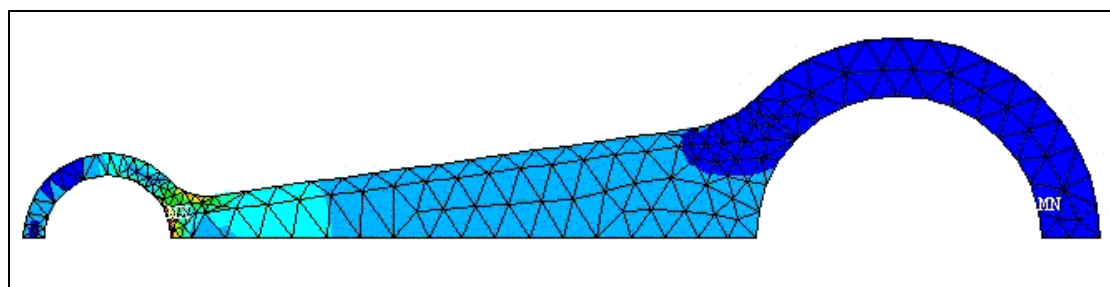


Figure 7.4: Stress distribution of the connecting rod.

command is issued in the program file by which ANSYS is called to get the maximum Von Mises stress:

```
NSORT, S, EQV,0,,10 !sorting the Von misses stresses
```

```
*GET,STRSS,SORT,,MAX !obtaining the maximum value from the sorted values
```

The maximum value is saved in a file so that ADAPRES can be used to find the best possible RSF according to the algorithm explained in Chapter 5.

7.3.2 Buckling Analysis of the Connecting Rod

Buckling analysis is a technique used to determine buckling loads and buckled mode shapes. The buckling loads are the critical loads at which a structure becomes unstable. Two methods are available in the ANSYS program to determine these critical loads and mode shapes: eigenvalue buckling analysis and nonlinear buckling analysis.

An eigenvalue buckling analysis predicts the theoretical buckling strength of an ideal linear elastic structure. This method corresponds to the textbook approach to elastic buckling analysis: for instance, an eigenvalue buckling analysis of a column will match the classical Euler solution. However, imperfections and nonlinearities prevent most real-world structures from achieving their theoretical elastic buckling strength.

A nonlinear buckling analysis is simply a nonlinear static analysis extended to a point where the structure reaches its maximum load. The basic approach is to constantly increment the applied loads until the solution begins to diverge. The only disadvantage of this method is that a diverging solution can also be caused by numerical instabilities. Tracking the load-deflection history of the simulated structure can help to decide if an unconverged load step represents structural buckling or not. For the case study in the present thesis, Eigenvalue Buckling analysis is used to find the effect of the applied load. Some theoretical information on the Euler buckling analysis is given in the following section.

7.3.2.1 Euler Buckling Analysis

A very brief derivation of Euler's equation can be well represented by a loaded pinned-pinned column shown in Figure 7.5.

A top section of the diagram is shown with the bending moment indicated, and in terms of the load P , and the deflection distance y , it is expressed as:

$$M = -Py$$

Since the bending moment is proportional to the curvature of the beam, which, for small deflection can be expressed as:

$$\frac{M}{EI} = \frac{d^2 y}{dx^2}$$

in which E represents Young's modulus, and I is the Moment of Inertia. After applying the boundary conditions, the critical buckling load is obtained as:

$$P_{cr} = \frac{\pi^2 EI}{L^2} \quad (7.3)$$

where P_{cr} stands for the critical load which can be applied before buckling is initiated. The critical stress can be obtained from Eq. (7.3) as:

$$\sigma_{cr} = \frac{P_{cr}}{A} = \frac{\pi^2 EI}{L^2 A}$$

where A is the cross-section of the beam. It must be pointed out that Euler's formula applies only while the material is in the elastic/proportional region that means the critical stress must not exceed the proportional limit state.

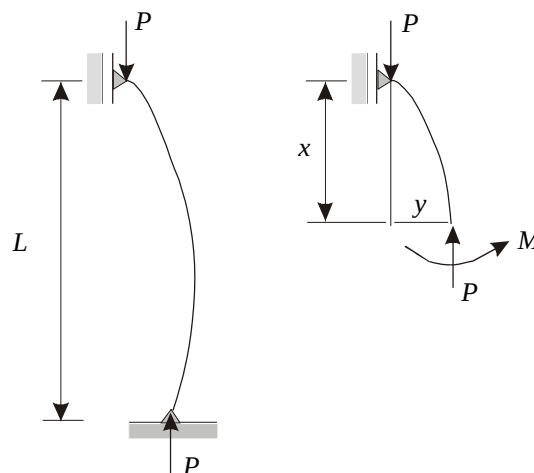


Figure 7.5: A loaded pinned-pinned column, adapted from [UW-stout 2000].

The basic steps of Eigenvalue Buckling Analysis in ANSYS can be summarised as follows:

1. Build the model.
2. Obtain the static solution.
3. Obtain the eigenvalue buckling solution.
4. Expand the solution.
5. Review the results.

Explanation of the buckling analysis used in the evaluation of the first constraint in Eq. (7.1) is given in the next section based on the above points given for ANSYS buckling analysis.

7.3.2.2 *Building the Model*

Although a model exploiting symmetry of the connecting rod was used for the stress analysis, a full model of the connecting rod had to be used for the buckling analysis because the symmetry boundary conditions on the centre-line of the rod shown in given Figure 7.3. would prevent the connecting rod from buckling, and therefore a full model of the connecting rod is generated in ANSYS, as shown in Figure 7.6.

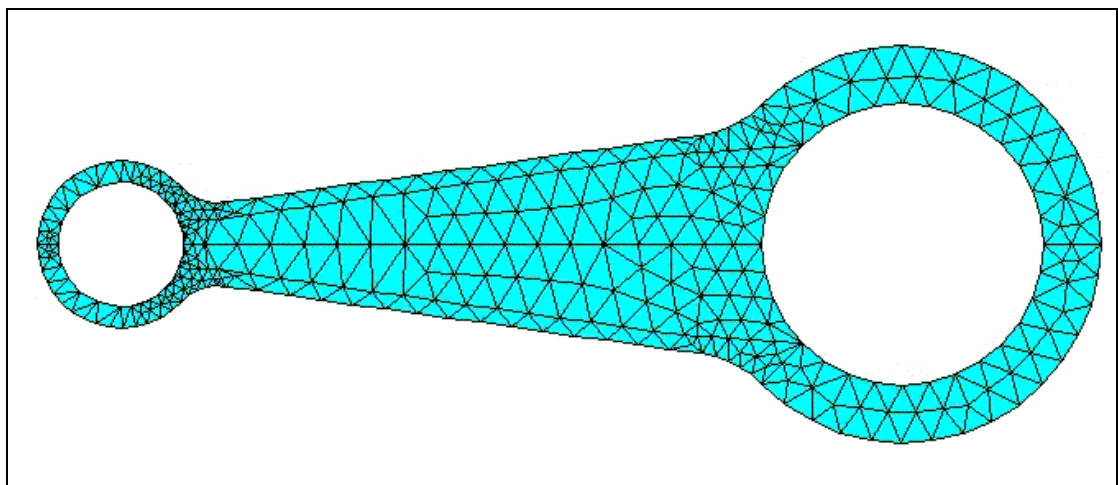


Figure 7.6: Meshed geometry of the full- connecting rod for buckling analysis.

7.3.2.3 *Obtaining the Static and the Buckling Solution*

Buckling analysis requires two analysis types to be carried out to obtain the critical buckling load: static analysis type and buckling analysis type. Prestress effects must be activated in the static analysis of the connecting rod since eigenvalue buckling analysis requires the stress stiffness matrix to be calculated. Unit loads are usually sufficient that is actual load values need not be specified. The eigenvalues calculated by the buckling analysis represent buckling load factors. Therefore, if a unit load is specified, the load factors represent the buckling loads.

For the buckling analysis type, three different methods can be selected for eigenvalue extraction methods, which are Subspace iteration method, Block Lanczos method and Reduced (Householder) method. More detailed information on these methods can be obtained from [ANSYS 2000].

7.3.2.4 *Expanding the Solution and Reviewing the Result*

In order to review the buckled mode shape(s), the solution must be expanded regardless of which eigenvalue extraction method is used. The critical buckling force (FCR) which is required to form the performance function in Eq. (7.2) can be extracted by issuing the following command:

```
*GET,FCR,MODE,1,FREQ
```

The result is saved and passed to ADAPRES to form the RSF for the buckling performance function given in Eq. (7.2).

7.4 *Response Surface Approximation by ADAPRES*

As stated earlier, the probabilistic evaluation of the constraint functions in Eq. (7.1) takes considerable time. Therefore, for each performance function given in Eq. (7.2), ADAPRES forms a RSF to be used in structural reliability analysis in order to obtain the POF for the corresponding constraints. As explained in Chapter 5, ADAPRES does not assume any type for the RSF to be formed at the beginning. Instead, it starts with a linear type and adds higher terms according to the modelling error assessment. The random variables used in ADAPRES are given in Table 7.1.

Table 7.1: Random variables for this case study.

<i>Random Variables</i>	<i>Mean Values</i>	<i>Standard deviation</i>
Combustion peak pressure (MPa)	7	0.5
Modulus of elasticity (MPa)	2.08E5	5.2E3
Passion's ratio	0.3	0.0055
Material Yield Strength (MPa) (AISI 4241 Forged Steel)	2650	500

7.5 Distributing the Jobs by ADAPRES_NET

To carry out the RBSO for the connecting rod, ADAPRES_NET is used to model the interaction and communications among the analyses required. Three workstations are used in ADAPRES_NET, named as Xenon, Athena, Brunel, and the tasks carried out in those workstations are explained in the next section.

7.5.1 Tasks in Main_Controller:

The programs that controls the Petri net diagram, given Figure 7.7 resides in the workstation named Xenon. The tasks carried out by the Main_controller are listed as follows:

- Starting the RBSO
- Sending the data to the two workstations
- Obtaining the volume of the connecting rod by calling ANSYS remotely from Brunel.
- Receiving the results from the workstations
- Passing it to the RBSO

After starting the RBSO, the generated value for the design variables are sent to the two workstations for the evaluation of the performance functions, which are called concurrently by ADAPRES_NET as explained in Chapter 6. The results from the workstations, which are the POF value for the constraints, are passed to the RBSO to assess whether they satisfy the constraint. In the mean time, the RSBSO might call the objective function to get the volume of the connecting rod according to the design

variables. The objective function and constraint function evaluation are carried out in the RBSO which is shown as the “Assessment” transition in Figure 7.7. If the design variables do not satisfy the constraint functions new values for the variables are generated by the CFSQP and the same loop is repeated until the optimum volume is obtained.

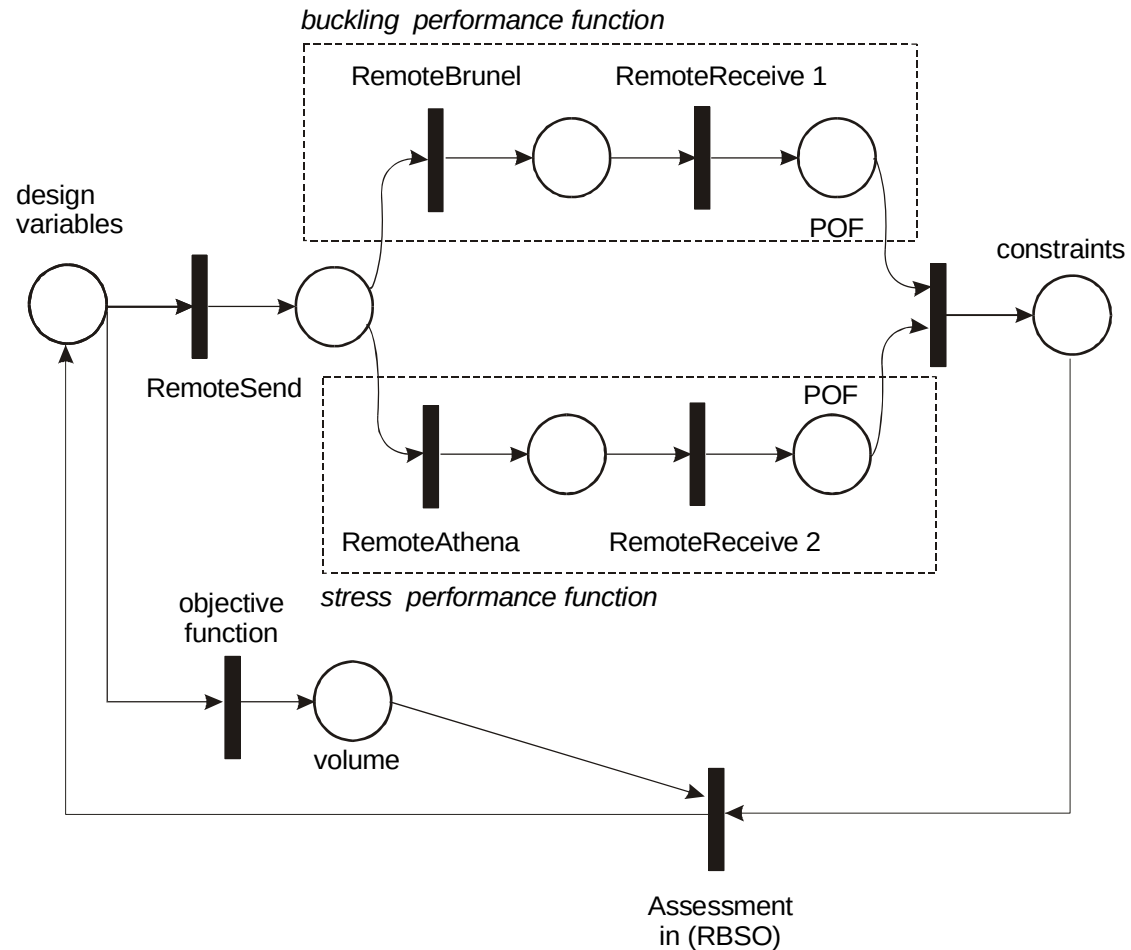


Figure 7.7: Petri nets for the RBSO of the connecting rod by ADAPRES_NET.

7.5.2 Tasks in Workstation_1(Brunel):

All the tasks required to evaluate the probabilistic constraints regarding the buckling analysis are carried out in WORKSTATION_1. These tasks are as follows:

- Receiving the data
- Forming the response surface for buckling analysis

- Obtaining the POF
- Sending the value of the POF to the main_controller

The generated design variables from the RBSO program are sent to WORKSTATION_1. A RSF is formed by ADAPARES by calling the external program ANSYS to get the response for the design matrix formed in ADAPRES. The value of the POF corresponding to the buckling constraint is obtained and sent to the MAIN_CONTROLLER.

7.5.3 Tasks in Workstation_2(Athena):

The evaluation of the probabilistic constraint regarding the stress analysis is carried out in WORKSTATION_2. Some of the main tasks carried out in this workstation are listed as follows:

- Receiving the data
- Remotely calling ANSYS to get the maximum stress result
- Forming the response surface for stress performance function
- Obtaining the POF
- Sending the value of the POF to the main_controller

Besides some of the common tasks such as receiving/sending data to MAIN_CONTROLLER, a remote call to WORKSTATION_2 is required to get the response for the design matrix from the ANSYS software which is only available on WORKSTATION_1. Then a RSF is formed by ADAPRES and the POF is computed by using the RSF in place of the stress performance function. For the assessment in the RBSO, the value of the POF is sent to MAIN_CONTROLLER.

7.6 Results and Discussion

The optimisation of the connecting rod to get the minimum volume but satisfying the constraints given in Eq. (7.1) was carried out as an RBSO problem by means of ADAPRES_NET in which the performance functions were replaced with the RSFs obtained from ADAPRES.

Table 7.2: Design Variables for RBSO.

<i>Design variables for RBSO</i>	<i>Min</i>	<i>Max</i>
<i>RI (cm)</i>	0.5	1.5
<i>W (cm)</i>	0.4	1
γ	3	8

Table 7.3: The parameters of the connecting rod shown in Figure 7.2.

<i>The parameters</i>	<i>Value</i>
<i>S</i>	0.3
<i>Rai</i>	1.2
<i>Rbi</i>	2.7
<i>Ra</i>	1.6
<i>Rb</i>	3.8
<i>Rr</i>	2

Beside the constant parameters given in Table 7.3 related the geometry of the connecting rod, there are two different design variable groups in this optimisation problem:

1. Design variables for the optimisation
2. Random variables for the constraint evaluation

Since three geometric parameters for which the initial values and the minimum and maximum values of the design variables are given in Table 7.2 are selected as the design variables of the connecting rod, the random variables given in Table 7.1 are only used for the evaluation of the performance functions. Therefore, for each generated value for the design variables in the optimisation problem, ADAPRES forms a new RSF for each performance function.

As an optimisation program the CFSQP [CFSQP 1998] is used for which more detailed information is given in Chapter 2. The results from the CFSQP are given for 5 runs, which are summarised in Table 7.4, and the best results according to minimum volume of the connecting rod is shown in shaded area.

Although the RSFs are formed for each performance function given in Eq. (7.2)

Table 7.4: Optimisation results after 5 runs.

Number of runs	1	2	3	4	5
Volume (cm³)	42.136	42.866	42.935	42.152	41.0089
Design Variables					
RI (cm)	1.5	0.849	0.8799	1.51	1.5
W (cm)	0.744	0.798	0.798	0.745	0.74
γ	5	7.85	7.856	5.01	7.613
Number of calls to the constraint functions	87	218	218	107	74
Total execution time of design optimisation (seconds)	13120	13812	14490	13150	13025

the execution time of the optimisation still took 13025 seconds (3.61 hours). In order to show the benefit of using ADAPRES in place of the buckling constraint evaluation, the evaluation of the buckling performance function for one set of the design variables was carried out by means of COMREL [Comrel 1999] in which conventional structural reliability methods such as the FORM/SORM are used to evaluate performance functions. COMREL requires approximately 19 calls to get the reliability data such as the reliability index and the POF. The linear Buckling analysis of the connecting rod in ANSYS approximately takes 17 seconds. Thus, the total time of the evaluation of the buckling constraint for one set of the design variables takes 323 seconds to complete. On the other hand, ADAPRES only requires approximately 8 calls to ANSYS to form a linear RSF, which results in a significant reduction in computational time as given in Table 7.5 with a comparison of the result obtained from COMREL [Comrel 1999].

For the optimisation problem, the responses of the objective and constraints functions are also observed, and are represented in Figure 7.8, 7.9 and 7.10, respectively. It must be noted that, the optimisation procedure and the methods are entirely determined by the optimisation software used, which is CFSQP [CFSQP 1998]. Although it allows to define some parameters related to the optimisation method used such as maximum number of iteration, perturbation size for finite

Table 7.5: Comparison of the execution times of the evaluation of the Buckling constraint by COMREL [Comrel 1999] and ADAPRES.

	COMREL [Comrel 1999]	ADAPRES
<i>Average number of design optimisation iterations</i>	74	74
<i>Average number of evaluations of the performance function</i>	19	8
<i>Approximate execution time of a FEA Buckling analysis (seconds)</i>	17	17
<i>Total time for the evaluation of the Buckling constrain (seconds)t</i>	23902	10064
<i>reduction in computational time</i>	57.27%	

difference, and so on, the value of the design variable for each iteration cannot be controlled, which is determined in the optimisation program. Therefore, some fluctuation in the response of the constrain functions can be observed during the optimisation iteration as can be seen in Figure 7.9 and 7.10, because CFSQP [CFSQP 1998] checks the response of the constraint function for minimum values of

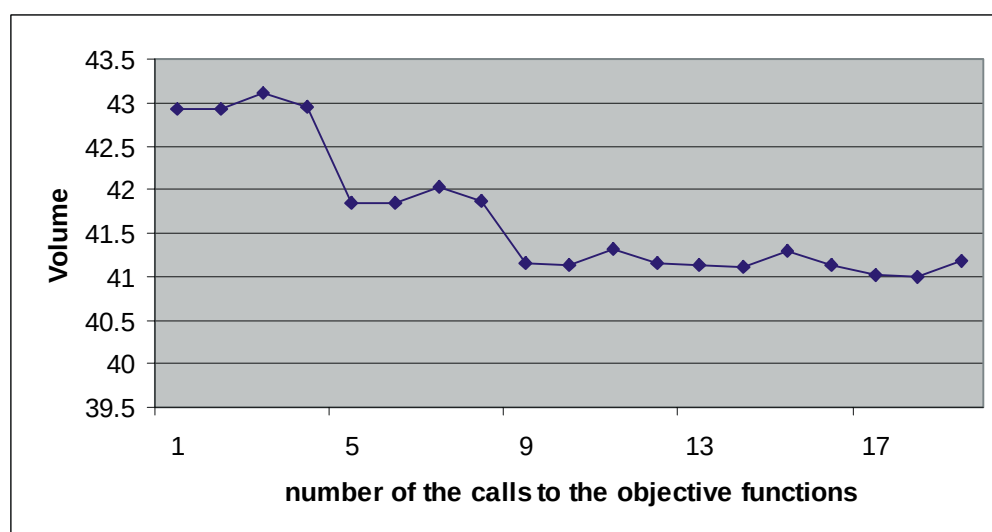


Figure 7.8: Changes in the volume during the design optimisation.

the design variables around the 15th iteration. However, carrying out a few iteration for the minimum values of the design variables, it recovers and searches around the values that give results very close to satisfying the constraints.

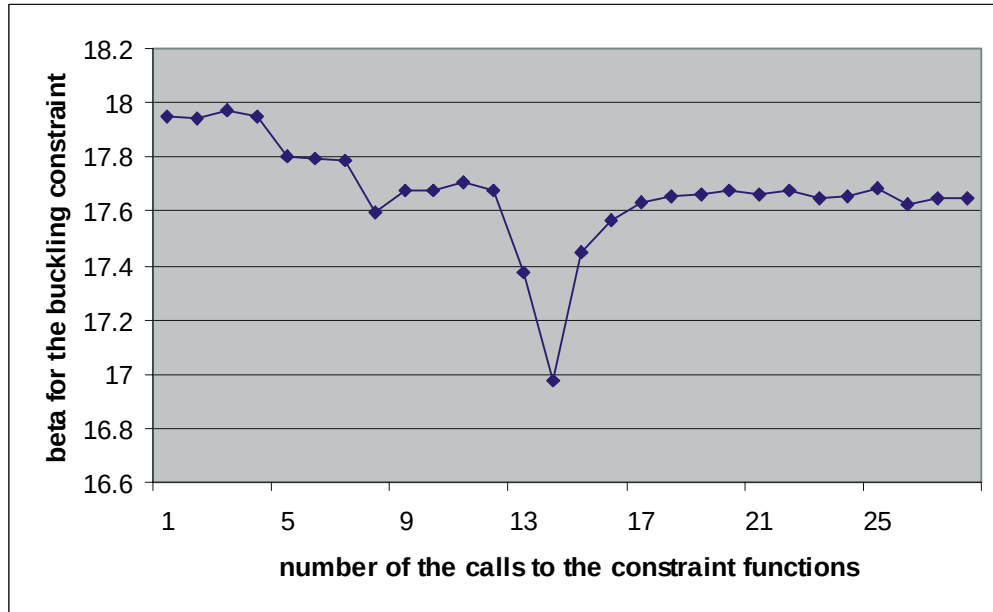


Figure 7.9: Changes in the value of the reliability index for the buckling constraint in Eq.(7.1).

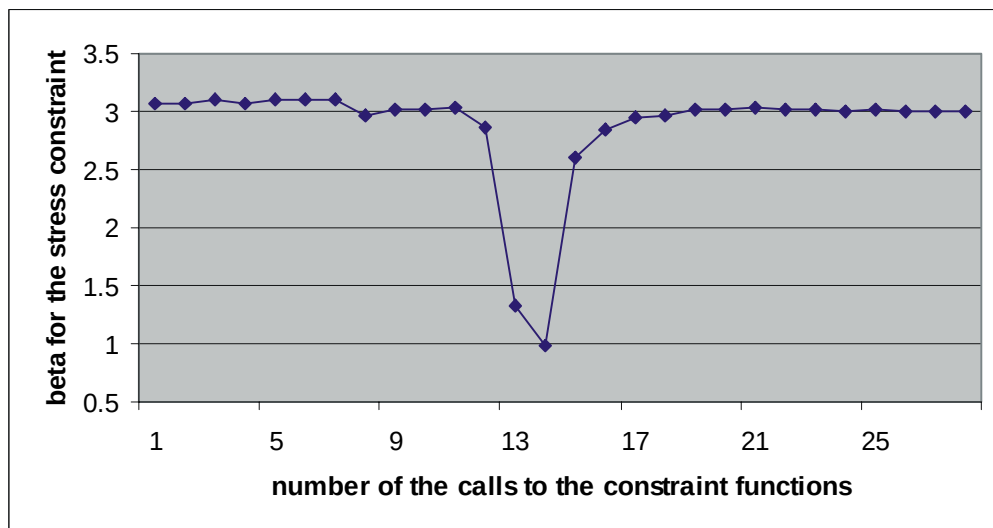


Figure 7.10: Changes in the value of the reliability index for the stress constraint in Eq.(7.1).

7.7 Conclusion

In this chapter, ADAPRES and ADAPRES_NET were applied to a case study involving a reliability based optimisation of a connecting rod for which two probabilistic constraints are set, one of which is based on the maximum Von Mises stress, and the other is the critical load for a linear buckling analysis

Since a linear analysis was used for both structural analyses; i.e. stress analysis and buckling analysis, ADAPRES yielded a linear RSF. It must be noted that real evaluation of the connecting rod should be done by means of applying nonlinear stress and buckling analyses from which the responses could be nonlinear for the input used to form the RSFs. In such cases, the evaluation of probabilistic constraints by applying conventional structural reliability methods takes considerably more time than that of the evaluation of a linear performance function as in this case study. Thus, ADAPRES can reduce the computational time for the evaluation of probabilistic constraints significantly as shown in this case study.

Chapter 8

Concluding Remarks and Future Works

The aim of this chapter is to provide an overall summary of the thesis. It starts with a restatement of the objectives of the work, and a brief review of the main approaches that were taken. It then gives a summary of the conclusions that can be drawn from the work and the limitations of the methods that have been developed. Future directions for the research following on from the thesis are presented. Finally, the publications resulting from this work are listed.

8.1 Development in the Present Thesis

The primary objectives of the present work were given as follows:

1. To develop a computationally efficient as well as a more accurate approximation model to the real performance function in order to use with structural reliability methods such as First Order Method, Second Order Method or Monte Carlo simulation.
2. To develop a probabilistic design system in which the evaluation of multiple performance functions can be distributed over a computer network and called concurrently in order to reduce the time and computational burden of structural reliability analysis.

The first objective has been accomplished via the development of an ADAPtive RESponse surface method, called ADAPRES, which is based on fitting a polynomial function to the responses of the performance function for structural reliability analysis. Firstly, a DoE method was developed by which the design matrix is formed with experimental points that yield the performance function closer to the limit state. Secondly, both linear and non-linear regression methods were combined into ADAPRES to obtain the coefficients of the RSF. Finally, a modelling error assessment was combined into ADAPRES to assess whether the formed RSF is a good approximation to the performance function. The formation of the RSF in ADAPRES is carried out dynamically by starting from a linear type of RSF and higher-order terms are added according to the modelling error assessment. In order to obtain the terms in a RSF, an indicator matrix was suggested in which a distinct area is assigned for terms having the same degree. Two examples were given in that chapter, showing the application of ADAPRES in which better results were obtained than by using a conventional response method.

To accomplish the second objective, the adaptive response surface method has been incorporated into a “Tool Management Systems”, which the authors’ research group has developed to support both the control and sequencing of computing processes using a network model and the execution of distributed processes on a computing network, called ADAPRES_NET. Further improvements have been done in this work mainly in two areas:

1. Data send/receive has been automated so that only the MAIN_CONTROLLER invokes the necessary programs such as SendData or ReceiveData in both the Server side and Client side of the local network.
2. The programs on the workstations can be remotely invoked by the MAIN_CONTROLLER which enables automation of the calls to those programs on the workstations.

In this program, the evaluation of multiple performance functions of a structural reliability problem using response surface functions or direct calls to the performance function can be distributed over the workstations on a network. In addition, the evaluation of the response surfaces approximating the different performance functions can also be carried out concurrently on the network.

8.2 Summary of Conclusions

The following conclusions can be drawn from the present thesis:

- The DoE approach developed in this study can replace the experimental points closer to the limit state by which a better RSF can be obtained for structural reliability analysis since the performance function is approximated around the design point.
- ADAPRES can form a linear RSF for the performance functions behaving linearly. Thus, the number of the iterations required to form a RSF can be reduced, and a better result than standard fixed-order approaches can be obtained with fewer experimental points, as shown in Example 1 in Chapter 5.
- For nonlinear performance functions, ADAPRES can form a RSF which starts with linear RSF, and adds higher-order terms until modelling error criteria are satisfied. Unlike the case where a fixed type of nonlinear RSF such as a second-order polynomial function is assumed, ADAPRES is capable of fitting higher-order polynomial functions to the responses of the performance function, with improved fitting of the response, as shown in Example 3 in Chapter 5.
- ADAPRES introduced a modelling error assessment to the RSM for structural reliability analysis, by which it can be determined that the formed RSF is a

good approximation to the responses of the performance function. It also guides ADAPRES in adding higher terms to the RSF, which results in a better approximation to the performance function.

- ADAPRES_NET is capable of distributing the evaluations of the performance functions on computing workstations over a network.
- ADAPRES_NET can run multiple response surface approximations to performance functions concurrently over a computer network, as was applied for the case study in Chapter 7. This results in a significant reduction in computational time of the evaluation of the performance function.
- ADAPRES_NET can handle data exchange among different process of the problem.

8.3 Limitations

As described in earlier chapters, the work carried out during the present study involves many methods in three different areas: methods used in structural reliability analysis, response surface methods and network programming. Therefore, the limitations of the developed work in the this study are given in those three fields in the next sections.

8.3.1 Limitations of Structural Reliability Analysis

ADAPRES makes use of some structural reliability methods such as FORM and MCS. Therefore, limitations imposed by these structural reliability methods affect ADAPRES, for which the principal limitations are given as follows:

- In ADAPRES, the design point value is obtained by using conventional methods FORM and MCS, in which an optimisation problem is set up to find the design point. Therefore, it suffers from the same limitations of any optimisation algorithm: the convergence problem and the number of iterations required to reach an optimum value.
- Random variables were assumed to have normal distribution in the thesis. Although some of the transformation methods that were given in Chapter 3 can be used for other types of random variables, this might be a problem because

all of the transformations have some level deficiencies and limitations.

- Although computing facilities such as processor speed, multiple processors machine and memory capacity have been increased dramatically, MCS still takes a significant time even to form the response surface function.

8.3.2 Limitations of ADAPRES

The primary limitations regarding ADAPRES can be explained in the following points:

- One of the main problems regarding any response surface method is to determine the minimum number of experiments required, for which there is no clear criterion in literature, except some suggestions for specific problems [Isukapalli 1999]. Although ADAPRES makes use of an approach in which the number of experiments is determined to be one more than the number of the terms in the response surface function to be formed, this approach must be taken as the least condition for forming a response surface function. It should be recalled that the more experimental points are used the more accurate the approximation that can be obtained.
- Selection of the experimental points in ADAPRES is based on the comparison of the results from the performance function, and experimental points yielding a response close to zero are included in the design matrix. This approach only works if the response for a random variable is globally monotonic.
- Modelling error assessment given in ADAPRES were originally developed for physical experiments that are quite different to computer experiments as pointed out in Chapter 5. Therefore, the modelling error assessment is given all the experimental points used to form the response surface function. However, the computation of the probability of failure by means of structural reliability methods only takes into account the points falling in the region of interest close to the Limit State. As shown in the second example given in Chapter 5, although the overall modelling error of the response surface function is in an acceptable range, the response surface function cannot approximate the performance function around the design point.

8.3.3 Limitations of ADAPRES_NET

Since ADAPRES_NET is based on network programming on UNIX systems, most of the limitations are related to the programming approach to networking, which are summarised as below:

- To set up ADAPRES_NET for a structural reliability problem, the user needs to input the problem parameters directly into the data file for which he/she must know the structure of the data files.
- Although Petri nets makes it easier to design the interactions and dependencies of the analyses carried out in structural reliability analysis, this task still needs to be carefully planned in terms of interactions and dependencies.
- Communication is based on file transfer rather than object or function calling, which makes the program slow as well as requiring extra work to send /receive the required data.
- The driver used to communicate with external programs was only developed for ANSYS. For other common analyses packages, special drivers must be developed in order to incorporate them with ADAPRES_NET.

8.4 Future Work

Similar to the limitations given in the previous sections, suggestions for future work are explained in the context of three areas as follows:

8.4.1 Future Work on the Structural Reliability Methods

- The method proposed in Chapter 3, based on using MCS to find an equivalent design point corresponding to the obtained POF needs to be improved by rearranging the optimisation problem used to find the minimum value of the reliability index.
- Other common types of probability distribution function, such as Lognormal or Weibull distribution, should be implemented in order to use the developed methods for a wider range of engineering applications.
- More advanced techniques such as the SORM should be implemented in order to overcome the deficiencies that arise from the use of the FORM to find the design point in ADAPRES.

- A general optimisation program CFSQP is utilised to find the design point. The features of a structural reliability analysis could be implemented in that optimisation problem to make it faster, for example arranging the boundaries of the design variables or adding extra constraints to reduce the sampling region for the search for the value of design point.

8.4.2 Future Works on ADAPRES

- Modelling error assessment used in ADAPRES is based on a general assessment that was developed mostly for statistical applications. The feature of a structural reliability analysis could be implemented such as assessing only the experimental points close the limit state. Leaving out other points might give better estimate for the reliability result since only the points that will contribute the computation of the probability failure are taken into account, which is already proposed in a conference paper by Kaymaz and McMahon [Kaymaz 1999].
- The relationship between the number of the experiments and the number of the terms in a response surface function is adjusted as the latter has one less than the former. Although this approach works for the example given in Chapter 5, as stated earlier, more points are needed to form a higher polynomial. Therefore, this relationship must be formed in a better way to avoid bias caused by using inadequate experimental points.
- The selection of the experimental points is based on a comparison of the result of the upper limit and lower limit of random variables. As was pointed out in Chapter 5, this approach only works if the effect of the random variables is globally monotonic. For any local highly nonlinear response, the selection of the experimental points becomes poor. Therefore more comprehensive assessment for the effect of a random variable should be implemented. One of the possible solutions is to use a sensitivity analysis.

8.4.3 Future Works on ADAPRES_NET

- Distributing the applications over the network needs careful planning for interactions and communication among the applications. Although using Petri

nets can alleviate these difficulties, preparation of the Petri net for a general structural reliability problem requires extra work because the user needs to interact with the text data file as well as to determine the interaction between transitions and places. Therefore, a user friendly interface for data input for Petri nets would make the implementation of ADAPRES_NET easy for a structural reliability problem.

- External calls to other programs are carried by means of special programs for each program as was explained for ANSYS in Chapter 7. It requires having special driver programs by which the communication between ADAPRES_NET and the external programs can be established. However, as an emerging standard in distributed applications, CORBA (Common Object Request Broker Architecture) could be used in order to call external CORBA-compliant programs with a reduced effort in the preparation of special drivers and other adaptations.

8.5 Publications Arising from Work Reported in This Thesis

- Kaymaz, I., McMahon C.A., "An approach to reliability analysis using the response surface method and Monte Carlo simulation", ASME 13th Reliability, Stress Analysis and Failure Prevention Conference, RSAFP-8846 (CD-ROM publication), Las Vegas on 15-17 September 1999.
- Kaymaz, I., McMahon C.A., Xianyi, M., "Reliability based structural optimisation using the response surface method and Monte Carlo simulation", UMTIK'98 8th International Machine Design and Production Conference, September 9-11, pp. 369-378, Ankara, Turkey
- Kaymaz, I., McMahon C.A., "An adaptive response surface method for reliability analysis problems", *submitted to Structural Safety*, 2000.

APPENDIX A

The Process Controller in the Petri Net Tool Manager

The process controller in the Petri Net based Tool Manager simulates the firing mechanism of a Petri net, and uses this to co-ordinate the interaction between the computing processes defined in the design process. It also automatically manages the sequential and concurrent execution of the different processes. The basic elements and tasks of the process controller are as follows:

- Matrices containing information on the dependencies of the transitions and the places in the Petri net.
- A mechanism to initiate child processes for each transition in the Petri net.
- Interprocess communication mechanisms to signal between the main process and child process when transitions are enabled and when they have completed.
- Locking and unlocking mechanisms to control access to the shared files.

The process controller carries out the tasks by maintaining the following information for the Petri net:

- The current marking of the net
- The current firing sequence of transitions
- The work files associated with each transition

In order to show how a process controller works, the example given in Chapter 2, which is taken from [McMahon 1996] is taken, which is again shown in Figure A.1. The matrices include the information describing the Petri net. The first task for the controller is to prepare the five matrices, which describe the dependencies between

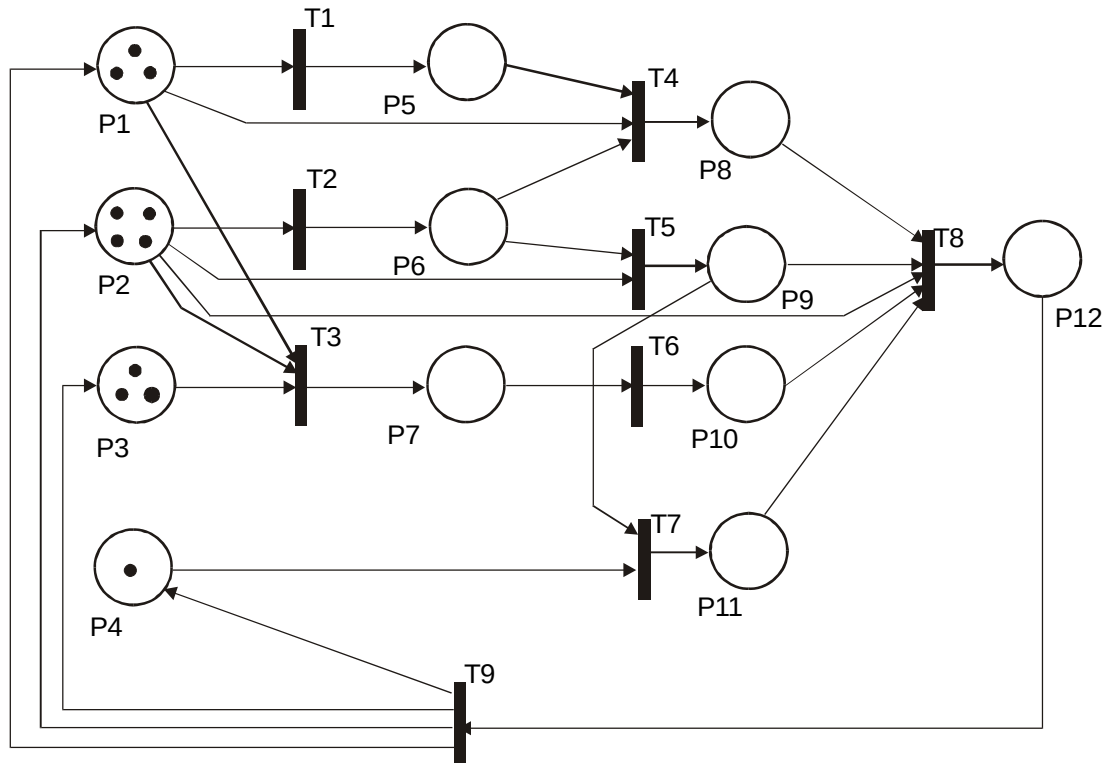


Figure A.1: Petri nets for the optimisation of Encastre Beam adapted from McMahon and Meng [McMahon 1996] .

The dependency matrices: Input (D-) and Output (D+)

i/j	1	2	3	4	5	6	7	8	9	10	11	12	i/j	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	
2	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	0	
3	1	1	1	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	1	0	0	0	0	
4	1	0	0	0	1	1	0	0	0	0	0	0	4	0	0	0	0	0	0	0	2	0	0	0	
5	0	1	1	0	1	1	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	2	0	0	
6	0	0	1	0	0	0	1	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	1	0	
7	0	0	0	1	0	0	0	1	1	0	0	0	7	0	0	0	0	0	0	0	0	0	0	1	
8	0	1	0	0	0	0	0	1	1	1	1	0	8	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	1	9	0	0	0	0	0	0	0	0	0	0	0	

(a)

(b)

Figure A.2: Input (a) and output dependency matrix (b).

Table A.1: Explanation of the transitions and places in Figure A.1.

Transitions	Task
T ₁	calculation of the inertia moment
T ₂	calculation of the moment
T ₃	calculation of the weight
T ₄	calculation of the stress
T ₅	calculation of the deflection
T ₆	calculation of the cost
T ₇	calculation of the constraint functions
T ₈	collecting the results
T ₉	assessment/iteration control

Places	Task
P ₁	beam dimensions
P ₂	problem parameters
P ₃	material parameters
P ₄	Limits
P ₅	inertia moment
P ₆	Moment
P ₇	Weight
P ₈	Stress
P ₉	Deflection
P ₁₀	Cost
P ₁₁	constraint functions
P ₁₂	Results

the transition and places, which are used for the controller program so as to fire and control the transitions in the correct order. In Figure A.1, i and j indicate the number of transitions and the number of places, respectively. The input dependency matrix, denoted by (D-), indicates which transition will take a token from which place. For example, D₋₃₂ means that transition 3 will take a token from place 2. The output dependency matrix is assigned the name D+, and specifies which transition will put a token in which place. D₊₄₈, for instance, specifies that transition 4 will set tokens at place 8.

Transition dependency matrix: D

The next matrix is the transition dependency matrix, which is significant in terms of the sequence of the firing of the transitions, that is, it indicates which transition will

be fired. It is denoted as D, and an example of it is given below:

<i>i/j</i>	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	1
4	1	1	0	0	0	0	0	0	1
5	1	1	0	0	0	0	0	0	1
6	0	0	1	0	0	0	0	0	1
7	0	0	0	1	1	0	0	0	1
8	0	0	0	1	1	1	1	0	1
9	0	0	0	0	0	0	0	1	0

Figure A.3: Transition dependency matrix, D.

This expression can be made clear by explaining line 8, which includes nonzero values as D_{84} , D_{85} , D_{86} , D_{87} and D_{89} . These mean that transition 8 depends on transitions 4, 5, 6, 7 and 9, and will be executed only after all these transitions are executed.

Extraction matrix and Marking Matrix: Dsub and MARKSV

The extraction matrix, D_{sub} , is used for reducing the marking matrix, MARKSV. The extraction matrix is formed by extracting matrix D^+ from, matrix D^- , for which

<i>i/j</i>	1	2	3	4	5	6	7	8	9	10	11	12
1	-1	0	0	0	2	0	0	0	0	0	0	0
2	0	-1	0	0	0	2	0	0	0	0	0	0
3	-1	-1	-1	0	0	0	1	0	0	0	0	0
4	-1	0	0	0	-1	-1	0	2	0	0	0	0
5	0	-1	-1	-1	-1	0	0	0	2	0	0	0
6	0	0	-1	0	0	0	-1	0	0	1	0	0
7	0	0	0	-1	0	0	0	-1	-1	0	1	0
8	0	-1	0	0	0	0	0	-1	-1	-1	-1	1
9	3	4	3	1	0	0	0	0	0	0	0	-1

Figure A.4: Extraction matrix for the Petri net in Figure A.1.

an example is given in Figure A.4 for the Petri net in Figure A.1. MARKSV includes the initial marking that shows the number of the tokens in the places. In this example, the matrix initially contains the vector given as follows:

3 4 3 1 0 0 0 0 0 0 0 0

Figure A.5: Initial marking in MARKSV.

After firing a transition, the controller program will update matrix MARKSV by subtracting matrix D_{sub} . When transition 1 fires, the updated MARKSV will be as below:

2 4 3 1 2 0 0 0 0 0 0 0

Figure A.6: After firing T1 in Figure A.1.

The controller program scans the matrix D - and vector MARKSV in order to determine if transitions are enabled, which is necessary to execute the transitions. It must be noted that all the matrices, except D_{sub} , given above are prepared in advance by the user.

APPENDIX B

The Structure of ADAPRES

In order to use the proposed ADAPRES method, a computer program called `AdapRes_main.c` was written that incorporates the algorithm given in Figure 5.1, and theoretical information was given in Chapter 5. In this section, however, setting up ADAPRES is explained in terms of preparing the necessary data files. The programming structure of ADAPRES is also represented by giving an explanation of some of the functions inside the main C program `AdapRes_main`, as shown in Figure B.2.

B.1 General Structure of ADAPRES

ADAPRES is formed by many subprograms for which sub folders are formed for each separate program as shown in Figure B.1. The main program is `AdapRes_main.c` in which the algorithm of ADAPRES given in Figure 5.1 was programmed using the C programming language. The data files that are shared with other subprograms are placed in a folder called `DATA_FILES`, and reference is made to them in the header file

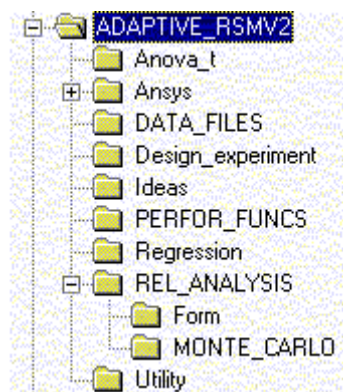


Figure B.1: The file structure of ADAPRES.

AdapRes_main.h, by which method access to the shared files becomes easy for the programs in the different directories. ADAPRES also includes some calls to external programs such as the ANSYS software or the I-DEAS suite, for which purpose specific files and data files are replaced in a separate directory such as Ansys in Figure B.1.

B.2 Preparing ADAPRES for a Structural Reliability Problem

To run ADAPRES for a structural reliability problem, the following main preparations must be done:

- Prepare the data file: prob_param.dat
- Input the performance function in the file: perfor_func.c

ADAPRES requires one input data file, prob_param.dat, in which all the data related the structural reliability problem exists. An example of this data file is shown in Table B.1 ,and brief information on each data item represented in the table is given in the following.

The first data item in Table B.1 shows the number, n , of random variables in the structural reliability problem, and the following n lines then give the mean values and standard deviations for each of the random variables. The values for $f_{\min}$ and $f_{\max}$, which represents the minimum and maximum value of the arbitrary factor in Eq. (4.7) are input in the next line so that the user can determine the boundary of the design domain from which the experimental points are selected.

In ADAPRES, three different DoE methods can be used to form the design matrix including the experimental points, assigned as:

- 1 : *Bucher DoE*
- 2 : *Central Composite Design (CCD)*
- 3 : *AdapResDesign*

This choice is given in the next line of the file. The criteria for selecting the linear type of RSFs and general type of RSFs, which could be in any polynomial form, are then input, denoted as crit_lin and crit_gen, respectively. The value of crit_perc is input to determine how close the formed RSF is to the limit state, which is compared with perc_dif in Eq. (5.9). The next input in the data file is the number of extra points which

Table B.1: A sample for prob_param.dat.

No. of random variables:	2
Mean value and standard deviation:	10 5
Mean value and standard deviation:	9.9 5
$f_{\min}$ and $f_{\max}$:	2.5 2.5
type of DoE:	3
crit_lin:	0.95
crit_gen:	0.9
crit_perc:	1
number of extra points:	1
type of RSM:	2
Method for design point:	1

shows the number of the extra experimental points that are added in the case that the RSFs generated from the existing design matrix does not satisfy both criteria; namely crit_lin and crit_gen.

ADAPRES can easily be converted to other RSMs by the input, type of RSM in the data file, which is assigned as follows:

- 1: *Bucher RSM*
- 2: ADAPRES

There are two different methods in ADAPRES to find the design point, assigned in prob_param.dat as follows:

- 1: *FORM*
- 2: *MCS*

Thus, different RSMs can be experimented with for the structural reliability problem as well as applying two different methods for one of the most important parameters of structural reliability analysis.

Another file that is needed to be prepared is perfor_func.c, which includes the information concerning the performance function. In ADAPRES, both explicit and implicit performance functions can be input in that file. An explicit performance function can be easily input in C program format. On the other hand, an implicit

performance function requires extra work, depending on the program that is used to obtain the result of the performance function for a given set of the random variables. When any FEA programs such as ANSYS, ADINA or similar are used to get the response of the stress analysis required by the performance function, for instance, program files are needed to be prepared, which are called **drivers** in ADAPRES. For the two program packages mainly used, ANSYS and I-DEAS, there are pre-programmed drivers that include necessary data exchange, and the program files used to invoke the particular external program. In order to call ANSYS, for example, the driver program called AnsysMain.c was prepared in which the parameters are saved in data files such as CR_ANSYS.dat and CR_random.dat for the case problem given in chapter 7. The format of those data file was also arranged according to the format of the input file of ANSYS. AnsysMain.c also includes a system call by which ANSYS is invoked in batch mode with the program file prepared for the case problem which is given in Appendix C.

B.3 Brief Explanation of Each of the Calls to the Functions

A brief explanation is given on some of the most important functions calls in AdapRes_main.c which is illustrated in Figure B.2

B.3.1 design_matrix

This function call in AdapRes_main.c forms the experimental points according to the DoE method input in the file prob_param.dat. At the moment, ADAPRES is capable of forming the design matrix for three DoE methods, which are Bucher DoE, Central Composite Design and the AdapResDesign method that has been developed during the study for which a more detailed explanation is represented in Chapter 5.

A user defined DoE can also be input in this function provided that the syntax of the function is written in C programming language.

B.3.2 perfor_func

As explained in the previous section, the performance function is input in this program, by using two functions: perfor_func() and perfor_func_single (). Although both these functions include the performance function, perfor_func () is called for the

design matrix including the experimental points for which the output from performance function is obtained as a vector while `perfor_func_single ( )` results in a single performance function value for one set of the experimental points. The latter, for example, is required when the output of the performance function is obtained for the value of the design point.

B.3.3 form_RSF

This function call involves forming the RSF according to the chosen RSM method, which is specified by the type of RSM in `prob_param.dat`. For ADAPRES, firstly a linear RSF function is formed, and checked against the criterion, `crit_lin`, in `prob_param.dat`. If the criterion is not satisfied, ADAPRES checks the indicator matrix by calling the function, `poss_terms( )`, which includes the information on the possible terms that can be added to the linear RSF formed in the previous iteration. The formation of the RSF is carried out by calling a function, `AdapRes( )` in which another function, `coeff_func( )`, is called to get the coefficients of the RSF by means of the programs obtained from [Press 1992].

As explained in Chapter 5, modelling error of the RSF is assessed by means of the Adj_R^2 method given in Eq. (5.4) whose value is computed by calling a function, `Anova( )` which consists of multiple programs that are adapted and modified from [decdfilib 1997].

B.3.4 call_FORM

This function is called to obtain the design point that is used both for assessing the RSF if it is close to the limit state, and for forming the design matrix for the second stage of the RSM, explained in Chapter 5. Since the computation of the design point consists of rather different computation than forming a RSF in ADAPRES, the method that is used to find the design point is called by a system command, `system("FORM")`. In ADAPRES, however, two different approaches can be utilised which are the FORM and MCS, for which necessary programs such as `beta_ADAPRES.c` for the FORM and `monte_carlo.c` for MCS are also replaced in the related directories of these methods as shown in Figure B.1

B.3.5 updateDesignMatrix

In the case that the design matrix is not sufficient to form a RSF that can satisfy both the criteria explained in the previous sections, the design matrix and the performance matrix are updated to include additional experimental points by calling this function. This function first computes the additional points by considering the effect of the random variables on the performance function, which is determined according to the approach explained in Chapter 5. In this function, `perfor_func_single()` is called to get the response of the performance function for the extra point(s) added.

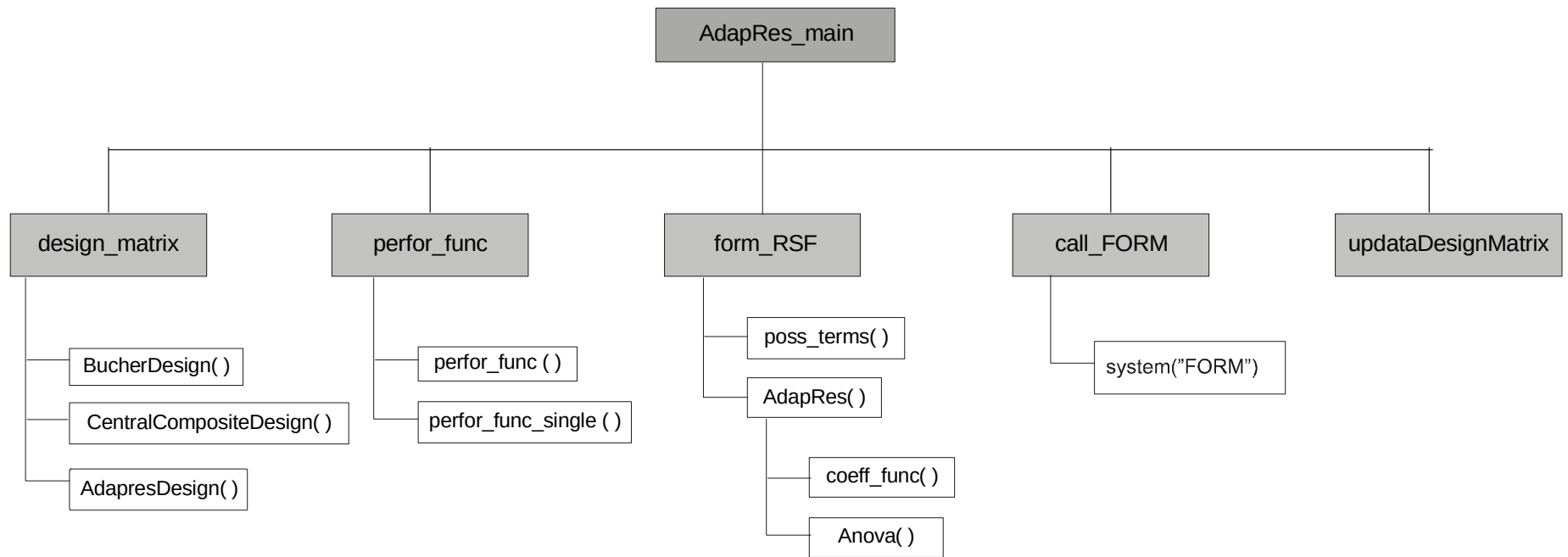


Figure B.2: Some of the functions in AdapRes_main.c.

APPENDIX C

APDL Files for the Buckling Analysis and Stress Analysis

The ANSYS Parametric Design Language (APDL) is used to create a parametric model of the connecting rod for both the linear buckling analysis and the stress analysis. The program code for both analyses is given in the next sections. The extraction of the critical buckling load and the maximum stress is in each case given near the end of the file.

C.1 APDL file for the linear Buckling Analysis

```
/CLEAR
```

```
! *****
!
!      Stress Analysis of Connecting Rod for
!
!      Reliability Based Structural Optimisation
!
!      Modified by: Irfan Kaymaz
!
!      Modification Date: 28 August 2000
! *****

!*****

! In the following several parameters are read from datafiles,
! CR_ANSYS.dat , CR_random.dat, des_var_const.dat
*****
```

```

/input,CR_ANSYS,dat

/input,CR_random,dat ! as the random variables changes in each iteration

/input,des_var_const,dat

/prep7

MP,EX,1,CR_EMOD      ! material properties of connecting rod

MP,NUXY,1,CR_NUXY

! *****

! In the following the necessary keypoints for the geometry of

! the model are calculated and set

! *****

div10=div7-div9      ! line division calculation

div12=div6-9

elminRb=l-Rb

elpluRb=l+Rb

elpluRbi=l+Rbi

elminRbi=l-Rbi

K,1,0,0      ! creation of keypoints

K,2,Rai,0

K,3,Ra,0

K,4,elminRb,0

K,5,l,0,

K,6,elpluRb,0

K,7,elpluRbi,0

K,21,elminRbi,0

*AFUN,DEG      ! units for angular functions set to degrees

xk8=SQRT((Ra+Rl)**2-(Rl+w)**2)

```

yk8=Rl+w

k,8,xk8,yk8

alpha=ANGLEK(1,2,8) ! angle is calculated of arc from KP1 to KP2

! with center KP8

xk9=xk8+Rl*COS(90-gamma)

yk9=w+Rl*(1-SIN(90-gamma))

K,9,xk9,yk9

xk10=xk9-(yk9-w)/tan(gamma)

yk10=w

k=(Rb+Rr)**2

m=tan(gamma)

a=(Rr/cos(gamma))+w+(l-xk10)*m

xk11=l-(m*a)/(1+m**2)-SQRT((((m*a)/(1+m**2))**2-(a**2-k)/(1+m**2))

yk11=(xk11-l)*m+a

k,11,xk11,yk11

xk12=xk11+Rr*sin(gamma)

yk12=yk11-Rr*cos(gamma)

K,12,xk12,yk12

beta=ANGLEK(5,4,11)

yk13=(Rb*yk11)/(Rb+Rr)

xk13=l-yk13/tan(beta)

K,13,xk13,yk13

xk14=l-Rbi*cos(beta)

yk14=Rbi*sin(beta)

K,14,xk14,yk14

u=(Rb+Rwr)**2

```
b=w+(l-xk10)*m-(s+Rwr)/cos(gamma)

xk15=l-(m*b)/(1+m**2)-SQRT(((m*b)/(1+m**2))**2-(b**2-u)/(1+m**2))

yk15=(xk15-l)*m+b

K,15,xk15,yk15

xk16=xk15-Rwr*sin(gamma)

yk16=yk15+Rwr*cos(gamma)

K,16,xk16,yk16

betax=ANGLEK(5,15,4)

xk17=l-Rb*cos(betax)

yk17=Rb*sin(betax)

K,17,xk17,yk17

e=(Ra+Rwl)**2

d=Ra+(w-xk10*m)-(s+Rwl)/cos(gamma)

f=d-Ra

xk18=-(m*f)/(1+m**2)+SQRT(((m*f**2)/(1+m**2))**2-(f**2-e)/(1+m**2))

yk18=xk18*m+d-Ra

K,18,xk18,yk18

xk19=xk18-Rwl*sin(gamma)

yk19=yk18+Rwl*cos(gamma)

K,20,xk19,yk19

alphax=ANGLEK(1,3,18)

yk20=Ra*sin(alphax)

xk20=yk20/tan(alphax)

K,19,xk20,yk20

K,1000,0,Rai,0
```

```

! *****

! In the following lines are created which make up the contour of the connecting rod

! *****

CYL4,,,Ra,alpha,Rai,180    ! cylinder on the left, lines 1,2,3,4

LARC,2,24,1,Rai           ! line/arc from KP2 to KP24 with radius

                           ! Rai and centre KP1  lines 5

LARC,21,14,5,Rbi         ! line 6

LARC,13,6,5,Rb           ! line 7

LARC,14,7,5,Rbi          ! line 8

LARC,12,13,11,Rr         ! line 9

LARC,10,9,8,Rl           ! line 10

L,20,16                   ! line 11

L,16,14                   ! line 12

L,2,3                     ! line 13

L,3,4                     ! line 14

L,4,21                    ! line 15

L,6,7                     ! line 16

L,9,12                    ! line 17

L,13,14                   ! line 18

LDIV,5,0.5,,2,1          ! lines L20,L19

L,25,20                   ! line L21

L,9,20                    ! line L22

L,12,16                   ! line L23

LDIV,3,0,1000,2,1        ! line L24 and L25

LESIZE,9,,,5

LESIZE,10,,,5

```

```

LESIZE,19,,,5
LESIZE,20,,,5
LESIZE,24,,,5
AL,1,2,24,25,4
AL,4,10,22,21,20
AL,22,17,23,11
AL,18,9,12,23
AL,19,13,14,15,6,12,11,21
AL,16,7,18,8
ALLSEL,ALL,ALL
arsym,y,all
ET,1,plane82,,,3
R,1,zet1
MSHAPE,1
AMESH,ALL

! *****

! In the following the tolerance for the NUMMRG command is calculated,

! *****

omega=ANGLEK(8,26,9)           ! precinct of fillet - small end
length=2*3.141*Rl*omega/360     ! the length of the arc is calculated
thi1=length/div11              ! thi1 is the element length of the arc
TOLER=0.1*thi1                 ! NOTE: the tolerance should be one tenth of
                                ! the smallest element length or linedivision
NUMMRG,NODE,TOLER              ! merges all nodes of the model together
NUMMRG,ELEM,TOLER
EPLOT

```

! *****

! In the following the pressure and the displacement constraints are applied

! *****

PHI=90 ! angle of curved surface is 90 degrees

Fmax=1

decrea= -((Fmax/90)) ! calculation of linear decrease

K,1001,0,0 ! origin of cylindrical CS

CSKP,12,CYLIN,1001,2,1000 ! create cylindrical CS (nbr. 12) with origin in
! KP 1 and reference KP 2 for x-axis, KP 1000
! for y-axis

CSCIR,12,0 ! set the singularity of CS 12 to 180 degrees (0)

LSEL,S,LINE,,24,,

LSEL,A,LINE,,20,,

LSEL,A,LINE,,19,,

NSLL,S,1 ! selects ALL (1) nodes attached to
! selected areas

SFGRAD,PRESS,12,y,0,decrea ! pressure with the defined decrease,
! starting at the angle of 0 degrees

SF,ALL,PRESS,Fmax ! pressure with starting force Fmax
! is applied on all selected nodes bottom half

Fmax=1

decrea= -((Fmax/90)) ! calculation of linear decrease

!K,1001,0,0 ! origin of cylindrical CS

CSKP,13,CYLIN,1001,2,30 ! create cylindrical CS (nbr. 13) with origin in
! KP 1 and reference KP 2 for x-axis, KP 1000
for y-axis

CSCIR,13,0 ! set the singularity of CS 13 to 180 degrees (0)

```

LSEL,S,LINE,,31,,
LSEL,A,LINE,,32,,
LSEL,A,LINE,,42,,
NSLL,S,1                ! selects ALL (1) nodes attached to selected areas
SFGRAD,PRESS,13,y,0,decrea    ! pressure with the defined decrease,
                                ! starting at the angle of 0 degrees
SF,ALL,PRESS,Fmax          ! pressure with starting force Fmax
                                ! is applied on all selected
ALLSEL,ALL,ALL            ! select everything

! *****
! In the following specified nodes are coupled for displacement constraints
! *****

CSYS,0                    ! go back to Cartesian CS, needed for entity selection
LSEL,S,LINE,,8
LSEL,A,LINE,,6
LSEL,A,LINE,,43
LSEL,A,LINE,,48
NSLL,S,1
D,ALL,ALL
ALLSEL,ALL,ALL
EPLOT
FINISH

! *****
! In the following static analysis is carried out.
! *****

/SOLU

```

ANTYPE,STATIC

PSTRES,ON

EPLOT

SOLVE

FINISH

! *****

! In the following the commands required for Eigenvalue Buckling analysis are given

! *****

/SOLU

ANTYPE, BUCKLE

BUCOPT,LANB,1

MXPAND,ALL

SOLVE

*GET,FCR,MODE,1,FREQ ! getting the critical buckling load, FCR

/OUTPUT, CR_ANSYS.res

*VWRITE,FCR ! writing the value of FCR

(E10.4)

/OUTPUT

FINIISH

C.2 ADPDL file for stress analysis of the connecting rod

```
/CLEAR
```

```
! *****
```

```
! Stress Analysis of Connecting Rod for
```

```
! Reliability Based structural Optimisation
```

```
! Modified by: Irfan Kaymaz
```

```
! Modification Date: 28 August 2000
```

```
! *****
```

```
! *****
```

```
! In the following several parameters are read from datafiles,
```

```
! CR_ANSYS.dat , CR_random.dat, des_var_const.dat
```

```
! *****
```

```
/input,CR_ANSYS,dat
```

```
/input,CR_random,dat ! as the random variables changes in each iteration
```

```
/input,des_var_const,dat
```

```
/prep7
```

```
MP,EX,1,CR_EMOD ! material properties of connecting rod
```

```
MP,NUXY,1,CR_NUXY
```

```
! *****
```

```
! In the following the necessary keypoints for the geometry of
```

```
! the model are calculated and set
```

```
! *****
```

```
div10=div7-div9 ! line division calculation
```

```
div12=div6-9
```

```

elminRb=l-Rb

elpluRb=l+Rb

elpluRbi=l+Rbi

elminRbi=l-Rbi

K,1,0,0                ! creation of keypoints

K,2,Rai,0

K,3,Ra,0

K,4,elminRb,0

K,5,l,0,

K,6,elpluRb,0

K,7,elpluRbi,0

K,21,elminRbi,0

*AFUN,DEG              ! units for angular functions set to degrees

xk8=SQRT((Ra+Rl)**2-(Rl+w)**2)

yk8=Rl+w

k,8,xk8,yk8

alpha=ANGLEK(1,2,8)    ! angle is calculated of arc from KP1 to KP2

                        ! with center KP8

xk9=xk8+Rl*COS(90-gamma)

yk9=w+Rl*(1-SIN(90-gamma))

K,9,xk9,yk9

xk10=xk9-(yk9-w)/tan(gamma)

yk10=w

k=(Rb+Rr)**2

m=tan(gamma)

a=(Rr/cos(gamma))+w+(l-xk10)*m

```

```
xk11=l-(m*a)/(1+m**2)-SQRT(((m*a)/(1+m**2))**2-(a**2-k)/(1+m**2))
yk11=(xk11-l)*m+a
k,11,xk11,yk11
xk12=xk11+Rr*sin(gamma)
yk12=yk11-Rr*cos(gamma)
K,12,xk12,yk12
beta=ANGLEK(5,4,11)
yk13=(Rb*yk11)/(Rb+Rr)
xk13=l-yk13/tan(beta)
K,13,xk13,yk13
xk14=l-Rbi*cos(beta)
yk14=Rbi*sin(beta)
K,14,xk14,yk14
u=(Rb+Rwr)**2
b=w+(l-xk10)*m-(s+Rwr)/cos(gamma)
xk15=l-(m*b)/(1+m**2)-SQRT(((m*b)/(1+m**2))**2-(b**2-u)/(1+m**2))
yk15=(xk15-l)*m+b
K,15,xk15,yk15
xk16=xk15-Rwr*sin(gamma)
yk16=yk15+Rwr*cos(gamma)
K,16,xk16,yk16
betax=ANGLEK(5,15,4)
xk17=l-Rb*cos(betax)
yk17=Rb*sin(betax)
K,17,xk17,yk17
e=(Ra+Rwl)**2
```

$d = Ra + (w - xk10 * m) - (s + Rwl) / \cos(\gamma)$

$f = d - Ra$

$xk18 = -(m * f) / (1 + m^2) + \text{SQRT}(((m * f^2) / (1 + m^2))^2 - (f^2 - e) / (1 + m^2))$

$yk18 = xk18 * m + d - Ra$

K,18,xk18,yk18

$xk19 = xk18 - Rwl * \sin(\gamma)$

$yk19 = yk18 + Rwl * \cos(\gamma)$

K,20,xk19,yk19

$\alpha_x = \text{ANGLEK}(1,3,18)$

$yk20 = Ra * \sin(\alpha_x)$

$xk20 = yk20 / \tan(\alpha_x)$

K,19,xk20,yk20

K,1000,0,Rai,0

! *****

! In the following lines are created which make up the contour of the connecting rod

! *****

CYL4,,,Ra,alpha,Rai,180 ! cylinder on the left, lines 1,2,3,4

LARC,2,24,1,Rai ! line/arc from KP2 to KP24 with radius

! Rai and centre KP1 lines 5

LARC,21,14,5,Rbi ! line 6

LARC,13,6,5,Rb ! line 7

LARC,14,7,5,Rbi ! line 8

LARC,12,13,11,Rr ! line 9

LARC,10,9,8,Rl ! line 10

L,20,16 ! line 11

L,16,14 ! line 12

```
L,2,3      ! line 13
L,3,4      ! line 14
L,4,21     ! line 15
L,6,7      ! line 16
L,9,12     ! line 17
L,13,14    ! line 18
LDIV,5,0.5,,2,1    ! lines L20,L19
L,25,20     ! line L21
L,9,20      ! line L22
L,12,16     ! line L23
LDIV,3,0,1000,2,1    ! line L24 and L25
LESIZE,9,,,5
LESIZE,10,,,5
LESIZE,19,,,5
LESIZE,20,,,5
LESIZE,24,,,5
AL,1,2,24,25,4
AL,4,10,22,21,20
AL,22,17,23,11
AL,18,9,12,23
AL,19,13,14,15,6,12,11,21
AL,16,7,18,8
ALLSEL,ALL,ALL
ET,1,plane82,,,,,,1
!KEYOPT,1,3,zet2
```

MSHAPE,1

AMESH,ALL

! *****

! In the following the tolerance for the NUMMRG command is calculated,

!*****

omega=ANGLEK(8,26,9) ! precinct of fillet - small end

length=2*3.141*Rl*omega/360 ! the length of the arc is calculated

thi1=length/div11 ! thi1 is the element length of the arc

TOLER=0.1*thi1 ! NOTE: the tolerance should be one tenth of

! the smallest element length or linedivision

NUMMRG,NODE,TOLER ! merges all nodes of the model together

NUMMRG,ELEM,TOLER

EPLOT

! *****

! In the following the pressure and the displacement constraints are applied

! *****

PHI=90 ! angle of curved surface is 90 degrees

Fmax=FORCE

decrea= -((Fmax/90)) ! calculation of linear decrease

K,1001,0,0 ! origin of cylindrical CS

CSKP,12,CYLIN,1001,2,1000 ! create cylindrical CS (nbr. 12) with origin in

! KP 1 and reference KP 2 for x-axis, KP 1000

!for y-axis

LSEL,S,LINE,,24,,

LSEL,A,LINE,,20,,

LSEL,A,LINE,,19,,

```

NSLL,S,1                ! selects ALL (1) nodes attached to
                        ! selected areas

SFGRAD,PRESS,12,y,0,decrea    ! pressure with the defined decrease,
                                ! starting at the angle of 0 degrees

SF,ALL,PRESS,Fmax           ! pressure with starting force Fmax
                                ! is applied on all selected nodes

ALLSEL,ALL,ALL              ! select everything

! *****

! In the following specified nodes are coupled for displacement constraints

! *****

CSYS,0                      ! go back to Cartesian CS, needed for entity selection

LSEL,S,LOC,Y,0

NSLL,S,1

DSYM,SYMM,Y,0              ! set displacement constraints on nodes y direction due
                            ! to symmetry

ALLSEL,ALL,ALL

LSEL,S,LINE,,8

LSEL,A,LINE,,6

NSLL,S,1

D,ALL,UX,

ALLSEL,ALL,ALL

EPLOT

FINISH

/SOLU

/DSCALE,,1                 ! displacement scale = true displacement (1)

/PBC,ROT,,1

```

```
/PSF,PRES,NORM,2      ! display pressure normal to the surface as arrows
/COLOR,WBAK,BLAC      ! change window background to black (default !)
/PLOPTS,VERS,ON       ! display ANSYS 5.3 logo
/PLOPTS,WINS,OFF      ! switch autosize of window 'off'
SOLVE                 ! solve the current LS
FINISH
/POST1
PLESOL,S,EQV,2        ! plot element solution for Von Mises stress,
NSEL,ALL
NSORT,S,EQV,0,,10     ! a useful command set for the input
*GET,STRSS,SORT,,MAX   !gets the maximum value from the sorted values
! *****
! Saving the maximum stress into a file, CR_ANSYS.res so that ADAPRES
! can use it in the stage of forming a RSF
! *****
/OUTPUT, CR_ANSYS.res
*VWRITE,STRSS
(E10.4)
/OUTPUT
FINIISH
```

References

- [Abet 2000] Abet, the Accreditation Board for Engineering and Technology, <http://www.abet.org>, 2000.
- [AFSC 1981] AFSC, U.S. Air Force Wright Aeronautical Laboratories, “*Integrated computer-aided manufacturing (ICAM) architecture part volume IV - Function modeling manual (IDEF0)*”, AFWAL-TR-81-4023 V.IV, June 1981.
- [Alefeld 1983] Alefeld, G. and Herzberger, J., “*Introduction to interval computations*”, New York: Academic Press, 1983.
- [Anderson 1989] Anderson D.C. and Crawford, R. H., “*Knowledge management for preliminary computer aided mechanical design in: Organisation of engineering knowledge for product modelling in computer integrated manufacturing*”, (ed.) Sata, T., Elsevier, pp. 15-34, 1989.
- [Ang 1975] Ang, A. H. S. and Wilson H. T., “*Probability concepts in engineering planning and design*”, London : John Wiley & Sons, 1975.
- [Ang 1984] Ang, A. H. S and Tang, W. H., “*Probability concepts in engineering planning and design II*”, New York: John Wiley & Sons, 1984.
- [ANSYS 2000] ANSYS Online Manual for Release 5.5, <http://sc.tamu.edu/softwareDocs/ansys/realtoc.html>.
- [Antonsson 1995] Antonsson, E. K. and Otto, K. N., “Imprecision in engineering design”, *Transactions of the ASME Special 50th Anniversary Design Issue*, Vol. 117, pp. 25-31, 1995.
- [ASRANET 2000] ASRANET, Integrating Advanced Structural Analysis with Structural Reliability Analysis, <http://www.asranet.com>, 2000.
- [Austin 1996] Austin, S., Baldwin, A. and Newton, A., “A data flow model to plan and manage the building design process”, *Journal of Engineering Design*, Vol. 7, No. 1, pp. 3-25, 1996.
- [Ayyub 1992] Ayyub, B. M., and Lai, K. L., “Structural reliability assessment with ambiguity and vagueness in failure”, *Naval Engineering Journal*, Vol. 104, No. 3, pp. 21-35, 1992.
- [Barthelemy 1993] Barthelemy, J. F. M. and Haftka, R. T., “Approximation concepts

for optimum structural design – A review”, *Structural Optimisation*, Vol. 5, No. 3, pp. 129-144, 1993.

[Bazaraa 1993] Bazaraa, M. S., “*Nonlinear programming: theory and algorithms*”, New York: John Wiley & Sons, 1993.

[Berke 1993] Berke, L., Patnaik, S. N. and Murthy, P. L. N., “Optimum Design of Aerospace Structural Components Using Neural Networks”, *Computers & Structures*, Vol. 48, pp. 1001-1010, 1993.

[Bjerager 1990] Bjerager, P., “On computation methods for structural reliability analysis”, *Structural Safety*, Vol. 9, pp. 79-96, 1990.

[Booch 1994] Booch, G., “*Object-oriented analysis and design with applications*”, Wokingham: Addison-Wesley, 1994.

[Boresi 1993] Boresi, A. P., Schmidt, R. J. and Sidebottom, O. M., “*Advanced mechanics of materials*”, Fifth Edition, New York: John Wiley & Sons, 1993.

[Borri 1997] Borri, A. and Speranzini, E., “Structural reliability analysis using a standard deterministic finite element code”, *Structural Safety*, Vol. 19, No. 4, pp. 361-382, 1997.

[Box 1951] Box, G. E. P. and Wilson, K. B., “On the experimental attainment of optimum conditions”, *Journal Royal Statistical Society, Series B*, pp. 1-45, 1951.

[Box 1958] Box, G. E. P. and Muller, M. E., “A note on the generation of random normal variates”, *Annals of Mathematical Statistics*, Vol. 29, pp. 610-611, 1958.

[Box 1960] Box, G. E. P. and Behnken, D. W., “Some new three-level designs for the study of quantitative variables”, *Technometrics* 30, 1960, pp. 1-40.

[Box 1987] Box, G. E. P. and Draper, N. R., “*Empirical model-building and response surfaces*”, New York: John Wiley & Sons, 1997.

[Breitung 1983] Breitung, K., “Asymptotic approximations for multinormal integrals”, *Journal of Engineering Mechanics*, Vol. 110, No. 3, pp. 357-366, 1983.

[Brookes 1994] Brookes, N.J., Backhouse, C. J. and Burns, N. D., “Modelling product introduction”, *Advances in Manufacturing Technology VIII*, Tenth National Conference on Manufacturing Research, (eds.) Case, K. and Newman, S., Loughborough University of Technology, ISBN 07484 0254 3, pp 452- 456, 1994.

- [Brooks 1992] Brooks, B. M., “Concurrent engineering – what is it?”, *IEE Colloquium on Concurrent Engineering*, (IEE), pp. 1-3, 1992.
- [Brown 1989] Brown, D., “Minutes of the IDEF-U.G. research issues group”, *Proceedings of the IDEF Users Group Conference*, Scottsdale, Az., May 1989.
- [Bucher 1988] Bucher, C. G., “Adaptive sampling-an iterative fast Monte Carlo procedure”, *Structural Safety*, Vol. 5, No. 2, pp. 119-126, 1988.
- [Bucher 1989] Bucher, C. G. and Bourgand, U., “*Efficient use of response surface methods*”, Report, University of Innsbruck, 1989.
- [Bucher 1990] Bucher, C.G., 1990, “A fast and efficient response surface approach for structural reliability problems”, *Structural Safety*, Vol. 7, pp. 57-66, 1990.
- [Caltech 1997] Caltech, “*Uncertainty management in commercial aircraft design*”, <http://www.cds.caltech.edu/conferences/1997/vecs/tutorial/Examples/Cases/777.htm>, 1997.
- [CFSQP 1998] AEMDesign, <http://gachinese.com/aemdesign/FSQPframe.htm>, 2000.
- [Chandu 1995] Chandu, S. V. L. and Grandi, R. V., “General purpose procedure for reliability based structural optimization under parametric uncertainties,” *Advances in Engineering Software*, Vol. 23, pp. 7-14, 1995.
- [Chang 1999] Chang, K. H., Silva, J. and Bryant, I., “Concurrent design and manufacturing for mechanical systems”, *Concurrent Engineering*, Vol. 7, No. 4, pp. 290-308, 1999.
- [Chajtman 1988] Chatjman, S., and Zyzik, M., “The identification of information processes in manufacturing”, *Towards the Factory of the Future: 8th International Conference on Production Research*, Springer Verlag, Bonn, 1988.
- [Chen 1999] Chen, S., Nikolaidis, E., Cudney, H. H. ,Rosca, R. and Haftka, R. T. “Comparison of probabilistic and fuzzy set methods for designing under uncertainty”, *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibition*, St. Louis, MO, pp. 2860-2874, 1999
- [Chiang 1987] Chiang, W. L. and Dong, W. M., “Dynamic response of structures with uncertain parameters: a comparative study of probabilistic and fuzzy set models”,

Probabilistic Engineering Mechanics, Vol. 2, No. 2, pp. 82-91, 1987.

[Coffin 1990] Coffin, S. “*Unix system V release 4: the complete reference*”, New York: McGraw-Hill, 1990.

[Comrel 1999] COMREL, <http://www.strurel.de>, 1999.

[Choobineh 1992] Choobineh, F. and Behrens, A., “Use of intervals and possibility distributions in economic analysis” *Journal of Opl Res. Soc.*, Vol. 43, No. 9, pp. 907-918, 1992.

[Cornell 1969] Cornell, C.A., “A probability based structural code”, *Journal of the American Concrete Institute*, Vol. 66, pp. 974-985, 1969.

[Cox 1961] Cox, R.T., “*The algebra of probable inference*”, Baltimore: The Johns Hopkins Press, 1961.

[Crossland 1997] Crossland, R., “*Risk in the development of design*”, PhD thesis, University of Bristol, August 1997.

[Curry 1996] Curry, D. A., “*Unix systems programming for SVR4*”, USA: O’Reilly & Associates Inc., 1996.

[Dean 1992] Dean, E. B. and Unal, R., “Elements of designing for cost”, *The AIAA 1992 Aerospace Design Conference*, Irvine CA, 3-6 February, 1992, AIAA-92-1057.

[Dong 1987] Dong, W. M. and Wong, F. S. “Fuzzy weighted averages and implementation of the extension principle”, *Fuzzy Sets and Systems*, Vol. 21, pp. 183-199, 1987.

[Douba 1999] Douba, S. M., “*UNIX unleashed*”, Third Edition, Indianapolis: Macmillan Computer Publishing, 1999.

[decdfliB 1997] dcdflib Version 1.1, http://odin.mdacc.tmc.edu/anonftp/page_2.html, 1997.

[DeMarco 1979] DeMarco, T., “*Structured analysis and system specification*”, New Jersey: Prentice-Hall, 1979.

[Ditlevsen 1979] Ditlevsen, O., “Generalised second moment reliability index”, *Journal of Structural Mechanics*, Vol. 7, No. 4, pp. 435-451, 1979.

[Ditlevsen 1981] Ditlevsen, O., “Principal of normal tail approximation,” *Journal of*

Engineering Mechanics Division, ASCE, Vol. 107, No. 6, pp. 1191-1209, 1981.

[Ditlevsen 1996] Ditlevsen, O. and Madsen, H. O., “*Structural reliability methods*” Chichester : John Wiley & Sons, 1996.

[Dwivedi 1990] Dwivedi, S. N. and Sobolewski, M., “*Concurrent engineering-an introduction*”, CRC Technical Report Series, CERC-TR-TM-90-006, Concurrent Engineering Research Center, West Virginia University, 1990.

[Engelund 1993] Engelund, S. and Rackwitz, R., “ A benchmark study on importance sampling techniques in structural reliability”, *Structural Safety*, Vol. 12, pp. 255-276, 1993.

[Enevoldsen 1994a] Enevoldsen, I., “Reliability-based optimization as an information tool”, *Mechanisms, Structures & Machines*, Vol. 22, No. 1, 117-135, 1994.

[Enevoldsen 1994b] Enevoldsen, I. and Sorensen, J. D., “Reliability-based optimization in structural engineering”, *Structural Safety*, Vol. 15, pp. 169-196, 1994.

[Ertas 1996] Ertas, A., and Jones, J. C., “*The engineering design process*”, Second Edition, Chichester: John Wiley & Sons, 1996.

[ESREL 99] ESREL, European Safety and Reliability Conference, <http://www.grs.de/esrel.htm>, 1999.

[Farkas 1994] Farkas, J. and Jarmai, K., “Minimum cost design of laterally loaded welded rectangular cellular plates”, *Structural Optimization*, Vol. 8, pp.262-267, 1994.

[Fiacco 1990] Fiacco, A. V. and McCormick, G. P. “*Nonlinear programming: sequential unconstrained minimization techniques*”, New York: SIAM Society for Industrial & Applied Mathematics, 1990.

[Fiessler 1979] Fiessler, B., H., Neumann, and Rackwitz, R., "Quadratic limit states in structural reliability", *Journal of the Engineering Mechanics Division, ASCE*, Vol. 105, No. EM4, pp. 661-676, 1979.

[Fogel 1998] Fogel, D. B., “*Evolutionary computation: The fossil record*”, New York: IEEE Press, 1998.

[Freudenthal 1956] Freudenthal, A. M., “Safety and the probability of structural failure”, *Journal of the American Society of Civil Engineers*, Vol. 121, pp. 1337-1397,

1956.

[Freudenthal 1966] Freudenthal, A. M., Garrelts, J. M., and Shinozuka, M., “The analysis of structural safety”, *Journal of Structural Division*, ASCE, Vol. 92, No. ST1, pp. 267-325, 1996.

[Gabriele 1977] Gabriele, G. A. and Ragsdell, K. M., “The generalised reduced gradient method: a reliable tool for optimal design”, *Journal of Engineering for Industry*, ASME Series B, Vol. 99, No. 2, pp. 394-400, 1997.

[Geist 1995] Geist, A., Beguelin, A., Dongarra, J. J., Weicheng, Manchek, Robert, S. and Vaidyalingam S., “PVM: *Parallel Virtual Machine: A User's Guide and Tutorial for Network Parallel Computing*”, Cambridge, Mass: The MIT Press, 1995.

[Giunta 1997] Giunta, A. A., “*Aircraft multidisciplinary optimisation using design of experiments theory and response surface modelling methods*”, Ph.D. thesis, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, .1997.

[Giunta 1998] Giunta, A. A., and Watson, L. T., “A comparison of approximation modelling techniques: polynomial versus interpolating models”, *ISSMO/NASA First Internet Conference on Approximations and Fast Reanalysis Techniques in Engineering Optimisation*, <http://www.aero.ufl.edu/~issmo/webconf.htm>, 1988.

[Glenney 1985] Glenney, N. and Mackulak, G. T., “Modelling and simulation provide the key to CIM implementation philosophy”, *Industrial Engineering*, pp. 76-84., 1985.

[Godwin 1989] Godwin, A. N., Glesson, J. W. and Gwillian, D., “An assessment of the IDEF notations as descriptive tools”, *Information Systems*, Vol. 14, No. 1, pp. 13-28, 1989.

[Goldberg 1989] Goldberg, D. E., “*Genetic Algorithms in Search Optimization and Machine Learning*”, Reading (MA): Addison-Wesley, 1989.

[Hahn 1996] Hahn, H. “*The Internet complete reference*”, Second Edition, U.S.A: Osborne McGraw-Hill, 1996.

[Harbitz 1983] Harbitz, A., “Efficient and accurate probability of failure calculation by the use of the importance sampling technique”, *Proceedings of the Fourth International Conference on Applications of Statistics and Probability in Soil and Structural Engineering*, Pitagora Editrice, Bologna, Italy, pp. 825-836, 1983.

- [Harbitz 1986] Harbitz, A., “An efficient sampling method for probability of failure calculation”, *Structural Safety*, Vol. 3, pp. 109-115, 1986.
- [Harkins 1993] Harkins, J. and Dubreil, M. “Concurrent engineering in product design/development”, *Plastic Engineering*, Vol. 49, pp. 27-31, 1993.
- [Hasofer 1974] Hasofer, M., Lind, N.C., “Exact and invariant second-moment code format”, *Journal of Engineering Mechanics Divisions*, ASCE, Vol. 100, No. EM1, pp. 111-121, 1974.
- [Haugen 1980] Haugen, E. B., “*Probabilistic mechanical design*”, Chichester : John Wiley & Sons, 1980.
- [Hohenbichler 1981] Hohenbichler, M., and Rackwitz, R., “Non-normal dependent vectors in structural safety”, *Journal of Engineering Mechanics Divisions*, ASCE, Vol. 107, No. EM6, pp. 1227-1238, 1981.
- [Huang 1996] Huang, X., Dudley, J., Haftka, R. T., Grossman, B. and Mason, W. H., “Structural Weight Estimation for Multidisciplinary Optimization of a High-Speed Civil Transport”, *Journal of Aircraft*, Vol. 33, No. 3, pp. 608-616, 1996.
- [Hubka 1984] Hubka, V. and Eder, W. E., “*Theory of technical systems*”, Heidelberg : Springer-Verlag Berlin, 1984.
- [Hulme 1997] Hulme, K.F. and Bloebaum, C. L., “Development of a multidisciplinary design optimisation test simulator”, *Structural Optimisation*, Vol. 14, No. 2-3, pp. 129-137, 1997.
- [IASSAR 2000] IASSAR, International Association for Structural Safety and Reliability, <http://www.bkm.dtu.dk/~iassar/>, 2000.
- [IBM 2000] IBM, <http://www.research.ibm.com/bluegene/>, 2000.
- [Ibrahim 1991] Ibrahim, Y., “Observations on applications of importance sampling in structural reliability analysis”, *Structural Safety*, Vol. 9, pp. 269-281, 1991.
- [IDEF 2000] IDEF, “*IDEF family of Methods*”, <http://www.idef.com/>, 2000.
- [ISO 1992] ISO, “*Guidelines for the development of application protocols for ISO 10303*”, Version 1.0, ISO TC184/SC4/WG4N34(P5), 1992.
- [ISO 1993] ISO 10303-Part II: “*The EXPRESS language reference manual*”, ISO TC184/SC4WG5, 1993.

- [Isukapalli 1999] Isukapalli, S. S., “*Uncertainty analysis of transport-transformation models*”, PhD thesis, The state University of New Jersey, 1999.
- [Jacob 1993] Jacob, R. “Quality in concurrent engineering”, *Annual Quality Congress Transactions*, (ASQC, USA), Vol. 47, pp. 902-908, 1993.
- [Jamsa 1995] Jamsa, K. and Cope, K., “*Internet programming*”, U.S.A: Jamsa Press, 1995.
- [Jagannathan 1991] Jagannathan, V., Cleetus, R. K., Matsumoto, A. S. and Lewis, J. W., “Computer support for concurrent engineering: four strategic initiatives”, *Concurrent Engineering*, pp. 14-30, 1991.
- [Karamchandani 1992] Karamchandani, A. and Cornell, C. A., “Sensitivity estimation within first and second order reliability methods”, *Structural Safety*, Vol. 11, pp. 95-107, 1992.
- [Kaymaz 1999] Kaymaz, I. and McMahon C. A., “An approach to reliability analysis using the response surface method and Monte Carlo simulation”, *ASME'99 Las Vegas*, 15-17 September 1999.
- [Keane 2000] Keane, A. J., <http://www.soton.ac.uk/~ajk/opt/welcome.html>, 2000.
- [Kernighan 1984] Kernighan, B. W. and Pike, R., “*The UNIX programming environment*”, New Jersey: Prentice-Hall, 1984.
- [Khan 1995] Khan, M. A. and Vance, J. M., “A mesh reduction approach to parametric surface polygonization”, *Proceedings of 1995 ASME Design Automation*, Boston, Ma, DE-Vol. 82, pp. 41-48, 1995.
- [Kim 1997] Kim, S. and Na, S., “Response surface method using vector projected sampling points”, *Structural Safety*, Vol. 19, No. 1, pp. 3-19, 1997.
- [Kirkpatrick 1983] Kirkpatrick, S., Gelatt Jr., C. D., and Vecchi, M. P., “Optimisation by Simulated Annealing”, *Science*, 220, pp. 671-680, 1983.
- [Der Kiureghian 1986] Der Kiureghian, A. and Liu, P., “Structural reliability under incomplete probability information”, *Journal of Engineering Mechanics*, Vol. 112, No. 1, pp. 85-104, 1986.
- [Der Kiureghian 1987] Der Kiureghian, A., Lin, H. Z. and Hwang, S. J., “Second-Order reliability approximations”, *Journal of Engineering Mechanics*, Vol. 113, No.

8, pp, 1208-1225, 1987.

[Kleinbaum 1987] Kleinbaum, D. G., Kuper, L. L. and Muller K. E., “*Applied regression analysis and other multivariable methods*”, Boston: PWS-KENT Publishing Company, 1987.

[Klir 1994] Klir, G.J. “The many faces of uncertainty”, Ayyub, B. M. and Gupta, M. M., *Uncertainty Modeling and Analysis: Theory and Applications*, pp. 3-19, Elsevier Science, 1994.

[Klir 1995] Klir, G. J. and Bo, Y., “*Fuzzy sets and fuzzy logic: theory and applications*”, London: Prentice Hall, 1995.

[Kodiyalam 1998] Kodiyalam, S., Lin, J. S., and Wujek, B. A., “Design of experiments based response surface models for design optimisation”, *Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Long Beach, CA, April 20-23, 1998.

[Koyluoglu 1994] Koyluoglu, H., and Nielsen, S. R. K., “New approximations for SORM integrals”, *Structural Safety*, Vol. 13, pp. 235-246, 1994.

[Kreyszig 1993] Kreyszig, Erwin, “*Advanced engineering mathematics*”, Seventh Edition, Singapore: John Wiley & Sons, 1993.

[Lang 1979] Lang, O. R. “Running gear for high-speed internal-combustion engines”, *Design Handbooks*, Vol. 22, 1979.

[Law 1991] Law, A. M. and Kelton, W. D., “*Simulation modelling and analysis*”, Second Edition, New York: McGraw-Hill, 1991.

[Lee 2001] Lee, J. and Hajela, P., “Application of classifier systems in improving response surface based approximations for design optimisation”, *Computers & Structures*, Vol. 79, No. 3, pp.333-344, 2001.

[Lehmer 1951] Lehmer D. H., “Mathematical methods in large-scale computing units”, *Annals of Computing. Laboratory*, Harvard Univ., Vol. 26: 1pp. 41-146, 1951.

[Liu 1994] Liu, Y. W. and Moses, F., “A sequential response surface method and its application in the reliability analysis of aircraft structural systems”, *Structural Safety*, Vol. 16, pp. 39-46, 1994.

[Lu 1999] Lu, J., Furuya, H. and Xu, Z., “A Feasibility Study on Combining Neural

Network and Gradient Methods Based Optimization Algorithms for Structural Optimization”, *Proceedings of 3rd World Congress of Structural and Multidisciplinary Optimization (WCSMO)*, Buffalo, New York, May 17-21, 1999.

[Malhotra 1992] Malhotra, R. and Jayaraman, S., “An integrated framework for enterprise modeling”, *Journal of Manufacturing Systems*. Vol. 11, No. 6, pp. 426-441, 1992.

[Marsaglia 1964] Marsaglia, G. and Bray, T. A., “A convenient method for generating normal variables”, *SIAM Rev.*, Vol. 6, pp. 260-274, 1964.

[Maymon 1994] Maymon, G., “Direct computation of the design point of a stochastic structure using a finite element code”, *Structural Safety*, Vol. 14, pp. 185-202, 1994.

[McMahon 1994] McMahon, C. A., “Observations on modes of incremental change in design”, *Journal of Engineering Design*, Vol. 5, No. 3, pp. 195-209, 1994

[McMahon 1995] McMahon, C. A., Xianyi, M., Brown, K. N. and Sims Williams, J. H., “A parallel multi-attribute transformation model of design”, *Proceedings of the 7th ASME International Conference on Design and Methodology*, Ed. A. C. Ward, Boston, pp. 341-350, Sept 1995.

[McMahon 1996] McMahon, C. A. and Meng, X., “A network approach to parametric design integration”, *Research in Engineering Design*, Vol. 8, pp. 14-32, 1996.

[McMahon 1998] McMahon, C. A. and Browne, J., “CAD/CAM: principles, practice and manufacturing management”, Harlow : Addison-Wesley , 1998.

[Melchers 1990] Melchers, R. E., “Search-based importance sampling”, *Structural Safety*, Vol. 9, pp. 117-128, 1990.

[Meng, 1998] Meng, X. and McMahon, C. A., “Use of influence surfaces in a network approach to probabilistic analysis”, *Computer Modeling and Simulation in Engineering*, Vol. 3 , No. 1, pp. 12-19, 1998.

[Metropolis 1949] Metropolis, N. and Ulam, S. “The Monte Carlo method”, *Journal of American Statistical Association*, Vol. 44, pp. 335–341, 1949.

[Meyer 1956] Meyer, M. A. (ed.), “Symposium on Monte Carlo methods”, New

York: John Wiley & Sons, 1956.

[Michelana 1999] Michelana, N., Scheffer, C., Fellini, R., and Papalambros, P., “A Corba-Based Object-Oriented Framework for Distributed System Design”, *Journal of Mechanics of Structures and Machines*, Vol. 27, No. 4, pp. 365-392, 1999.

[Moore 1966] Moore, R. E., “Interval analysis”. Englewood Cliffs, N.J.: Prentice-Hall, 1966.

[Moore 1979] Moore, R. E., “Methods and applications of interval analysis”. Philadelphia: SIAM, 1979.

[Myers 1995] Myers, R. H. and Montgomery, D. C., “*Response surface methodology: process and product optimisation using designed experiments*”, New York: John Wiley & Sons, 1995.

[NCSA 2000] NCSA, <http://www.ncsa.uiuc.edu/Apps/CMP/RNG/www-rng.html>, 2000.

[Nikolaidis 1988] Nikolaidis, E. and Burdisso, R., “Reliability based optimisation: a safety index approach”, *Computers & Structures*, Vol. 28, pp. 781-788, 1988.

[Ohsuga 1989] Ohsuga, S., “Toward intelligent CAD systems”, *Computer-Aided Design*, Vol. 21, No. 5, pp. 317-337, 1989.

[OptServ 2000] OptServ, “CORBA-based distributed optimization system to solve general non-linear, constrained optimization problems”, <http://crunch.inf.bi.ruhr-uni-bochum.de/optserv/>, 2000.

[Orfali 1997] Orfali, R. and Harkey, D., “Client/Server programming with JAVA and CORBA”, USA: John Wiley & Sons, 1997.

[Otto 1993] Otto, R. L., “*An introduction to statistical methods and data analysis*”, New York: Thomson Learning, 1993.

[Pahl 1984] Pahl, G. and Beitz, W., “*Engineering design*”, London: The Design Council/Springer, 1984.

[Panier 1993] Panier, E. R. and Tits, A. L., “On combining feasibility, descent and superlinear convergence in inequality constrained optimisation”, *Mathematical Programming*, Vol. 59, pp. 261-176, 1993.

[Papadrakakis 1996] Papadrakakis, M., Papadopoulos, V. and Lagaros, N. D.,

“Structural reliability analysis of elastic-plastic structures using neural networks and Monte Carlo simulation”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 136, pp. 145-163, 1996.

[Peterson 1981] Peterson, J. L., “Petri net theory and the modelling of systems”, New Jersey: Prentice Hall, 1981.

[Prasad 1995] Prasad, B., “Sequential versus concurrent engineering-an analogy”, *Concurrent Engineering*, Vol. 3, No. 4, pp. 250-255, 1995.

[Press 1992] Press, W., H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., “*Numerical recipes in C*”, Second Edition, London: Cambridge Press, 1992.

[PROVERBS 2000] PROVERBS, Probabilistic Design Tools for Vertical Breakwaters, <http://www-public.tu-bs.de:8080/~i5102401/proverbs.html>, 2000.

[Pugh 1991] Pugh, S., “*Total design: integrated methods for successful engineering*”, Wokingham: Addison-Wesley, 1991.

[Rabb 1996] Rabb, R., “Fatigue failure of a connecting rod”, *Engineering Failure Analysis*, Vol. 2, No. 1, pp. 13-28, 1996.

[Radcliffe 1994] Radcliffe, N. J., “The Algebra of Genetic Algorithms”, *Annals of Mathematics and Artificial Intelligence*, Vol. 10, pp. 339-384, 1994.

[Rao 1980] Rao, S. S., “Structural optimisation by chance constrained programming techniques”, *Computers & Structures*, Vol. 12, pp. 777-781, 1980.

[Rackwitz 1976] Rackwitz, R., “Practical probabilistic approach to design”, Bulletin 112, Committee European du Beton, Paris, France, 1976.

[Rackwitz 1978] Rackwitz, R. and Fiessler, B., “Structural reliability under combined random load sequences”, *Computers & Structures*, Vol. 9, pp. 489-494, 1978.

[Rajashekhar 1993] Rajashekhar, M. R. and Ellingwood, B. R., “A new look at the response surface approach for reliability analysis”, *Structural Safety*, Vol. 12, pp. 205-220, 1993.

[Reisig 1985] Reisig, W., “*Petri nets: an introduction*”, New York : Springer-Verlag, 1985.

[Rosenblatt 1969] Rosenblatt, M., “Remarks on multivariate transformation”, *Annals of Mathematical Statistics*, Vol. 23, No. 3, pp. 470-472, 1969.

[Ross 1994] Ross, T. "Fuzzy logic with engineering applications", London: McGraw-Hill, 1994.

[Rumelhart 1986] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., "Learning representations by backpropagation errors", *Nature*, No.323, pp. 533-536, 1986.

[Wendt 1997] Wendt, R. A., "*Investigation of the behaviour of parametrically modelled parts in the finite element program Ansys5.3, and analysis of the response surfaces with mathematical techniques*", Diplomarbeit, University of Bristol, April 1997.

[Rubinstein 1981] Rubinstein, R. Y., "*Simulation and Monte Carlo method*", New York: John Wiley & Sons, 1981.

[Ryan 1997] Ryan, T. "*Modern regression methods*", New York: John Wiley & Sons, 1997.

[Sacks 1989] Sacks, J., Schiller, S. B. and Welch, W., "Designs for computer experiments", *Technometrics*, Vol. 31, No. 1, pp. 41-47, 1989.

[Schinozuka 1983] Schinozuka, M., "Basic analysis of structural safety", *Journal of Structural Engineering*, Vol. 109, No. 3, pp. 1707-1724, 1983.

[Schueller 1987] Schueller, G. I. and Stix, R., "A critical appraisal of methods to determine failure probabilities", *Structural Safety*, Vol. 4, pp. 293-309, 1987.

[Schmit 1974] Schmit, L. A. and Farshi, B., "Some approximation concepts for structural synthesis", *AIAA Journals*, Vol. 12, No.5, pp. 692-699, 1974.

[Schütz 2000] Schütz, M. and Schwefel, H. P., "Evolutionary approaches to solve three challenging engineering tasks", *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 2-4, pp. 141-170, 2000.

[Siddal 1983] Siddal, J. N., "*Probabilistic engineering design*", New York: Marcel Dekker, 1983.

[Simpson 1997] Simpson, T. W., Peplinski, J. D., Koch, P. N. and Allen, J. K., "On the use of statistics in design and the implications for deterministic computer experiments", *Proceedings of the 1997 AMSE Design Engineering Technical Conference*, Sacramento, California, 1997, Paper No. DETC97/DTM-3881.

[Simpson 1998] Simpson, T. W., "Comparison of response surface and kriging

models in the multidisciplinary design of an aerospike nozzle”, *ICASE Report*, NASA/CR-1998-206, 1998.

[Sobieszczanski-Sobieski 1984] Sobieszczanski-Sobieski, J.; Barthelemy, J. F. M. and Giles, G. L., “Aerospace engineering design by systematic decomposition and multilevel optimisation”, *14th Congress of the International Council of the Aeronautical Sciences (ICAS)*, Toulouse, France, September 1984.

[Sohlenius 1992] Sohlenius, G., “Concurrent engineering”, *CIRP Annals*, Vol. 41, pp. 645-656, 1992.

[Song 1997] Song, B. F., “A technique for computing failure probability of a structure using importance sampling”, *Computers & Structures*, Vol. 62, No. 4, pp. 659-665, 1997.

[Stark 1986] Stark, H., “*Probability, random processes, and estimation theory for engineers*”, London: Prentice-Hall, 1986.

[STEPTOOLS 2000] STEPTOOLS, STEP solutions for Interoperability & e-Manufacturing, <http://www.steptools.com/index.html>, 2000.

[ST-REPOSITORY 2000] ST-REPOSITORY, <http://www.steptools.com/products/strepo/index.html>, 2000.

[Stevens 1990] Stevens, W. R., “UNIX network programming”, New Jersey: Prentice-Hall Inc., 1990.

[Sundararajan 1994] Sundararajan, C., “*Probabilistic structural mechanics handbook*”, London: Chapman & Hall, 1994.

[Sweeting 1992] Sweeting, T. J. and Finn, A. F., “A Monte Carlo method based on first-order and second-order reliability approximations”, *Structural Safety*, Vol. 11, pp. 203-212, 1992.

[SWRI 2000] SWRI, *South West Research Institute*, <http://www.swri.org/3pubs/papers/d06/probmech.htm>, 2000.

[Taguchi 1986] Taguchi, G., “*Introduction to quality engineering*”, New York: Asian Productivity Organization, 1986.

[Taylor 1992] Taylor, D. L., “Computer-Aided design”, New York: Addison – Wesley, 1992.

- [Toft-Christensen 1982] Toft-Christensen, P. and Baker, M. J., “*Structural reliability theory and its applications*”, Berlin: Springer-Verlag, 1982.
- [Tsai 1995] Tsai, C., Chang, K. and Wang, J., “Integration infrastructure for a simulation-based design environment”, *the 15th Annual international computers in engineering*, ASME, September 17-20, Boston, Massachusetts, 1995.
- [Tvedt 1983] Tvedt, L., “Two second-order approximations to the failure probability”, Veritas report RDIV/20-004083, Det norske Veritas, Oslo, 1983.
- [Tvedt 1988] Tvedt, L., “Second-order reliability by an exact integral”, *Proceedings of 2nd IFIP working conference on reliability and optimisation on structural systems*, P. Thoft-Christensen (ed.), New York: Springer-Verlag, pp. 337-384, 1988.
- [Tvedt 1990] Tvedt, L., “Distribution of quadratic forms in normal space - application to structural reliability”, *Journal of Engineering Mechanics*, Vol. 116, No. 6, 1990.
- [Ullman 1992] Ullman D. G., “*The mechanical design process*”, New York: McGraw Hill, 1992.
- [Ullman 1995] Ullman, D. G. and D’Ambrosio, B., “A taxonomy for classifying engineering decision problems and support systems”, *Proceedings of the 7th International Conference on Design Theory and Methodology*, (ed.) Ward, A. C., held as part of the ASME Design Engineering Technical Conferences, Boston, Mass., Vol. 83, No. 2, pp. 627-637, Sept 1995.
- [Unal 1994] Unal, R., Stanley, D. O. and Joyner, C. R., “Parametric model building and design optimisation using response surface methods”, *Journal of Parametrics*, Vol. 14, No. 1, pp. 81-96, 1994.
- [UW-stout 2000] University of Wisconsin-stout, “*Stress & Strength of Materials*” <http://physics.uwstout.edu/statics/indexfbt.htm#STATICS%20&%20STRENGTH%20OF%20MATERIAL,2000>.
- [Vanderplaats 1984] Vanderplaats, G. N., “*Numerical optimisation techniques for engineering design: with applications*”, New York: Mc-Graw Hill, 1984.
- [Van Laarhoven 1987] Van Laarhoven, P. J. M. and Aarts, E. H. L., “*Simulated Annealing: Theory and Applications*”, Dordrecht: Kluwer Academic Publishers, 1987.

[Venter 1996] Venter, G., Haftka, R. T. and Starnes, J. H., “Construction of response surfaces for design optimisation applications”, *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*, Bellevue, Washington, September 4-6, Paper No. 96-4040, Part 1, pp. 548 – 564, 1996.

[Venter 1997] Venter, G. and Haftka, R.T., “Minimum-Bias based experimental design for constructing surfaces in structural optimisation”, *Proceedings of the 38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, Florida, April 7-10, 1997, Paper No. 97-1053, Part 2, pp. 1225-1238.

[Venter 1998] Venter, G. “*Non-Dimensional response surface for structural optimisation with uncertainty*”, PhD thesis, University of Florida, 1998.

[Virtual 2000] <http://random.mat.sbg.ac.at/links/index.html>, 2000.

[Vogel 1998] Vogel, A. and Duddy, K., “Java Programming with CORBA”, New York: John Wiley and Sons, 1998.

[Voon 1991] Voon, B. K. and Austin, M. A., “Structural Optimization in a Distributed Computing Environment”, *Report at the Institute for System Research at University of Maryland*, Report No:TR91-104, 1991.

[Wang 1996] Wang, L. and Grandhi, R. V., “Safety index calculation using intervening variables for structural reliability analysis”, *Computers & Structures*, Vol. 59, No. 6, pp. 1139-1148, 1996.

[Wang 1997] Wang, W., Wu, J. and Lust, R.V., “Deterministic design, reliability-based design and robust design”, *MCS 1997 Aerospace User’s Conference Proceedings*, 1997.

[Welch 1989] Welch, W. J., Mitchell, T. J. and Wynn, H. P., “Design and analysis of computer experiments”, *Statistical Science*, Vol. 4, No. 4, pp. 409-435, 1989.

[Wilson 1996] Wilson, P., “*EXPRESS-I language reference manual*”, ISO TC184/SC4 N440, http://www.nist.gov/sc4/step/parts/part012/tech_rpt/part12.pdf, 1996.

[Winner 1988] Winner, R. I., Pennel, J. P., Bertrand, H. E. and Slasarezuk, M. M.

G., “*The role of concurrent engineering in weapons system acquisition*”, IDA Report R-338, Institute for Defence Analyses, Alexandria VA, 1988.

[Wood 1994] Wood, K. L., Antonsson, E. K. and Beck, J. L., “Representing imprecision in engineering design: comparing fuzzy and probability calculus”, *Research in Engineering Design*, Vol. 1, pp. 187-203, 1990.

[Wu 1987] Wu, Y. and Wirsching, P. H., “New algorithm for structural reliability estimation”, *Journal of Engineering Mechanics*, Vol. 113, No. 9, pp. 1319-1336, 1987.

[Wu 1992] Wu, J. K., Koppes, B., Standefer, J., Ciarelli, K., Calbeck, D. and Smuda, B., “*Integrated CAD/CAE system for tracked vehicle design and evaluation*”, Technical Report R128, University of Iowa, Center for Computer-Aided Design, 1992.

[Yao 1996] Yao, T. H. J. and Wen, Y. K., “Response surface method for time-variant reliability analysis”, *Journal of Structural Engineering*, Vol. 122, No. 2, 1996.

[Yi 1999] Yi, C., Mingwu, W. and Ling, T., “Fatigue strength reliability analysis of diesel connecting rod based on stochastic finite element method”, *Journal of Engineering for Gas Turbines and Power*, Vol. 121, 1999.

[Yourdon 1989] Yourdon, E., “*Modern structured analysis*”, New Jersey: Prentice-Hall, 1989.

[Yu 1998] Yu, X., Chang, K. H. and Choi, K. K., “Probabilistic structural durability prediction”, *AIAA Journal*, Vol. 36, No. 4, pp. 628-637, 1998.

[Zha 1996] Zha, G. C., Smith, D., Schwabacher, M., Gelsey, A. and Knight, D., “High performance supersonic missile inlet design using automated optimization”, *Proceedings of the 6th AIAA/USAF/NASA/ISSMO Symposium on Multi-disciplinary Analysis and Optimization*, 1996.

[Zadeh 1965] Zadeh, L. A., “Fuzzy sets,” *Information and Control*, Vol. 8, pp. 29–44, 1965.

[Zadeh 1975] Zadeh, L. A., “The concept of linguistic variable and its application to approximate reasoning”, *Information Science*, Vol. 8, 1975.

[Zadeh 1978] Zadeh, L. A., “Fuzzy Sets as a basis for a theory of possibility”, *Fuzzy*

Sets and Systems, Vol. 1, pp. 3-28, 1978.

[Zhao 1997] Zhao, Y.G. and Ono, T., “An empirical reliability index based on SORM”, *Proceedings of the 7th International conference on structural reliability analysis*, Kyoto, Japan, 1997.

[Zhao 1999a] Zhao, Y.G. and Ono, T., “New approximation for SORM: Part 1”, *Journal of Engineering Mechanics*, Vol. 125, No. 1, pp. 79-85, 1999.

[Zhao 1999b] Zhao, Y.G. and Ono, T., “New approximation for SORM: Part 2”, *Journal of Engineering Mechanics*, Vol. 125, No. 1, pp 86-93, 1999.