STRUCTURAL OPTIMIZATION USING ANSYS AND REGULATED MULTIQUADRIC RESPONSE SURFACE MODEL

by

AJAYKUMAR MENON

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2005

To my parents for their belief in my education

and to my brother Ajith

for his constant support and love

ACKNOWLEDGEMENTS

I would like to express my deep gratitude and admiration to my advisor and mentor Dr. Kent L. Lawrence, who is a constant source of guidance and inspiration to me. He has been very patient and extended excellent support, never accepting less than my best efforts. I would like to mention a special thanks to Dr. B. P. Wang for bringing up this concept and for providing valuable information on response surface based optimization and also on the MATLAB codes that forms the basis of this research. This thesis would never have been completed without their inspiration and continuous support.

I want to extend my sincere appreciation and thanks to my committee members, Dr. Seiichi Nomura and Dr. Wen S. Chan, who have been extremely helpful during the whole process of thesis review and have helped enrich my thesis with their valuable suggestions.

I would also like to thank the entire faculty of Mechanical Engineering for their help in building my broad understanding of the subject and in preparation of this thesis. Many thanks as well to my friends and roommates who have always been ready to help me especially, Ajit Kailas, Ajit John, Charlie, Deepu, Divija, Gijesh, Krishna Kumar, Manne, Raghu, Rajeev, Rahul, Robin, Roopesh, Sarang, Sebastian, Shyam Venugopal, Sheril, Sushanth, Unni and Vinu.

To my loving brother Ajith and to all my cousins, I want to express my appreciation for the encouragement they made in support of my study at UTA.

Most importantly I would like to dedicate this work to my parents who have been there for me always and are responsible for what I am today and without whom, the joy of completing the educational phase of my life would be incomplete.

November 14, 2005

ABSTRACT

STRUCTURAL OPTIMIZATION USING ANSYS
AND REGULATED MULTIQUADRIC
RESPONSE SURFACE MODEL

Publication No. ______

Ajaykumar Menon, M.S.

The University of Texas at Arlington, 2005

Supervising Professor: Dr. Kent L. Lawrence

The high computational expense of large non-linear and complex finite element analysis limits or often prohibits the use of conventional codes in engineering design and multidisciplinary optimization. Consequently alternate methods such as Design of experiments (DOE) and Response surface approximation are commonly used to minimize the computational cost of running such analysis and simulation. The basic approach of such methods is to construct a simplified mathematical approximation of the computationally expensive simulation and analysis code, which is then used in place of the original code to facilitate multidisciplinary optimization, design space

exploration, reliability analysis etc. When such codes along with powerful finite element analysis tools such as ANSYS are tied with good optimization algorithms, solving complex structural optimization problems are no longer an issue.

This research work aims at defining such an automation process in MATLAB that incorporates a response surface approximating tool called MQR which is based on Radial basis function, ANSYS a finite element solver and a suitable gradient based optimization algorithm (SQP). Certain standard test cases are considered that are based on size and dynamic response optimization. The results obtained from the proposed method are compared with ANSYS DesignXplorer goal driven optimization which is based on DOE and also with ANSYS First order optimization technique. The comparison of the results demonstrates the accuracy and effectiveness of the proposed MQR based optimization process.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS……………………………………………………………iv

ABSTRACT………………………………………………………………………vi

LIST OF FIGURES……………………………………………………………..xi

LIST OF TABLES...xiv

Chapter

1.INTRODUCTION 1

1.1 Introduction to Engineering Optimization 1

1.2 Design of Experiments and Response Surface Modeling in Optimization 2

1.3 Objective and Approach 3

2.OPTIMIZATION AND ITS ROLE IN STRUCTURAL DESIGN 5

2.1 Definition and Applications 5

2.1.1. Statement of an Optimization Problem 6

2.2 Discussion on Commonly used Optimization Techniques 10

2.2.1 Sequential Quadratic Programming 11

2.2.2 Genetic Algorithm 12

2.2.3 First Order Optimization Method 14

2.3 Pareto Optimality Criterion 14

2.4 Optimization based on Finite Elements 15

2.4.1 Classification of Finite Element based
Optimization problems......17

3.RESPONSE SURFACE METHODOLOGY (RSM) AND
DESIGN OF EXPERIMENTS (DOE) 19

3.1 Introduction 19

3.1.1 “What is good” about DOE and RSM in Design Optimization...... 21

3.1.2 “What’s not so good” about DOE and RSM in Design Optimization...... 21

3.2 Meta-modeling Methodologies...... 22

3.2.1 Second Order Response Surface Approximation 23

3.2.2 Radial Basis Function 25

3.3 Design of Experiments 27

3.3.1 Comparison of Classical and Modern DOE Techniques...... 29

3.3.2 Quasi-Monte Carlo Sampling 31

4.DESIGN OPTIMIZATION PROCESS IMPLEMENTATION...... 32

4.1 Introduction 32

4.1.1 Development of ANSYS APDL file...... 33

4.1.2 Algorithm to find the Optimal Weight or Coefficient Matrix. 33

4.1.3 Algorithm of the Design Automation Process. 37

5.APPLICATIONS...... 43

5.1 Introduction 43

5.2 9-Bar Truss 45

5.3 25-Bar Truss 54

5.4 Plate with an Elliptical Hole...... 61

5.5 Two Dimensional Vehicle Suspension 70
5.5.1 Validation of ANSYS model 81
6.CONCLUSIONS AND RECOMMENDATIONS 86
6.1 Introduction 86
6.1.1 Conclusions 86
6.1.2 Recommendations 87
Appendix
A. ANSYS APDL SAMPLE FILES 89
REFERENCES 113
BIOGRAPHICAL INFORMATION 116

LIST OF FIGURES

Figure Page

2.1 Function plot depicting optimum solution for a 2 design variable set. ... 9

3.1 CCD samples for 2 design variable set ... 30

4.1 Block diagram of an MQR process ... 39

4.2 Two bar truss problem ... 40

4.3 MQR Block diagram for 2 bar truss ... 40

4.4 General layout of MQR optimization process ... 42

5.1 9-Bar truss ... 46

5.2 Free body diagram of 9-Bar truss ... 46

5.3 Quasi-Monte Carlo sampling for 9-bar truss. ... 47

5.4 Data points used at the end of convergence ... 48

5.5 Function convergence plot for the MQR optimization process ... 48

5.6 Function convergence 9-Bar truss- ANSYS First order optimization process. ... 49

5.7 Results from ANSYS DesignXplorer goal driven optimization for 9-bar truss. ... 50

5.8 Response plot for total volume. ... 50

5.9 Response plot for maximum stress. ... 51

5.10 Response plot for maximum displacement of nodes along X direction. ... 51

5.11 Response plot for maximum displacement of nodes along Y direction. 52

5.12 Tradeoff plot for first sample generation -9-Bar Truss. 53

5.13 Tradeoff plot for last sample generation -9-Bar Truss 53

5.14 25-Bar truss 55

5.15 Element groups for the 25-Bar truss problem. 55

5.16 Function convergence plot for 25-Bar truss structure with 7 element groups. 57

5.17 Optimum values of the design variables. 57

5.18 Function convergence plot in ANSYS for 25-Bar truss structure with 7 element groups. 58

5.19 Optimum values of the design variables. 58

5.20 Results from DesignXplorer goal driven optimization for 25-bar truss problem -7 design variables. 59

5.21 Tradeoff plot for first sample generation -25-Bar Truss 60

5.22 Tradeoff plot for last sample generation -25-Bar Truss 61

5.23 Plate with the elliptical hole subjected to tensile load. 62

5.24 Convergence plot for the plate problem with four design variables 64

5.25 Convergence plot for the plate problem with two design variables. 65

5.26 ANSYS Convergence plot for the plate problem-2 design variables. 66

5.27 Finite element model of the plate with
an elliptical hole 66

5.28 Results from DesignXplorer Goal driven optimization 67

5.29 Tradeoff plot- first sample generation
Plate with an elliptical hole 67

5.30 Tradeoff plot -last sample generation
Plate with an elliptical hole 68

5.31 Response plot of Total volume 'TVOL' 69

5.32 Response plot for maximum stress 69

5.33 2D vehicle model 71

5.34 Excitation function plot 72

5.35 Nonlinear characteristic of the rear suspension spring 73

5.36 Function convergence plot 77

5.37 Function convergence - First order optimization 77

5.38 Results from DesignXplorer goal driven optimization 78

5.39 Tradeoff plate-first sample generation 79

5.40 Tradeoff plot-last set of sample generation 79

5.41 Dynamic response of the displacement of sprung mass 80

5.42 Dynamic response as observed in reference [20]. 81

5.43 Deflection plot using fourth order Runge Kutta
method in MATLAB 82

5.44 One dimensional vehicle model 83

5.45 Deflection plot-mathematical model 85

5.46 ANSYS- deflection plot for 1D vehicle model 85

LIST OF TABLES

Table Page

5.1 Elements used in ANSYS for 2D vehicle suspension..................................... 74

CHAPTER 1

INTRODUCTION

1.1 Introduction to Engineering Optimization

The ever-increasing demand to lower the production costs due to increased competition has prompted engineers to look for rigorous methods of decision making such as optimization. As a result engineering optimization was developed to help engineers design systems that are both more efficient and less expensive and to develop innovative methods to improve the performance of the existing systems. Engineering optimization can best be classified as a rigorous mathematical approach to identify and select a best candidate from a set of probable design alternatives (Rao, [1]).

Optimization in its broad sense can be applied to solve any engineering problem. Having reached a degree of maturity over the past several years, optimization techniques are currently being used in a wide variety of industries, including aerospace, automotive, MEMS, chemical, electrical and manufacturing industries. With the development of computer technology, complexity of problems being solved using optimization methods is no longer an issue. Optimization methods coupled with modern tools of computer-aided design are also being used to enhance the creative process of conceptual and detailed design of engineering systems.

There is no single method or technique for solving all optimization problems efficiently. Hence a number of optimization methods have been developed for solving different types of optimization problems. It is in the entire discretion of the engineer to choose a method which is computationally efficient, accurate and appropriate for his design problem.

1.2 Design of Experiments and Response Surface Modeling in Optimization

Optimization methods known as mathematical programming techniques are generally studied as a part of Operations Research. This is a branch in mathematics that employs scientific methods and techniques to decision making problems with the aim of establishing the best or optimal solutions [1]. Design of experiments (DOE) is one such well defined area of operation research. This method enables one to analyze the experimental data and build empirical models to obtain the most accurate representation of the physical situation.

Today's engineering structures are often analyzed using Finite Element Methods. The finite element method is a numerical procedure for analyzing structures and continua (Cook, [2]). Extensive research is being done in the field of design optimization with finite element analysis as a simulation and evaluation tool. However there are certain optimization problems such as structural optimization that often involve expensive function evaluations. For instance a normal crash simulation of a passenger car takes about 27 hours with an estimated computational cost of about $5200 (Yang, [3]). Consequently alternate methods of function evaluations such as design of experiments (DOE) and response surface modeling (RSM) are commonly employed in engineering design to minimize the computational cost involved in such analysis and simulation.

The basic approach of such methods is to construct a simplified mathematical approximation of the computationally expensive simulation and analysis code, which is then used in place of the original code to facilitate Multidisciplinary Optimization (MDO), reliability analysis, design space exploration etc. A variety of approximation models exist such as polynomial response surfaces, Kriging model, radial basis functions, neural networks and multivariate adaptive regression splines (Simpson, [4]). In this research a classification of radial basis function known as Regulated Multiquadric Response Surface (MQR) model is employed to approximate the expensive simulation and analysis code.

1.3 Objective and Approach

As discussed in the previous section, there are certain complex optimization problems that demand expensive function evaluations. Resort to alternate methods of function evaluations such as Design of experiments (DOE) and Response surface modeling (RSM) are made to minimize the computational expenses incurred in solving such problems.

The main objective of this research is to explore the possibility of solving such structural optimization problems accurately using suitable DOE and RSM tools. This is achieved by developing a suitable design automation code in Matlab that incorporates ANSYS a finite element analysis Tool and MQR (Multiquadric Response surface) a DOE and RSM tool with a suitable optimization technique. The optimization tool available in Matlab (‘fmincon’) based on sequential quadratic programming is used as the optimization method to solve the selected set of application problems. The results obtained adopting this method are then compared with those obtained using ANSYS inbuilt First Order optimization method and also with ANSYS DesignXplorer module available within ANSYS WORKBENCH 9.

The application problems considered in this research are

1. Weight minimization of 9-bar truss, by finding optimal cross-sectional areas of the truss members.
2. Weight minimization of 25-bar truss, by finding optimal cross-sectional areas of the truss members.
3. Finding the optimal size of an elliptical hole in a rectangular plate for minimum weight so as to withstand the applied tensile load.
4. Optimal design of a 2 dimensional vehicle suspension for ride quality and comfort.

CHAPTER 2

OPTIMIZATION AND ITS ROLE IN STRUCTURAL DESIGN

2.1 Definition and Applications

Optimization may be defined as the process of maximizing or minimizing a desired objective function while satisfying the prevailing constraints (Belegundu, [5]). In every stage of design, construction and maintenance of engineering systems, engineers are bound to take certain technological and managerial decisions. The ultimate goal of all such decisions is either to minimize the effort required or maximize the desired benefit. Since either of these goals in any physical situation can be expressed as a function of certain design variables, *optimization* may also be defined as the process of finding the conditions that give the maximum or minimum value of a function [1].

Nature provides abundance examples of optimization. For example in metals and alloys, atoms take the position of least energy to form unit cells. It is these unit cells that define the crystalline structure of materials. Another example of nature's optimization process is genetic mutation for survival. Like nature, organizations and businesses employ optimization in their work process to meet the current consumer demands and increased competitions. In engineering, optimization can be used to solve any problem. Some typical applications from different engineering disciplines are-

1. Design of aircraft and aerospace structures for minimum weight.

2. Vibration and noise optimization of automobile for ride quality, comfort and handling.
3. Optimal design of electric networks.
4. Analysis of statistical data and building empirical models from experimental results to obtain the most accurate representation of the physical phenomenon.
5. Optimal production planning, controlling and scheduling etc.

As discussed before, a number of optimization methods are available to solve such problems. However, for engineers to apply optimization at their work place they need to understand the theory, the algorithm and the techniques behind these methods. This is because practical problems may require modifying algorithmic parameters and even scaling and adapting the existing methods to suit the specific application. Above all, the user may have to try out a number of optimization methods to find one that can be successfully applied.

2.1.1 Statement of an Optimization Problem

Majority of engineering problems often involve constrained minimization. An example of such constrained minimization problem is finding the minimum weight design of a structure subject to constraints on stress and deflection. Constrained problems may be expressed in the following general nonlinear programming form [4]:

$$\begin{aligned}
&\text{minimize} && f(\mathbf{x}) && \\
&\text{subject to} && g_i(\mathbf{x}) \le 0 && i = 1,\dots\dots,m \\
&\text{and} && h_j(\mathbf{x}) = 0 && j = 1,\dots\dots,l
\end{aligned} \tag{2.1}$$

where $\mathbf{x} = (x_1, x_2, \ldots\ldots, x_n)^T$ is a column vector of n real-valued design variables. f is the objective or cost function, g's are inequality constraints, and h's are equality constraints. The inequality constraints in Eq. (2.1) include explicit lower and upper bounds on the design variables. We may also express Eq. (2.1) in the form: minimize $f(\mathbf{x})$, $\mathbf{x} \in \mathbf{\Omega}$ where

$$\mathbf{\Omega} = \{\mathbf{x}: \mathbf{g} \leq 0, \mathbf{h} = \mathbf{0}\} \tag{2.2}$$

$\mathbf{\Omega}$ is the feasible region or feasible set. For unconstrained problems the feasible region is the entire space or $\mathbf{x} \in \mathbf{R^n}$. Objective function and constraints of linear programming problems involve linear functions of **x**, where as objective function in quadratic programming problems is a quadratic function of the variables while the constraints are linear.

The *design space or design variable space* in an optimization problem can be considered as an n-dimensional Cartesian coordinate space where each coordinate axis represents a design variable x_i (i=1,…..n). A design point is a point on the design space that may represent a possible or an impossible solution. Design variables cannot be chosen arbitrarily; they have to satisfy certain specific functional requirements to produce an acceptable design. These restrictions that must be satisfied in a design are called *design constraints*.

Design constraints are classified into two; one that represent limitations on the behavior or performance of the system and one that pose physical limitations on the design variables such as availability ,fabric ability, transportability etc. While the former is referred to as behavior or functional constraint, the latter is known as geometric or side constraints.

The values of the design variable belonging to the set **x** that satisfy $g_i(\mathbf{x}) = 0$ forms a hyper-surface on the design space called the *constraint surface*. This is an (n-1) dimensional subspace where n represents the number of design variables. The constraint surface divides the design space into two; one where $g_i(\mathbf{x}) < 0$ and the other in which $g_i(\mathbf{x}) > 0$. Design points on the hyper-surface i.e. points that satisfy $g_i(\mathbf{x}) = 0$ satisfy the constraint $g_i(\mathbf{x})$ critically. Those lying on the region where $g_i(\mathbf{x}) > 0$ are infeasible and unacceptable while those on the region belonging to $g_i(\mathbf{x}) < 0$ are feasible and acceptable. The collection of all constraint surfaces i.e. $g_i(\mathbf{x}) = 0$, i=1,….m that separates the acceptable region is known as the *composite constraint surface*. A design point that lies on one or more constraint surfaces is known as a node point and its associated constraint as an *active constraint*. Those points that do not lie on the constraint surface are known as *free points*. Depending on the location of a design point on the design space, it can be classified into four as:

1. A free and acceptable point
2. A free and unacceptable point
3. A bound and acceptable point and
4. A bound and unacceptable point.

In general there will be more than one acceptable design point and our objective is to choose the best from the lot. This is obtained by specifying a criterion to compare the acceptable design and choosing the best one from it. This criteria or function is known as the *cost or objective function* of the optimization problem. When there are more than one objective function then the problem is known as a *multi-objective programming* problem. Like constraint surfaces, objective functions also form hyper-surfaces known as *objective function surfaces*. Once the objective function surfaces are drawn along with constraint surfaces on the design space, the optimum point can be easily located graphically as shown below.

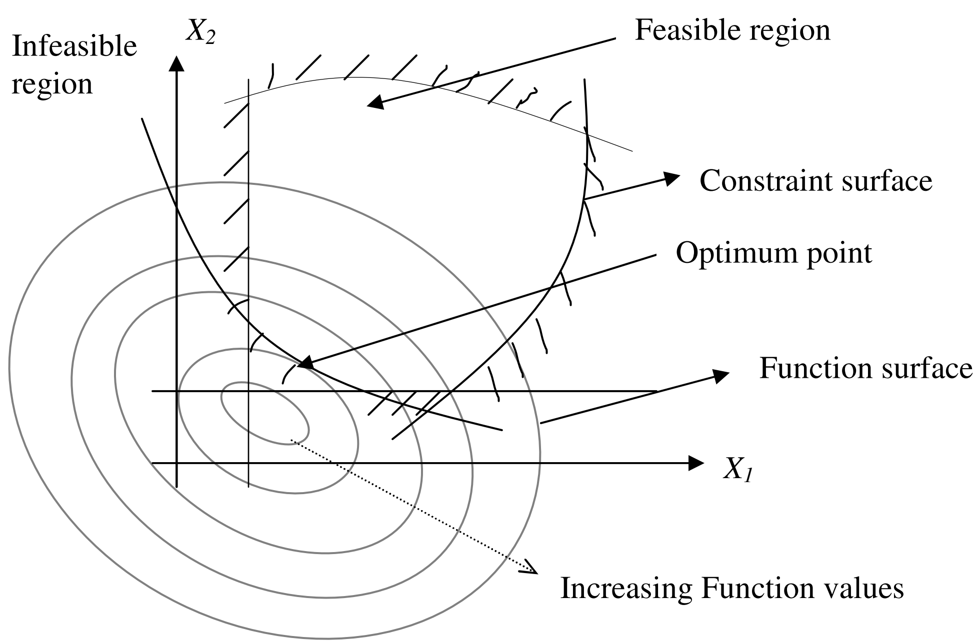


Figure 2.1 Function plot depicting optimum solution for a 2 design variable set

It can be observed that for a two design variable problem, the optimal point can be easily visualized and solved graphically. However when the number of design variables exceeds two then it becomes difficult to visualize the problem and can be only solved mathematically [1].

2.2 Discussion on Commonly used Optimization Techniques

As discussed in the previous chapter, Optimization techniques are studied as a part of operations research. The optimum seeking methods of operations research are categorized as [1]:

1) *Mathematical Programming Techniques*: Mathematical Programming Techniques are used to find the minimum of a function of several variables under a prescribed set of constraints. Examples of such methods are: Calculus methods; nonlinear programming; geometric programming; quadratic programming; sequential quadratic programming (SQP); linear programming; genetic algorithm etc.

2) *Stochastic Process Techniques:* They are used to analyze problems described by a set of random variables with known probability distribution. Examples are: Markov Process; queuing theory; statistical decision theory etc.

3) *Statistical Methods:* Statistical methods are used to build empirical models from experimental data through analysis in order to obtain the most accurate representation of the physical situation. Examples are: Regression analysis; Design of Experiments (DOE) etc.

The choice of methods depends on the type of problems being solved. There is no single method to solve all optimization problems efficiently. Hence one has to try various methods in order to choose the best one that proves to be computationally efficient and accurate. The subsequent subsections of this topic provides a brief discussion on the optimization methods used in this research work along with a discussion on one of the optimality criterion known as *Pareto optimality* employed in ANSYS DesignXplorer to find the best optimal solution for the design problem.

2.2.1 Sequential Quadratic Programming

Sequential Quadratic Programming (SQP) has received a lot of attention in the recent years owing to the superior rate of convergence (Schittkowski, [6]). It represents a state of the art in non linear programming methods. Finite element based problems, that involve relatively large number of degrees of freedom and design variables are quite effectively solved using SQP. The formulation of an SQP is based on Newton's method and Karush-Kuhn-Tucker (KKT) optimality conditions for constrained problems. This method was first published by Pshenichny in 1970 and was called a "linearization method" (Pshenichny, [7]). While this method has received some attention in the past for engineering applications, the algorithms have deviated from the theory originally presented by Pshenichny; consequently the algorithms work well on certain problems but fail on others.

All gradient methods involve two major tasks: 1) Direction finding or where to go in the design space, and 2) step size selection or how far to go. Once these two parameters are determined, a new and improved design point on the design space can be obtained as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \, \mathbf{d}_k \tag{2.3}$$

where $\mathbf{d_k}$ signifies the direction vector at the given point $\mathbf{x_k}$ and $\boldsymbol{\alpha_k}$ represent the step size. A number of iterations are performed to reach at the best optimum point. Iteration involves solving a quadratic programming sub problem to find the direction vector at the point $\mathbf{x_k}$. The step length parameter $\boldsymbol{\alpha_k}$ is determined by an appropriate line search algorithm that minimizes the merit function.

The sequential quadratic programming (SQP) has several attractions: 1) The starting point can be infeasible. 2) Gradients of only active constraints are needed. 3) Equality constraints can be handled in addition to the inequalities. 4) And Non linear constrained problems can be solved in less iteration than an unconstrained problem. One of the reasons for this is that, because of the limits on the feasible area, the optimizer can make informed decisions regarding directions of search and step length.

On account of these factors, in this research work a MATLAB optimization tool called "fmincon" based on SQP was employed.

2.2.2 Genetic Algorithm

Genetic Algorithms are search algorithms based on the principle of evolution and survival of the fittest. John Holland, from the University of Michigan began his work on genetic algorithms at the beginning of the 60s (Holland, [8]). The computational techniques developed by Holland simulated the evolution process and applied it to mathematical programming. These algorithms guide the evolution of a set of randomly selected design variables from the design space towards a near and in some cases to an optimal solution.

Genetic Algorithms differ from more traditional optimization techniques; in that they involve a search from a "population" of solutions, not from a single point. Each iteration or generation of a Genetic Algorithm involves a competitive selection that weeds out poor solutions. The solutions with high "fitness" are "recombined" with other solutions by swapping parts of a solution with another. Solutions are also "mutated" by making a small change to a single element of the solution. Recombination and mutation are used to generate new solutions that are biased towards regions of the space for which good solutions have already been seen. But such algorithms have proved to be computationally expensive when solving complex design problems. However the development in computer technology and the nature of such algorithms have rendered them suitable for implementation on parallel processing machines. The general steps followed by a Genetic Algorithm process can be summarized as:

1) Initialize the population

2) Evaluate initial population

3) Perform competitive selection

4) Apply genetic operators to generate new solutions

5) Evaluate solutions in the population

6) Repeat steps 3 through 5 until some convergence criteria are satisfied.

Genetic Algorithm in ANSYS DesignXplorer optimization module is used to find optimal solutions to the application problems in this research .Results of the proposed MQR optimization method is then compared with the results obtained from ANSYS

DesignXplorer and also from ANSYS First Order Optimization method, to check for the accuracy of the MQR method.

2.2.3 First Order Optimization Method

First Order optimization method as employed in ANSYS uses penalty function approach to minimize the objective or cost function. This is done by converting a constrained minimization problem into an unconstrained problem by adding penalty functions. The penalties are added to the objective function to account for the imposed constraints.

The first order method uses gradients of the dependent variables with respect to the design variables. Gradient calculations are performed in order to determine a search direction, and a line search strategy is adopted to minimize the unconstrained problem (ANSYS Documentation, [9]).

2.3 Pareto Optimality Criterion

Most real-life engineering optimization problems require simultaneous optimization of more than one objective function. In these cases, it is unlikely that the same values of the design variables will results in the best optimal values for all objectives. Hence, some trade-off between the objectives is needed to ensure a satisfactory design. This can be done mathematically correctly only when some optimality principle such as *Pareto Optimality* is used.

Pareto's principle states that: for a design variable **x** to be a Pareto optimal solution to a multi objective optimization problem, there should be no other solution that better satisfies all the other objectives simultaneously. That is, there can be other solutions that are better in satisfying one or several objectives, but they must be worse than the Pareto-optimal solution in satisfying the remaining objectives. Mathematically stated, a feasible solution **x** is called Pareto optimal if there exists no other feasible solution **y** such that $f_i(\mathbf{y}) \leq f_i(\mathbf{x})$ for $i = 1,2,3,\ldots\ldots,k$ with $f_j(\mathbf{y}) < f_j(\mathbf{x})$ for at least one j where $1 \leq j \leq k$, where k denotes the number of objective functions to be satisfied in a multi objective optimization problem.[1]

There are several methods that have been developed to solve multi objective optimization problems. Genetic Algorithm is one such method that has wide application in multi objective optimization problems. In this research Genetic Algorithm in ANSYS DesignXplorer is used to generate a set of Pareto optimal solutions and based on certain specific criterion or rule, the best Pareto optimal solution is selected as the solution for the optimization problem. The way this is being done would be discussed more in the chapter detailing results and discussion.

2.4 Optimization based on Finite Elements

Today's engineering structures are often analyzed using Finite Elements which is a well-known tool for structural analysis. Finite elements are applied to capture the dynamic response, heat transfer, fluid flow and other phenomena of a system and also to determine the deformation and stresses in a structure subjected to loads and boundary conditions. Mathematically it may be considered as a numerical tool to analyze problems governed by partial differential equations that describe the behavior of the system being studied.

Of all the engineering disciplines, structural designs have seen tremendous development and application of numerical optimization methods. It was Lucien Schmit in 1960 that recognized the potential for combining optimization techniques in structural design. He was the first to introduce nonlinear programming techniques to the design of elastic structures (Schmit, [10]).Today, various commercial finite element codes are available that have optimization capabilities inbuilt to it. Research is also being conducted to explore possibility of solving complex problems by integrating modern optimization tools like MATLAB with finite element packages such as ANSYS.

Optimization problems based on finite element can generally be expressed as :

$$\begin{aligned} &\text{Minimize} \quad f(\mathbf{x}, \mathbf{U}) \\ &\text{Subject to} \quad g_i(\mathbf{x}, \mathbf{U}) \le 0 \qquad i = 1,\ldots\ldots,m \\ &\text{And} \qquad h_j(\mathbf{x}, \mathbf{U}) = 0 \qquad j = 1,\ldots\ldots,l \end{aligned} \tag{2.4}$$

where **U** is an ($ndof \times 1$) nodal displacement vector from which the displacement field $\mathbf{u}(x, y, z)$ is readily determined. '*ndof*' refers to the number of degrees of freedom in the structure and **x** corresponds to the design variable set. It is to be noted here that **U** is an implicit function of **x** .i.e. any change made to the element parameter x_i will affect the displacement. The relation between **U** and **x** is governed by partial differential equations of equilibrium.

Based on Finite element theory these differential equations can be expressed as:

$$\mathbf{K}(\mathbf{x})\,\mathbf{U} = \mathbf{F}(\mathbf{x}) \tag{2.5}$$

where **K** is a ($ndof \times ndof$) square stiffness matrix and **F** is a ($ndof \times 1$) load vector. The functions f, g_i, h_j are all implicit functions of design variables **x**. They depend explicitly on **x**, and also implicitly through **U** as given by Eq. (2.5). [4]

2.4.1 Classification of Finite Element based Optimization problems

Depending on the type of design variables **x** finite element based optimization may be classified as *parameter or size, shape and topology* optimization. In ***parameter or size*** optimization the objective function f is typically the weight of the structure, and g_i are the constraints reflecting limits on stress and displacement. The design variable set **x** can take various forms. In the case of a pin-jointed truss section, x_i can be the cross sectional area or the length of the truss member. In the plane stress case or in a shell structure, x_i can be the thickness of each finite element used to mesh the region or in case of a beam cross-section x_i can represent moment of inertia. It is important to note that, when formulating any finite element based optimization problem, the constraints are to be expressed in normalized form. .i.e. If the stress developed in a structure has to be less that 10,000 psi, then the stress constraint can be expressed as:

$$\mathrm{g} \equiv \frac{\sigma}{10000} - 1 \leq 0 \tag{2.6}$$

This way it ensures that the constraints when satisfied will have values lying in the interval [0, 1].

Shape optimization problems deals with determining the outline of a body, shape and/or size of a hole, etc. In a '*sizing*' problem mesh geometry is unchanged as the parameters that are changed are those that affect **K** and **F** whereas in '*shape*' problems, the ***X, Y, Z*** coordinates of the nodes or grid points in the finite element model are changed iteratively. The main concept involved in shape optimization is *mesh parameterization* i.e. how can the coordinates of the grid points be related to a finite number of parameters. A common experience observed by analyst in shape optimization of CAD dependent parametric model is inconsistency of mesh pattern. If the mesh pattern cannot be kept, there may be unexpected variation of stress in addition to that caused by parametric changes. This observed phenomenon that usually causes problems in the optimization process is termed "stress oscillation". This difficulty is overcome by using CAD-Independent parametric modeling technique such as "Contour Natural Shape Function" (Chen, [11]). The main idea behind such an approach is to make parametric changes in the structure without changing the existing mesh connectivity and pattern.

Topology optimization on the other hand has to do with distribution of material, creation of holes, ribs or stiffeners, creation/deletion of elements, etc., in the structure. By contrast, in shape optimization of continua, the genus of the body is unchanged [4].By genus it means the number of cuts necessary to separate the body. While shape and size optimization is quite well known, topology optimization is beginning to gain its importance in commercial optimization codes. Ideally, shape, size and topology optimization should be integrated. However such a capability has been an area of current research.

CHAPTER 3

RESPONSE SURFACE METHODOLOGY (RSM) AND DESIGN OF EXPERIMENTS (DOE)

3.1 Introduction

Computer based simulation and analysis is used extensively in engineering for a wide variety of tasks. Despite the steady and continuing growth of computing power and speed, the computational cost of complex engineering analysis and simulations maintain pace. For instance, a crash simulation of a full passenger car can take more than 26 hours [3]. To minimize the computational expenses incurred by such analysis, resort on alternate methods of design and optimization such as Design of experiments and Response surface modeling is sought.

Response surface methodology was first developed by Box and Wilson in 1951 and can be defined as a method for constructing global approximations to system behavior based on results calculated at various points in the design space (Roux, [12]).The basic principle behind this method is to construct a simplified mathematical approximation of the computationally expensive analysis and simulation code and then use this code in place of the original code to facilitate multidisciplinary design optimization. Since the approximation model acts as a *surrogate* for the original code, it is often referred to as a surrogate model, surrogate approximation, approximation model or a metamodel [4]. Response surface methodology works well with *Design of* *experiments* and is valid only, over a part of the design space called the region of interest. An important objective in response surface construction is to achieve an acceptable level of accuracy while attempting to minimize the functional evaluations (computational cost). The accuracy of a response surface is based on two important factors: 1) the choice of the approximation function and 2) the selection of design points on the design space where design will be evaluated, i.e. the design of experiments (DOE). Increasing the number of experimental points or rather design points could improve accuracy .However it would be expensive to use a large number of points as the accuracy may be affected by other factors such as the order of the approximating functions, sub region size under investigation etc.

As mentioned before there are a number of methods to formulate the approximation function as well as to select the sampling points (DOE) for the response surface construction. In this study a variant of Radial basis function (RBF) called *multiquadric RBF* is employed to formulate the approximation function and *Quasi Monte Carlo* sampling technique is used to select the design points on the design space.

The following subsections and sections of this chapter gives the reader an idea about the pros and cons of employing RSM and DOE methods for design optimization, about commonly used methods to formulate approximation functions with an emphasis on RBF and Multiquadric RBF that is used in this study and a brief description about Quasi Monte Carlo sampling technique and why it was preferred.

3.1.1 "What is good" about DOE and RSM in Design Optimization

1) RSM attempts to substitute an original design optimization problem with computationally expensive functions by a sequence of much simpler problems where all the functions are approximated by response surfaces. This allows solving real life design optimization problems that are not solvable by other means within a reasonable amount of time.

2) The basic principle behind DOE and Response surface creation is very simple to be understood by any engineer. This is one of the reasons for its popularity.

3) Approximations based on Response surface methodology do not need design sensitivities but can use them if available, are insensitive to numerical noise and can be efficiently parallelized.

3) RSM with analytical models can be used more in the design process, even without optimization. This way visualization of the design space becomes possible.

4) RSM based optimization is beneficial for robust design optimization.

3.1.2 "What's not so good" about DOE and RSM in Design Optimization

1) When the number of design variables is large in case of global approximations, the use of RSM leads to excessive function evaluations.

2) In case of pure physics based problem, the response surface model cannot be treated as an approximation. This is because even though it can interpolate reasonably well, the quality of extrapolation beyond the available data may be poor.

3.2 Meta-modeling Methodologies

A variety of meta-modeling techniques exist. Quite popular meta-modeling techniques that are widely used for design simulation and optimization are: polynomial regression; multivariate adaptive splines (MARS); Radial basis function (RBF) and Kriging models. Of these Radial basis function originally introduced by Hardy in 1971 and MARS have began to draw a lot of attention of researchers (Jin, [13]).

There are a multiple factors that contribute to the success and effectiveness of a given approximation model, ranging from the nonlinearity and dimensionality of the problem to the associated data sampling technique and internal parameter settings for the various modeling techniques. Overall the knowledge of the performance of different approximation models with respect to different modeling criteria is utmost importance to designers when trying to choose an appropriate technique for a particular application.

A comparison of the four promising techniques mentioned above in a study by Jin, Chan and Simpson [4], revealed that radial basis function outperforms the other three methods in terms of robustness and accuracy, while solving high-order non linear problems. Moreover the impact of sample size on robustness and accuracy of a RBF was proved to be the smallest [4]. Radial basis functions have also been shown to produce good fits to arbitrary contours of both deterministic and stochastic response functions (Powell, [14]). RBF was also used to study the Multi-objective crashworthiness optimization of a car (Fang, [15]). On account of all these factors, it was decided in this study to use a form of RBF called *Regulated Multiquadric RBF* for creating the approximation models of the design cases studied. Above all, applying a

RBF is relatively straight forward and no parameters need to be specified by the user. They are considered very easy to implement.

RBF like any other meta-modeling technique can be summarized as a process of creating an approximation model using function values at some sampling points, which are typically determined using design of experiments (DOE) methods such as factorial design, central composite design, Taguchi orthogonal array or the Quasi Monte Carlo technique that has been employed in this research. In the following subsection a brief overview of the RSM and RBF that are of interests to this study are given.

3.2.1 Second Order Response Surface Approximation

A commonly used RSM methodology in engineering design optimization is the second order response surface approximation model. It’s based on a quadratic polynomial, which provides an explicit relationship between the design variables and the response of interest. The unknown coefficients in the model are approximated using the method of least squares. The ANSYS DesignXplorer used in this study employs the second order approximation method in response surface creation.

Let $f(x)$ be the true response function and $f'(x)$ its approximate obtained using the second-order RS model in the form

$$f'(x) = \beta_0 + \sum_{i=1}^{m} \beta_i x_i + \sum_{i=1}^{m} \beta_{ii} x_i^2 + \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \beta_{ij} x_i x_j \tag{3.1}$$

where m is the total number of design variables, x_i is the scaled value of the i^{th} design variable, and βs are the unknown coefficients. The scaling of design variables is done either between [-1, 1] or between [0, 1].

For n sampling of design variables x_i (k = 1, 2,….n, i = 1, 2,…m) and the corresponding function values f_k (k = 1, 2,….n), Eq. (3.1) leads to n linear equations expressed in matrix form as

$$\mathbf{f} = \mathbf{X}\hat{\boldsymbol{\beta}} \tag{3.2}$$

where the coefficient vector $\hat{\beta}$ is the least square estimator of the true coefficient vector and is solved using the method of least squares as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}(\mathbf{X}^{\mathrm{T}}\mathbf{f}) \tag{3.3}$$

There are many statistical evaluation tools to check the model fitness. The major statistical parameters for evaluating model fitness are the F statistic, R^2, adjusted R^2 (R^2_{adj}), and root mean square error ($RMSE$). These parameters are not totally independent of each other and are calculated as

$$F = \frac{(\mathrm{SST - SSE}) / \mathrm{p}}{\mathrm{SSE} / (\mathrm{n\text{-}p\text{-}1})} \tag{3.4}$$

$$R^2 = 1 - \mathrm{SSE} / \mathrm{SST} \tag{3.5}$$

$$R^2_{adj} = 1 - (1\text{-}R^2)\ \frac{n-1}{n-p-1} \tag{3.6}$$

$$RMSE = \sqrt{\frac{SSE}{n - p - 1}} \tag{3.7}$$

Where '*p*' is the number of non-constant terms in the RS model, *SSE* is the sum of square errors, and *SST* is the total sum of squares. Generally speaking, the larger the value of R^2 and R^2_{adj} , and the smaller the value of *RMSE*, the better the fit.

In addition to these statistics, the accuracy of the RS model can also be measured by checking its predictability of response using the prediction error sum of squares (*PRESS*) and R^2 for prediction ($R^2_{prediction}$) calculated as

$$PRESS = \sum_{i=1}^{n}[f_i - f'_{(i)}]^2 \tag{3.8}$$

$$R^2_{prediction} = 1 - PRESS / SST \tag{3.9}$$

where $f'_{(i)}$ is the predicted value at the i^{th} design point and f_i the true value.

3.2.2 Radial Basis Function

A radial basis function model uses a series of basis function that is symmetric and centered at each sampling point (scaled sample point). Let $f(x)$ represent the true function value and $f'(x)$ the approximate value found using RBF. Then

$$f'(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|) \tag{3.10}$$

Where n is the number of sampling points, $\boldsymbol{x}$ is the scaled vector of design variables, $\boldsymbol{x_i}$ is the scaled vector of design variables at the i^{th} sampling point, $\|x - x_i\|$ is the Euclidean distance, ϕ is a basis function and λ_i is the unknown coefficient. Here we can observe that an RBF is a linear combination of n basis functions with weighted coefficients. There are quite a few basis function used by RBF such as Thin-plate spline, Gaussian, Multiquadric and Inverse Multiquadric. This study has adapted an already developed MATLAB code based on multiquadric basis function to evaluate the approximation function '$f'(x)$'. In this work the design sample points generated using the quasi-Monte Carlo DOE technique is scaled between 0 and +1.

The general *multiquadric basis function* can be expressed as

$$\phi(\|x - x_i\|) = \sqrt{\|x - x_i\|^2 + h} \tag{3.11}$$

where '$\|x - x_i\|$' represents the Euclidean distance and h represents a parameter known as the shift parameter or the smoothness parameter where the value of h varies between 0 and 1. The multiquadric radial basis function is used because it is capable of providing an analytical multi-minimum approximation of the objective function with a limited number of sampling points (Hardy, [16]).

In regulated multiquadric basis function, the sum of squared errors (SSE) or the prediction error sum of squares (PRESS), used to calculate the accuracy of fit of the approximation model is augmented with a term '$r\sum_{i=1}^{n}\lambda_i^2$' which penalizes the large weights (coefficients). The parameter r called the regularization parameter controls the balance between fitting the data and avoiding the penalty. A small value of r means the data can be fit tightly without causing a large penalty. Hence a value of r = 0.001 is used for all the test cases studied in this research.

By replacing $\boldsymbol{x}$ and $\boldsymbol{f'(x)}$ in Eq. (3.10) with n vectors of design variables and their corresponding function values at the sampling points, we obtain a series of n equations which can be represented in the matrix form as

$$\boldsymbol{f} = \boldsymbol{A}\,\boldsymbol{\lambda} \qquad (3.12)$$

where $\boldsymbol{f} = [f'(x_1), f'(x_2), f'(x_3), \ldots\ldots]^T$, $A_{ij} = \phi(\|x_i - x_j\|)$ (i=1,2,….n , j =1,2,…n) and $\boldsymbol{\lambda}$ = ($\lambda_1, \lambda_2, \ldots\ldots \lambda_n$). The coefficient vector $\boldsymbol{\lambda}$ is obtained by solving Eq. (3.12).

The choice of shift parameter h can greatly affect the accuracy of the approximation. By adjusting the shift parameter, the accuracy of the approximation can be considerably increased. It has been found that by increasing the shift parameter, the RMS (root mean square) error of the fit dropped to a minimum and then grew rapidly thereafter (Kansa, [17]). Thus there exists an optimum value of shift parameter h that will yield minimum RMS for the fitted function. Efficient methods for parameter optimization in multiquadric approximation have been developed to compute the optimum shift parameter as to take the most advantage of the excellent performance of this process (Wang, [18]). In this research we have experimented with values of h between 0.3 and 1.

3.3 Design of Experiments

Design of experiments (DOE) can be defined as a procedure for choosing a set of samples in the design space, with the general goal of maximizing the amount of information gained from a limited number of samples (Giunta, [19]). One of the goals of a typical DOE study is to estimate and predict the trends in response data. Hence response surface approximations are often associated with design of experiments (DOE). As is the case with response function development, DOE study also involves scaling of the sample sets. In this study the bounds of the design variable that is significant in the sample point development is first scaled between 0 and +1 prior to the DOE study.

In general the DOE techniques can be classified as *classical* and *modern DOE* techniques. The classical DOE techniques were developed for laboratory and field experiments that possess random error sources while the modern DOE techniques pertain to deterministic computer simulations. Another feature that differentiates classical and modern techniques is the choice of probability distribution associated with design variables. In classical method the design variables are assumed to be uniformly distributed within the design bounds i.e. within the lower and upper bounds. In contrast, modern DOE methods are designed to handle both uniform and non-uniform distribution of design variables. A common characteristic shared by both the methods is the independent nature of the sampling points, which makes them amenable to concurrent evaluations.

Examples of classical techniques are central composite design, Box-Behnken design and full- and fractional-factorial design. These classical techniques work well when the sample points are put at the extremes of the design space. Examples of modern techniques are quasi-Monte Carlo sampling, Orthogonal array sampling, Latin hypercube sampling etc. The modern techniques are also known as space filling methods as they put the sampling points in the inner space as compared to the extremes of the design space in order to accurately extract the response trend information [19].

3.3.1 Comparison of Classical and Modern DOE Techniques

A measured response quantity in a classical DOE can be represented as

$$y_m(x) = y_t(x) + \varepsilon \tag{3.13}$$

where $y_m(x)$ represents the measured response, $y_t(x)$ represents the true response and ε represents the random error term in the Eq.(3.13). In classical DOE, the goal of minimizing the effects of random error has the affects of placing the samples on the boundaries and/or vertices of the design space and placing very few samples in the interior of the design space. This causes the interior of the design space to be largely unexplored. For example let us consider a classical DOE technique such as the *central composite design (CCD)* where the number of samples is given by the formula $2^n + 2n + 1$. Here n denotes the number of design variables or the number of dimensions in the design space. The 2^n samples in the formula correspond to the corners points of the design space, while the $2n$ samples correspond to the point lying outside the design space.

In case of a two dimension CCD problem where the design boundaries are scaled to range from 0 to +1, eight of the nine samples are on or outside the boundary of the design space, and only one sample, the center point , lies in the interior of the design space as shown below. As a result much of the response trend is unexplored in the interior of this design space.

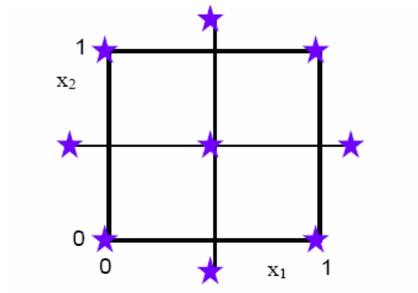


Figure 3.1 CCD samples for 2 design variable set

Another drawback of the classical technique is when the number of dimensions '*n*' is large. It can be seen that the number of samples in CCD scales as 2^n, a rate that can be unacceptable if *n* is large and/or if experiments are expensive [19] as is the case with finite element design optimization. On account of all these factors a modern technique called quasi-Monte Carlo sampling based on *Halton* sequencing is used in this research work.

In modern DOE there is no notion of random errors i.e. a computer simulation always produces the same response for an input data irrespective of the number of simulation runs. When using modern DOE methods an assumption that, the repose trend is unknown is made in addition to the assumption that there is no random error. On account of this assumption modern DOE method tends to place the scaled sample points in the interior of the design space so as to minimize the bias error. These errors arise when there is a difference between the functional forms of the true response trend, and the functional form of the assumed or estimated trend.

3.3.2 Quasi-Monte Carlo Sampling

This research employs an already developed MATLAB code on quasi-Monte Carlo sampling based on Halton sequencing. The *central composite design* scheme is used by ANSYS DesignXplorer to generate automatic sample points in the optimization process. The quasi-Monte Carlo sampling is observed to perform better compared to the other modern methods of DOE. This section would provide a brief description of quasi-Monte Carlo sampling and its advantages over the other methods.

Quasi-Monte Carlo technique or, alternatively known as a low discrepancy sampling were developed for multidimensional integration. This method seeks to distribute sample sites evenly throughout the design space, but does not employ a regular grid or a Cartesian lattice of sample sites. The term discrepancy refers to a quantitative measure of how much the distribution of samples deviates from an ideal uniform distribution. Hence, low-discrepancy is a desired feature of this class of sampling methods. It’s observed that quasi-Monte Carlo sampling has a lower integration error and computational time compared to other Monte Carlo methods and are also best suited for higher dimensional design spaces since their error bounds are exactly known [19]. On account of all these factors quasi-Monte Carlo techniques have been used for generating DOE in this study.

CHAPTER 4

DESIGN OPTIMIZATION PROCESS IMPLEMENTATION

4.1 Introduction

As mentioned in the previous chapter advantage of already developed codes in MATLAB for generating the DOE points using quasi-Monte Carlo methods and creation of Response surface was used in the development of the main design optimization code. The objective of this work was to set up a design automation code in MATLAB to solve finite element optimization problems. The MATLAB code so developed for this purpose incorporates the DOE, the adapted Response surface codes, ANSYS the finite element solver and a suitable optimization tool for optimization. A system call command in MATLAB was employed to call ANSYS for each simulation run that involved calculation of the responses to the input design variables. ANSYS *Parametric Design Language (APDL)* was used to model the design problem. The APDL file included the definition of design problem (parametric model of the design), solution process and the post-processing phase where the responses are gathered and recorded for optimization process in MATLAB. This APDL file thus created is used by ANSYS running in batch mode, during each system call command from MATLAB. The entire implementation was performed on a Windows based computer with a Pentium 4 processor and 512 MB RAM.

4.1.1 Development of ANSYS APDL file

The ANSYS APDL file was developed such that it could handle multiple sets of input design variables at a time during each system call. This way we could save the time taken to start and restart the ANSYS batch process for each design variable set. In short a population of data was send to ANSYS during the system call by MATLAB for the DOE study. The solution process involves iteration over each population set until the specified number of population size is met.

In the post-processing or the results recording phase of the analysis, the solutions are arranged in a matrix of size corresponding to (responses per iter × popsize) and the matrix thus formed is saved as a file "Pdata.var" .This file is then read by MATLAB for the response function creation phase.

4.1.2 Algorithm to find the Optimal Weight or Coefficient Matrix

The accuracy of such an optimization process mainly depended on how accurate the response surface modeling tool can model the approximation function used for the original optimization. This in fact depends on the choice of the weight or coefficient matrix $\boldsymbol{\lambda}$ used to model the approximation function as shown by Eq. (3.12).

As we know, a radial basis function model uses a series of basis function that is symmetric and centered at each scaled sampling point. In this study the design samples were scaled between [0, 1] for both the DOE and response function creation. If $f(x)$ represents the true function value and $f'(x)$ the approximate value found using RBF, then $f'(x)$ is given by the Eq. (3.10). For a multiquadric radial basis function

'$\phi(\|x - x_i\|)$' in Eq. (3.10) is given by $\phi(\|x - x_i\|) = \sqrt{\|x - x_i\|^2 + h}$, where '$\|x - x_i\|$' represents the Euclidean distance calculated using the scaled set of sample design points and h is a parameter known as the shift or the smoothness parameter whose value lies between [0,1]. Eq. (3.10) can also be written as

$$f'(x) = \sum_{j=1}^{m} \lambda_j \phi(\|x - x_j\|) = \sum_{j=1}^{m} \lambda_j h_j(x) \tag{4.1}$$

where $h_j(x) = \phi(\|x - x_j\|)$. In general matrix notation, the Eq. (4.1) takes the form

$$\mathbf{f} = \mathbf{H}\hat{\boldsymbol{\lambda}} \tag{4.2}$$

where $\mathbf{f} = [f'(x_1), f'(x_2), f'(x_3), \ldots\ldots f'(x_n)]^T$, $\boldsymbol{H}_{ij} = \phi(\|x_i - x_j\|)$ (i=1,2,….n , j =1,2,…m) and $\hat{\boldsymbol{\lambda}} = (\lambda_1, \lambda_2, \ldots\ldots \lambda_m)$. Here λ is the unknown coefficient matrix whose optimum values needs to be found out for a near accurate approximation of the response function '**f** '. A parameter called prediction error sum of squares is used to measure the accuracy of fit of a response surface model. In the case of a regulated multiquadric RBF this parameter is given by

$$PRESS = C = \sum_{i=1}^{n} [f(x_i) - f'(x_i)]^2 + r \sum_{j=1}^{m} \lambda_j^2 \tag{4.3}$$

where r denotes the regularization parameter. The objective is to find the minimum *PRESS* parameter that would give a good approximate of the response function. As is known from elementary calculus to find an extremum of a function, we have to 1) differentiate the function with respect to the free variables, λ in this case, 2) equate the result with zero and 3) solve the resulting equations. Let us carry out this optimization for the j-th weight, first differentiating ***C*** with λ we get

$$\frac{\partial C}{\partial \lambda_j} \;=\; 2\sum_{i=1}^{n}[f^{'}(x_i)-f(x_i)]\;\frac{\partial f^{'}(x_i)}{\partial \lambda_j}+2\;r\;\lambda_j \tag{4.4}$$

The derivative of $f^{'}(x_i)$ can be easily obtained from Eq. (4.1) and is given by

$$\frac{\partial f^{'}(x_i)}{\partial \lambda_j} \;=\; h_j(x_i) \tag{4.5}$$

Substituting this into Eq. (4.4) and equating the result to zero leads to the equation

$$\sum_{i=1}^{n} f^{'}(x_i)\;h_j(x_i)+r\;\lambda_j \;=\; \sum_{i=1}^{n} f(x_i)\;h_j(x_i) \tag{4.6}$$

There is m such equations, for $1 \le j \le m$, each representing one constraint on the solution. Since there are exactly as many constraints as there are unknowns, the system of equations has a unique solution. The above equation in matrix form can be written as

$$\mathbf{h_j^T}\,\mathbf{f}+r\;\lambda_j \;=\; \mathbf{h_j^T}\;\hat{\mathbf{Y}} \tag{4.7}$$

where

$$\mathbf{h_j} \;=\; \begin{bmatrix} h_j(x_1) \\ h_j(x_2) \\ \ldots\ldots \\ h_j(x_n) \end{bmatrix},\;\; \mathbf{f} \;=\; \begin{bmatrix} f^{'}(x_1) \\ f^{'}(x_2) \\ \ldots\ldots \\ f^{'}(x_n) \end{bmatrix},\;\; \hat{\mathbf{Y}} \;=\; \begin{bmatrix} f(x_1) \\ f(x_2) \\ \ldots\ldots \\ f(x_n) \end{bmatrix}$$

Since there is one of these equations for each value of j from 1 up to m, we can stack them one on top of the other to create a relation between two vector quantities as given below

$$\begin{bmatrix} \mathbf{h}_1^T\,\mathbf{f} \\ \mathbf{h}_2^T\,\mathbf{f} \\ \cdots\cdots \\ \mathbf{h}_m^T\,\mathbf{f} \end{bmatrix} + \begin{bmatrix} r\,\lambda_1 \\ r\,\lambda_2 \\ \cdots\cdots \\ r\,\lambda_m \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1^T\,\hat{\mathbf{Y}} \\ \mathbf{h}_2^T\,\hat{\mathbf{Y}} \\ \cdots\cdots \\ \mathbf{h}_m^T\,\hat{\mathbf{Y}} \end{bmatrix} \tag{4.8}$$

This is just equivalent to

$$\mathbf{H}^{\mathbf{T}}\,\mathbf{f} + \mathbf{\Lambda}\,\hat{\boldsymbol{\lambda}} = \mathbf{H}^{\mathbf{T}}\,\hat{\mathbf{Y}} \tag{4.9}$$

where $\mathbf{H} = \left[\mathbf{h}_1\,\mathbf{h}_2 \quad \cdots \mathbf{h}_m\right]$ and $\mathbf{\Lambda} = r\,\mathbf{I}_m$. Here $\mathbf{I}_m$ is an identity matrix of length m. Substituting for vector **f** in the above equation from Eq. (4.2) we get

$$\begin{aligned} \mathbf{H}^{\mathbf{T}}\,\hat{\mathbf{Y}} &= \mathbf{H}^{\mathbf{T}}\,\mathbf{f} + \mathbf{\Lambda}\,\hat{\boldsymbol{\lambda}} \\ &= \mathbf{H}^{\mathbf{T}}\,\mathbf{H}\,\hat{\boldsymbol{\lambda}} + \mathbf{\Lambda}\,\hat{\boldsymbol{\lambda}} \\ &= \left(\mathbf{H}^{\mathbf{T}}\,\mathbf{H} + \mathbf{\Lambda}\right)\hat{\boldsymbol{\lambda}} \end{aligned} \tag{4.10}$$

The solution to which is

$$\hat{\boldsymbol{\lambda}} = \left(\mathbf{H}^{\mathbf{T}}\,\mathbf{H} + \mathbf{\Lambda}\right)^{-1}\,\mathbf{H}^{\mathbf{T}}\,\hat{\mathbf{Y}} \tag{4.11}$$

It is to be noted that all the values on the R.H.S of the above equation is known. Hence from this equation optimal value of the coefficient or the weight matrix $\hat{\boldsymbol{\lambda}}$ that gives a good approximate of the response function can be found. Once the optimal coefficient matrix $\hat{\boldsymbol{\lambda}}$ is known, the approximate function can be found using Eq. (4.2). This is the algorithm that has been employed in the regulated MQR MATLAB code.

4.1.3 Algorithm of the Design Automation Process

The steps involved in the design automation process are discussed in this section followed by a general layout that depicts the optimization process. The first and foremost step before implementing this process is to understand thoroughly the design problem being studied and then to generate an ANSYS parametric design language (APDL) file of the design problem, that defines the preprocessor ,solution and postprocessor phase of the finite element analysis. In the preprocessor phase, the geometry and the boundary conditions of the model are defined using a set of APDL commands. This model that has been defined in the preprocessor phase is then solved in the solution phase using a suitable solution process such as STATIC, HARMONIC, THERMAL, etc. depending on the nature of the responses sought. In the postprocessor phase the results from the analysis is gathered and presented in a suitable format (file). This result file forms the input to the main optimization process.

The steps followed in the optimization process is as given below

1) *Definition of the design problem*: This step includes as mentioned before, generating an APDL file that defines and solves the problem, specifying the initial input data to run the optimization process such as the bound of the design variables (XL,XU), total number of samples to be generated (NC), Initial number of samples to specify the response surface (N0), Initial value of the design variables for optimization (X0) etc.
2) *Design of Experiments (DOE)*: This step can be considered as a two phase process. In the first phase the 'QMCSamples.m' file which is an already developed DOE

code in MATLAB on quasi-Monte Carlo sampling based on Halton sequencing, is used to generate an input data sample (XdataAll) of size NC. Of these a sample (Xdata) of size N0 is selected to undergo the first simulation run using ANSYS .The measured responses of these samples are stored in a matrix called Ydata. A matrix 'XCentre= Xdata' is defined as the set of centre points about which the approximation functions are built

3) Once this is done an iteration of length NMAX=NC-N0 is started. A counter 'i' is initialized a value zero at the beginning of the iteration and at each iteration its value is incremented by one.

4) *Response surface creation*: At the beginning of this step Xdata matrix is updated by picking a sample from the (N0 + i^{th}) row of the matrix XdataAll. The true response of this picked sample is calculated using ANSYS. The true response matrix 'Ydata' is updated using the responses of the newly picked sample. The updated response matrix 'Ydata',the Xdata matrix ,the already defined XCentre matrix, the lower and upper bounds of design variables and the values of the regularization and smoothness parameter are fed into the MQRFunN.m MATLAB code that evaluates the approximation to the objective and constraint functions. Algorithm described in the previous subsection is employed by this MATLAB code. The result from this code also contains the approximation errors and also the matrix containing the optimum value of the coefficient or weight matrix.

5) *MQR optimization process*: The optimum coefficient matrix evaluated in the previous step is used to generate the approximation functions of the data points

generated as a part of the optimization process. The approximation functions are found using the Eq. (4.2). The input data points to the optimization process are first scaled between 0 and +1. The basis functions for this input points are then evaluated using the formula given by the Eq. (3.10). These basis functions are used in conjunction with the optimum weight coefficients to formulate the approximate response functions used by the optimization process. The input parameters for the optimization are the starting vector of design variables (X0), upper and lower bounds of design variables, the output from MQRFunN.m etc. An interesting thing noted here is the definition of the optimization problem that is defined to solve multi minima problems. Lets consider the figure shown below

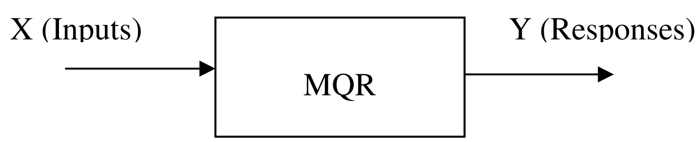


Figure 4.1 Block diagram of an MQR process

Let X be the input design variable and Y be the approximated value of the responses. Then the optimization problem to be solved can be defined as find X

$$\text{to minimize } \mathbf{f} = [\mathbf{AxObj}]\,[X] + [\mathbf{AyObj}]\,[Y] \tag{4.1}$$

$$\text{Such that } \mathbf{g} = [\mathbf{AxCon}]\,[X] + [\mathbf{AyCon}]\,[Y] - \mathbf{B} \leq \mathbf{0}$$

where **AxObj, AyObj** are the matrix that defines the input and output objectives and **AxCon** and **AyCon** and **B** are the matrix that defines the input and output constraints to the problem. This is best explained by a simple problem of finding the minimum weight of a two dimensional two bar truss fixed at the two ends as shown below.

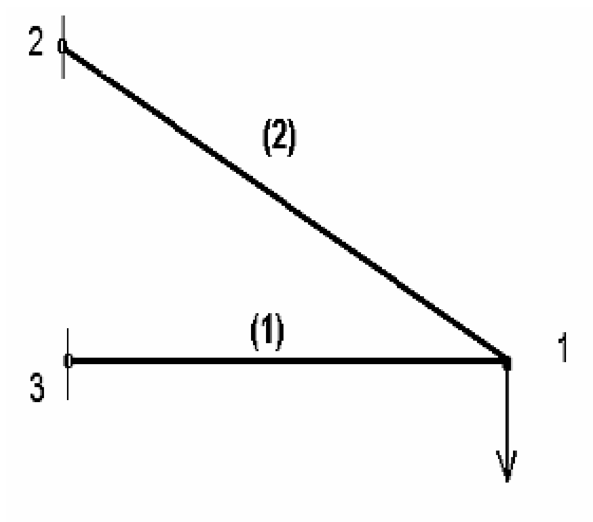


Figure 4.2 Two bar truss problem

Let A1 and A2 are the two design variables that represent the cross sectional areas of the truss members. The objective is to find the minimum weight (volume) so that element stress is less than a specified allowable value σ_{All}. The stress constraints can be formulated as $\sigma \leq \sigma_{All}$ or $(\sigma/\sigma_{All})-1 \leq 0$. For this problem the MQR block diagram is as shown below.

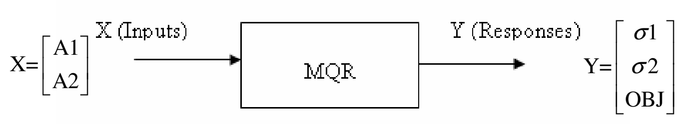


Figure 4.3 MQR Block diagram for 2 bar truss

The matrices that define the objective and constrain function as given by Eq. (4.1) are defined as

$$\text{AxObj} = \begin{bmatrix} 0 & 0 \end{bmatrix},\ \text{AyObj} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},\ \text{AxCon} = \begin{bmatrix} 0\,0 \\ 0\,0 \\ 0\,0 \end{bmatrix},\ \text{AyCon} = \begin{bmatrix} 1/\sigma_{\text{All}} & 0 & 0 \\ 0 & 1/\sigma_{\text{All}} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Once the optimization is performed, the optimum value is recorded and is updated to the Xdata matrix. The true response to this optimum value is calculated using ANSYS simulation and this value is updated to the response matrix Ydata.

6) *Convergence Check*: This step involves defining the convergence of the optimization problem based on certain criteria. The criteria used are,1) the difference of the objective function values for two successive iterations should be less than or equal to a tolerance specified by fEPS and 2) the maximum value of the constraint should be less than or equal to a value gEPS. Both these parameters fEPS and gEPS are user defined.

7) If it is observed that the problem does not converge at the obtained optimum, then steps 4 through 6 are repeated until a convergence is reached.

The above steps are summarized in a work process flow chart that depicts a general layout of the design automation process developed in this research. Figure 4.4 shows the general layout of the optimization process.

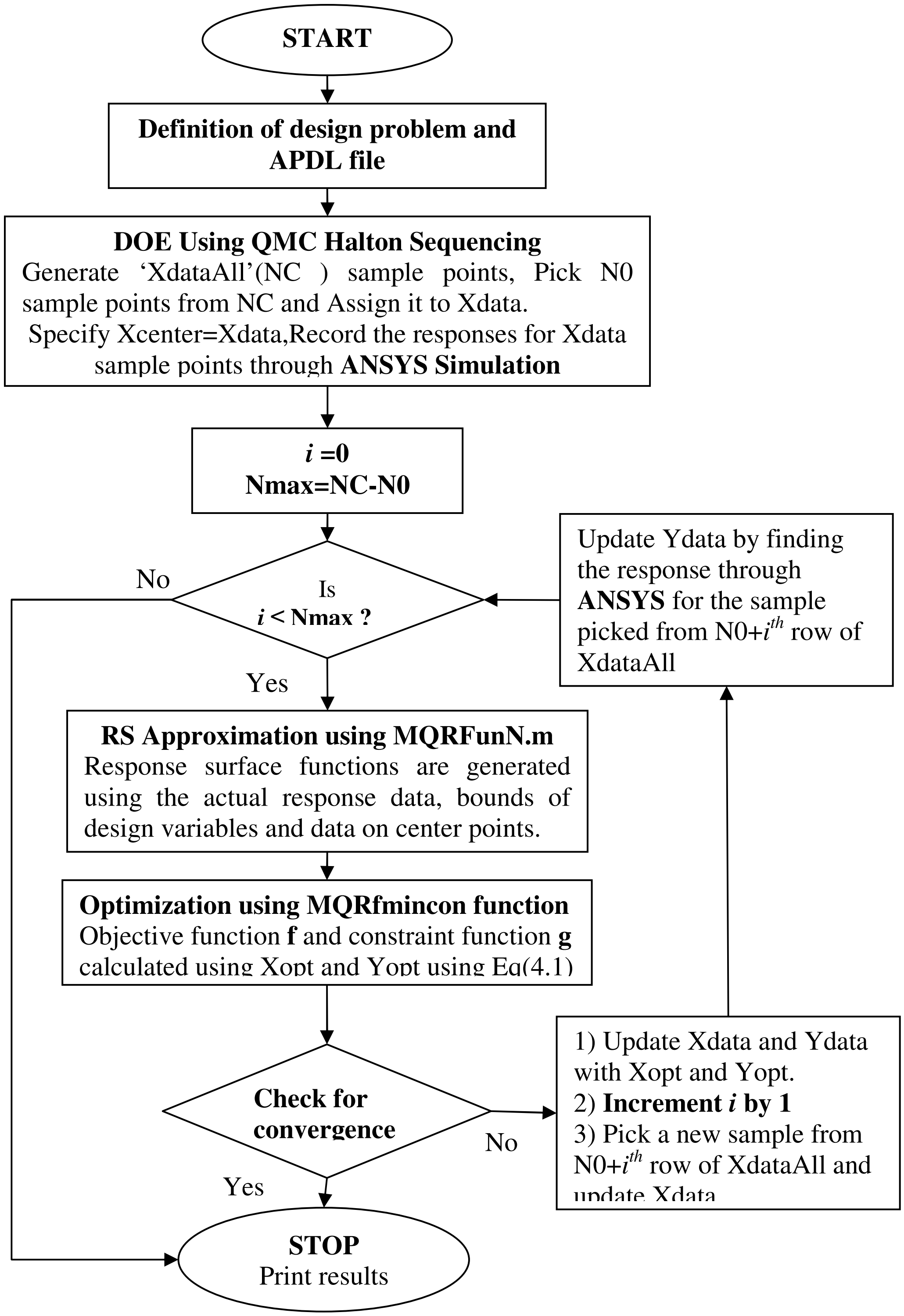


Figure 4.4 General layout of MQR optimization process

CHAPTER 5

APPLICATIONS

5.1 Introduction

In this section we will discuss the results obtained implementing the MQR optimization algorithm to solve a selected set of application problems. The application problems studied are

5. Weight minimization of 9-bar truss, by finding optimal cross-sectional areas of the truss members.
6. Weight minimization of 25-bar truss, by finding optimal cross-sectional areas of the truss members.
7. Finding the optimal size of an elliptical hole in a rectangular plate for minimum weight so as to withstand the applied tensile load.
8. Optimal design of a 2 dimensional vehicle suspension for ride quality and comfort.

The first three problems belong to size optimization category while the last problem deals with dynamic response optimization. The results obtained solving these optimization problems using the proposed MQR algorithm is then compared with the results from ANSYS first order conventional optimization technique and also with ANSYS DesignXplorer optimization method. The ANSYS DesignXplorer employs a similar scheme of approaching optimization problems using DOE and response function modeling methods.

However the ANSYS first order optimization is a conventional optimization method where the true functions (Objective and Constraints) are used for the optimization. An APDL file that defines the pre-processing, solution, post-processing and the optimization phase is used for the conventional ANSYS optimization. In the optimization phase, the algorithm for optimization, the number of iterations ,the upper and lower bounds of the design variables and the limits on constraints are specified. The process converges within the specified iteration if a minimum value of objective function has been found that obeys the given constraints. If it does not converge a second iteration has to be setup with a new starting point. This goes on until a convergence is met within the specified constraints.

For the ANSYS DesignXplorer the procedure for optimization can be summarized in the following steps.

1. Read the ANSYS APDL file into DesignXplorer and record the input and output parameters.
2. Through DesignXplorer central composite DOE scheme create candidate designs (Automatic design points).
3. Create response surface using second order polynomial based regression analysis using the candidate designs and the true responses.

4. Define design goals for the optimization such as allowable constraints etc.
5. Create new design points through sample generation from the specified goals
6. Select the best candidate/candidates from tradeoff study and verify the validity of the candidate design points by running analyses on candidate designs in simulation and there by creating reference design points.

All the optimization tests for the application problems were carried out on a Pentium 4 based PC running on Microsoft Windows platform.

5.2 9-Bar Truss

9-bar truss was the first optimization problem to be studied in this research. The objective of this problem is to design a 9-bar Steel truss structure of minimum weight (volume) that can withstand applied loads within the limits of allowable stress and displacement. The material properties for steel are Young's modulus $EX = 2.973 \times 10^7$ lbf / in^2 and Poisson's ratio $PRXY = 0.3$. The applied forces are $f_1 = -5{,}000$ lbs, $f_2 = 2{,}000$ lbs and $f_3 = 7{,}000$ lbs. The lengths of the truss members are fixed. The design variables or the input variables for the optimization problem are the cross sectional areas of the truss members. The cross sectional areas of the truss members are allowed to vary between 0.1 and 2.0 inch^2. The stress limits are 10,000 psi in tension and compression and the allowable displacement is 0.05 inch in all direction.

The geometry of the problem is shown in Figure. 5.1. The structure is hinged at node 1 and roller supported as shown at node 4.

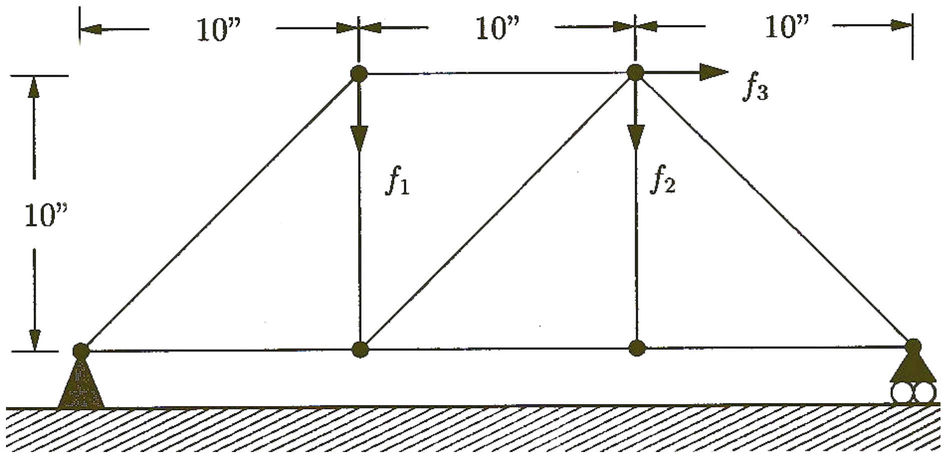


Figure 5.1 9-Bar truss

The truss members are classified into 2 element groups thus defining two design variables for the optimization problem. This classification is shown by the free body diagram from ANSYS as shown in Figure 5.2. ANSYS LINK1 element is used to model the truss elements for the simulation process.

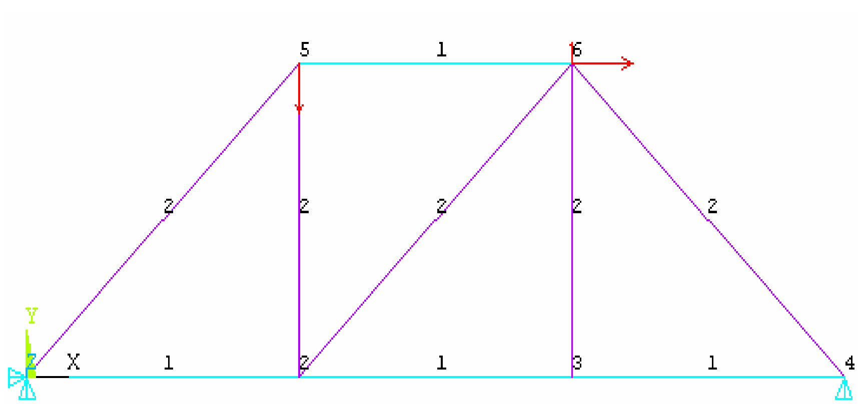
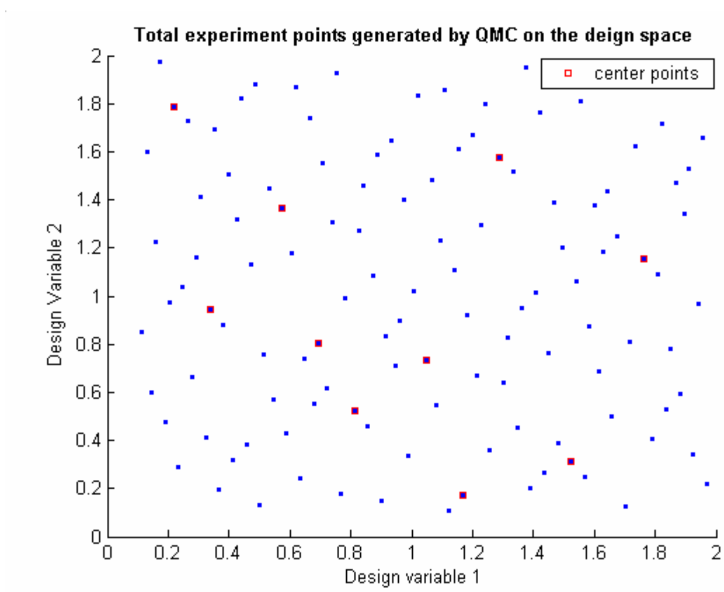


Figure 5.2 Free body diagram of 9-Bar truss

The values of the parameters that define the MQR optimization process are $h = 0.5$, $r = 0.001$, $fEPS = 0.01$ and $gEPS = 0.01$. The total number of samples defined for the DOE is $NC = 120$. The number of initial samples that is equal to the number of samples that define the center points is given by $N0 = 10$. Figure 5.3 shows the distribution of entire set of data points in the design space as sampled by the Quasi-Monte Carlo DOE scheme. Figure 5.4 shows the data points that have been ultimately used as the problem converged to an optimal solution. The cluster of points crowded together in this figure shows the convergence of the data points to an optimal value. At the end of the optimization it was found that after about 28 iterations the function value converged to a minimum total volume *TVOL* of 73.7662 in^3. The function convergence is shown in Figure 5.5. The optimum values of design variables are $A1 = 0.7538$ in^2 and $A2 = 0.6986$ in^2.



. Figure 5.3 Quasi-Monte Carlo sampling for 9-bar truss

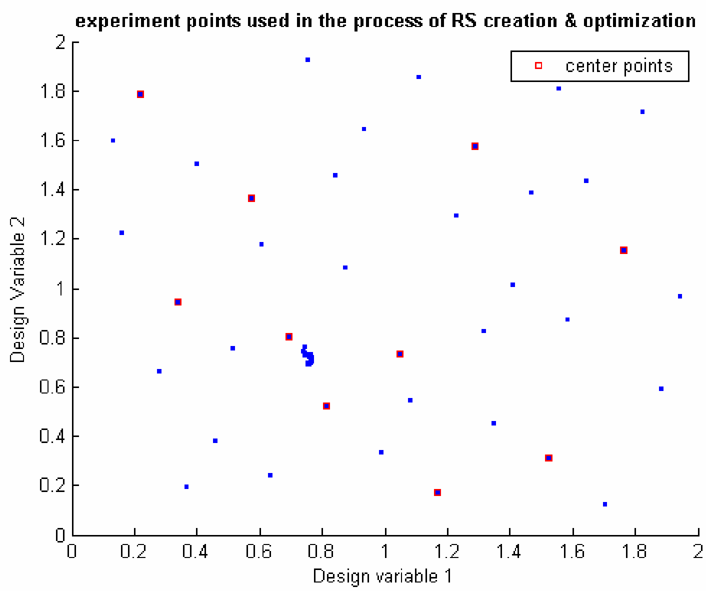


Figure 5.4 Data points used at the end of convergence

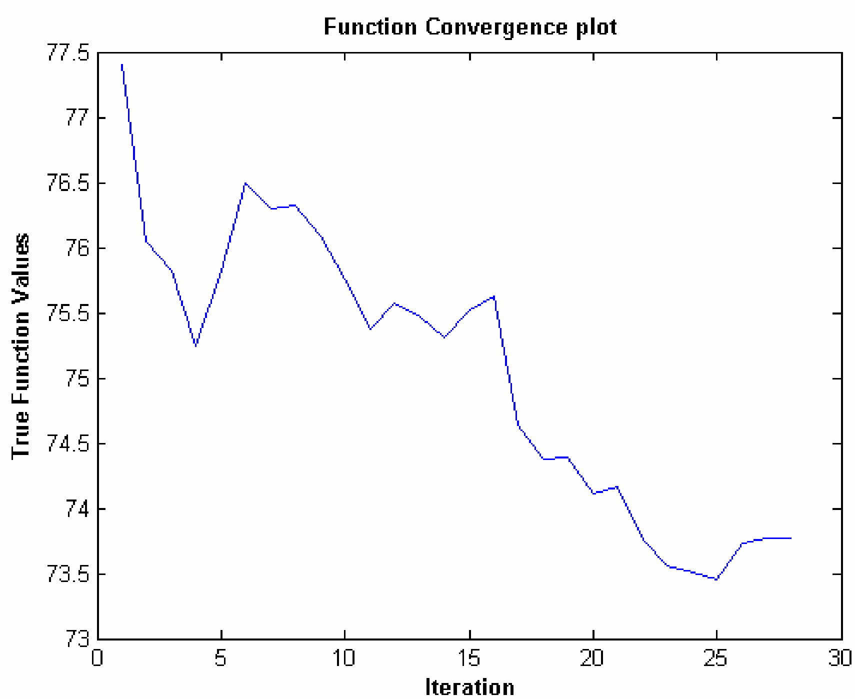


Figure 5.5 Function convergence plot for the MQR optimization process

The same problem was also solved using the ANSYS First order optimization method and with ANSYS DesignXplorer. The maximum number of iterations was set as 35 for the ANSYS First order optimization process. The optimization problem converged to a minimum value of objective function given by *TVOL* = 72.572 in^3. The optimal design variables were *A1* = 0.7546 in^2 and *A2* = 0.679 in^2. The function convergence plot for the ANSYS first order optimization process is shown in Figure 5.6.

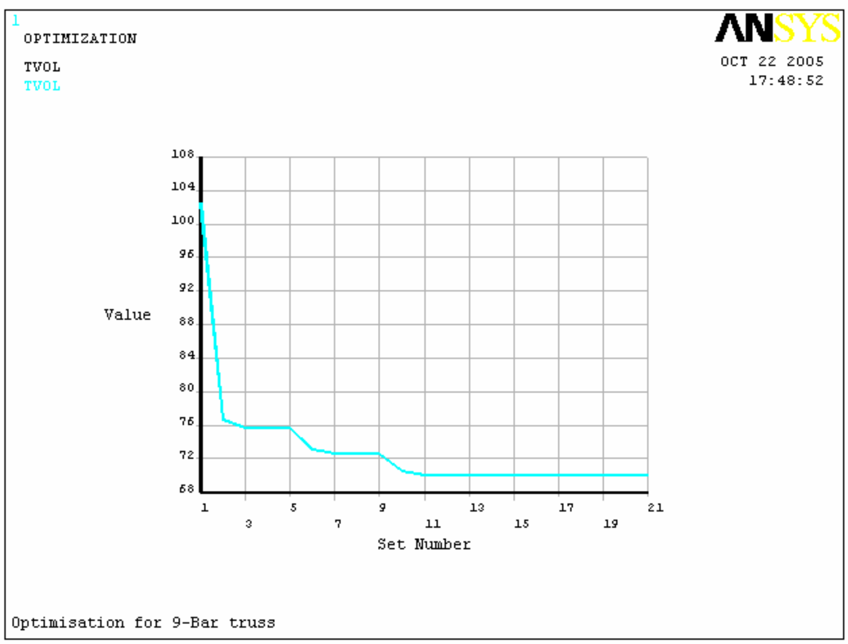


Figure 5.6 Function convergence 9-Bar truss- ANSYS First order optimization process

The results obtained for this problem using the ANSYS DesignXplorer goal driven optimization process is given in Figure 5.7. The generated response surfaces for the total volume, stress and displacement constraints are shown in Figure 5.8, Figure 5.9 Figure 5.10 and Figure 5.11.

Design Point 1

This design was rated BEST.

Parameter values

Name	Value	Rating
A1	0.76625	–
A2	0.68494	–
XMAX	7.0123e-003	★★★
YMAX	-7.4917e-003	★★★
SMAX	9993.5	–
TVOL	73.408	–

Reference Design Point 1

This design was not rated.

Parameter values

Name	Value	Rating
A1	0.76625	–
A2	0.68494	–
XMAX	7.0301e-003	★★★
YMAX	-7.4931e-003	★★★
SMAX	9635.4	★
TVOL	73.408	–

Figure 5.7 Results from ANSYS DesignXplorer goal driven optimization for 9-bar truss

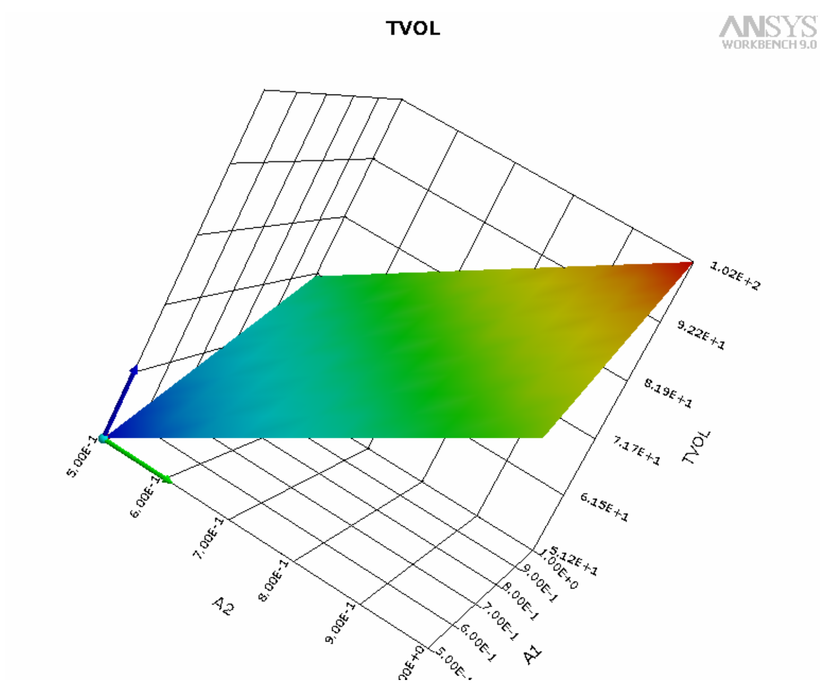


Figure 5.8 Response plot for total volume

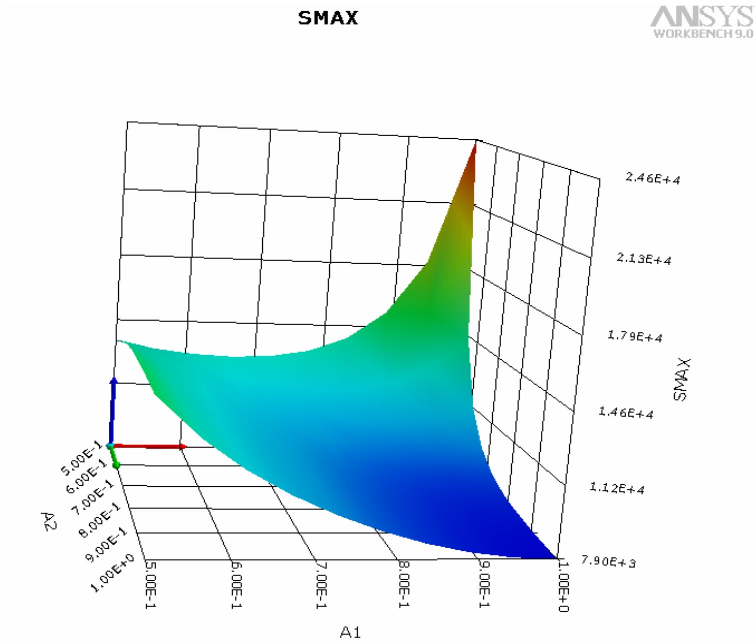


Figure 5.9 Response plot for maximum stress

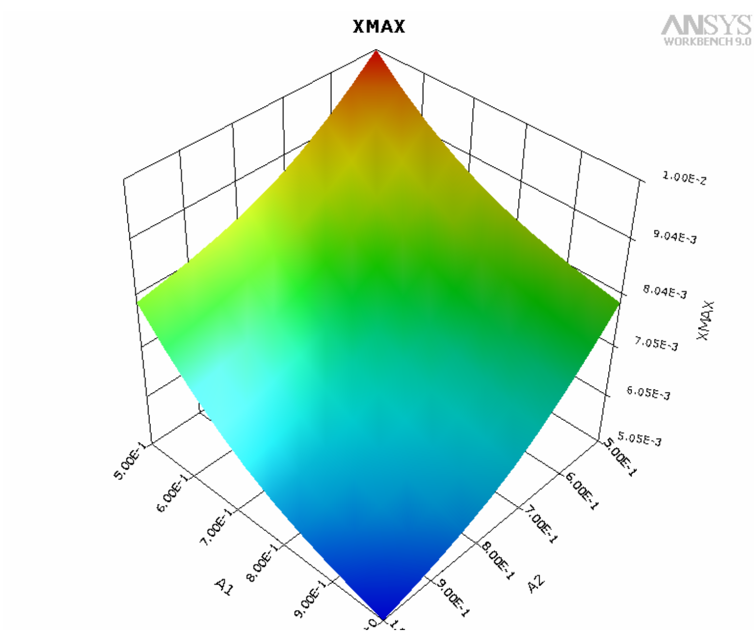


Figure 5.10 Response plot for maximum displacement of nodes along X direction

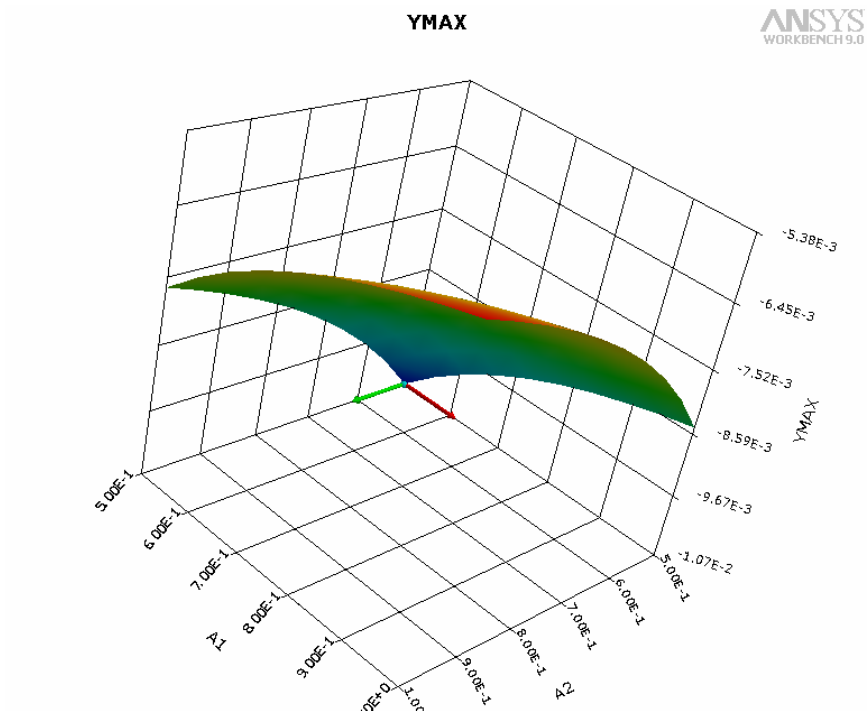


Figure 5.11 Response plot for maximum displacement of nodes along Y direction

The tradeoff plots that were generated as a part of the first and the last generation of sample points during the goal driven optimization is shown in Figure 5.12 and Figure 5.13. In Figure 5.12 those points that are represented by pyramids are infeasible design points in the generated samples with the red color being the worst. The points that are represented by blue color blocks are probable Pareto optimal points. These probable design points are selected and an advanced sample generation process based on genetic algorithm is performed till a good set of probable points are generated.

Figure 5.13 shows the final set of probable points from which the best design variable is selected and rated.

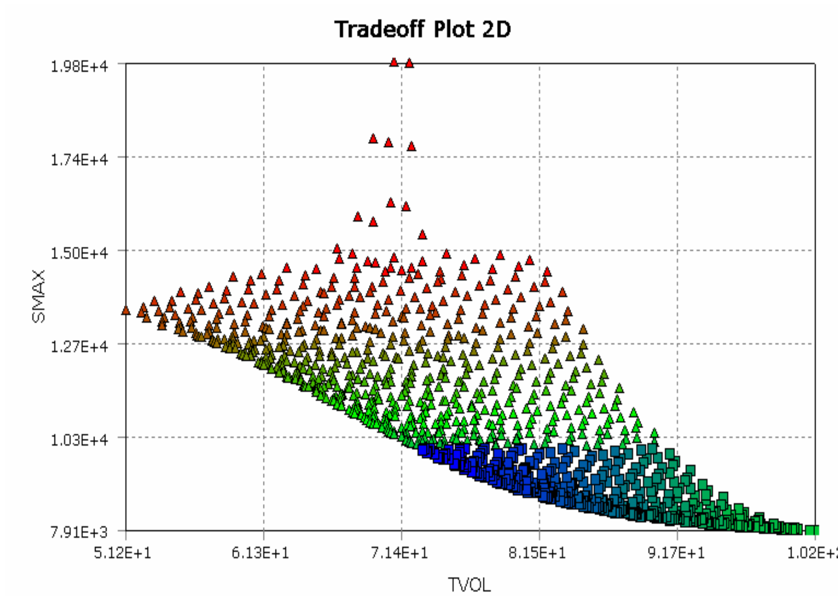


Figure 5.12 Tradeoff plot for first sample generation-9-Bar Truss

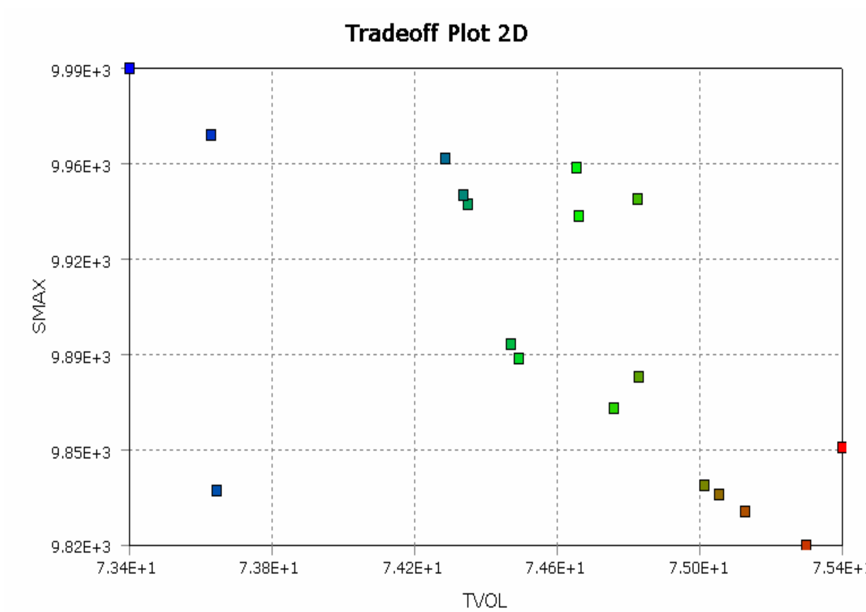


Figure 5.13 Tradeoff plot for last sample generation-9-Bar Truss

It is observed here that the results for the objective function of the MQR optimization process compare well with those of ANSYS First order optimization process and also with ANSYS DesignXplorer goal driven optimization.

5.3 25-Bar Truss

The 25-Bar truss structure is shown in Figure 5.14. The objective in this problem is to design a 25-Bar Aluminum truss structure for minimum weight (Volume) so as to withstand the given loads. The material properties are Young's modulus $EX = 1.0498 \times 10^7$ psi, Poisson's ration $PRXY = 0.3$ and weight density ρ of 0.1 lbf / in^3. The constraints specified for this problem are the stress constraints which define the element stresses to be within 40,000 psi in tension and compression. No displacement constraints at the nodes were specified. The length of the truss structure is constant and only the cross-sectional areas of the truss members are considered as the design variables. There are four loads acting on the structure .They are $f_y = 20{,}000$ lbs and $f_z = -5{,}000$ lbs acting on node 1 and $f_y = -20{,}000$ lbs and $f_z = -5{,}000$ lbs acting on node 2 respectively. ANSYS LINK8 element is used to model and simulate the structure for analyses. The truss members are divided into seven element groups as shown in Figure 5.15.

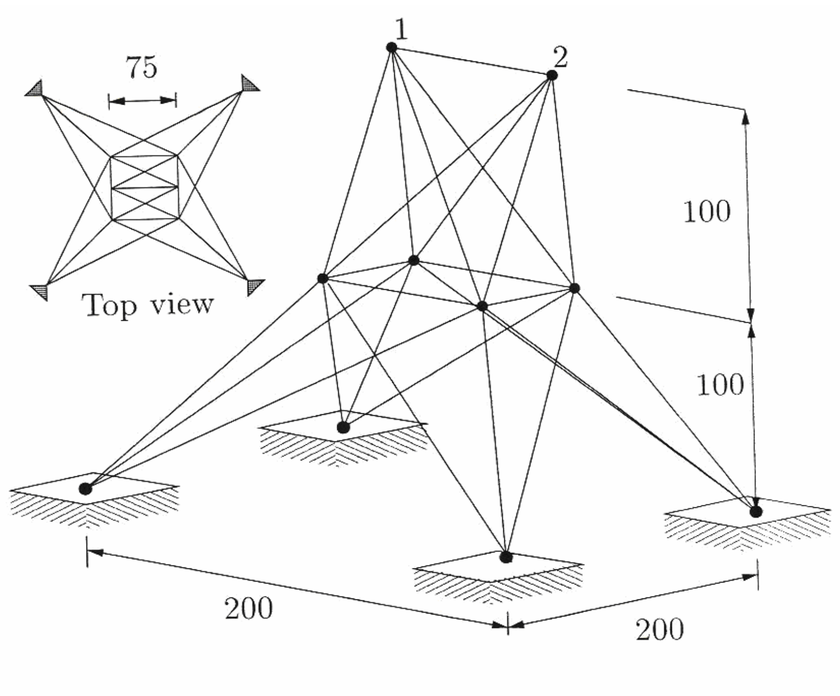


Figure 5.14 25-Bar truss

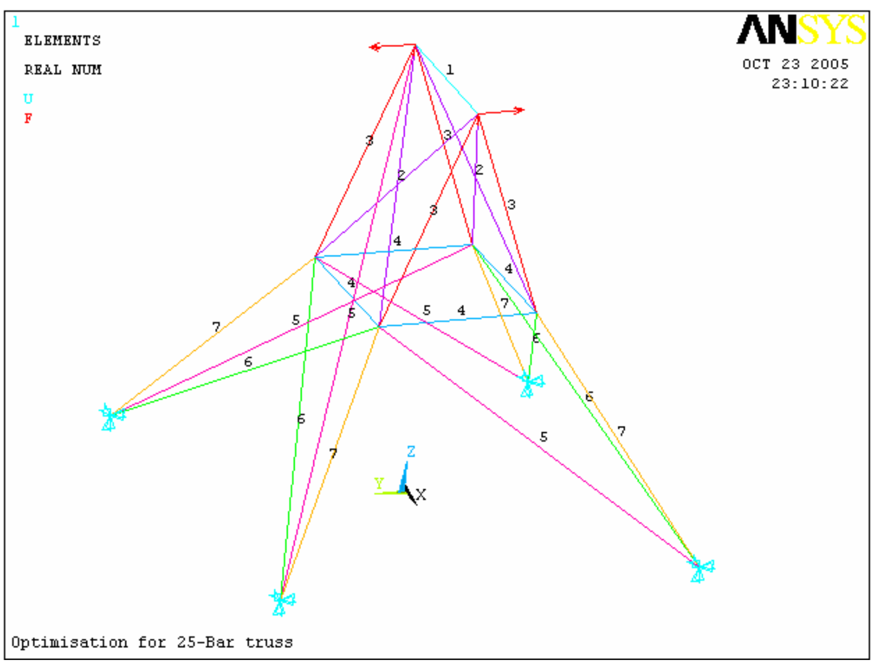


Figure 5.15 Element groups for the 25-Bar truss problem

The seven element groups signify seven design variables. The design space in optimization can be considered as a seven dimensional space where each design variable (cross-sectional area) represent seven coordinates of the design space. As was mentioned in the section detailing about optimization, it is difficult to visualize the sample points in a seven dimensional design space. In this optimization problem the design variables are allowed to vary between 0.1 and 2.0 inch^2.The values of the parameters that define the MQR optimization process are h = 0.8, r = 0.001, *fEPS* = 0.01 and *gEPS* = 0.01. The total number of samples defined for the DOE is *NC* = 120. The number of initial samples that is equal to the number of samples that define the center points is given by *N0* = 25. The convergence plot for the MQR optimization process is shown in Figure 5.16.

It is seen here that the objective function converges in 94 iterations to a minimum objective function value of *TVOL* = 1024.6833 inch^3 with the constraint function *SMAX* = – 39484 psi. The optimum values of the design variables are shown in Figure 5.17.

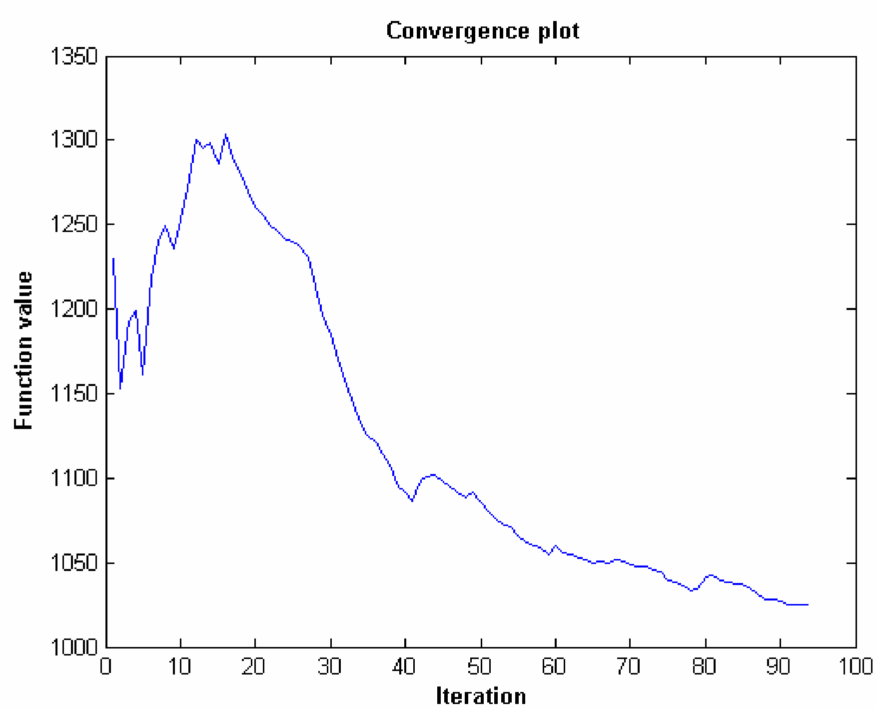


Figure 5.16 Function convergence plot for 25-Bar truss structure with 7 element groups

Xopt (in^2)	1.1151
	0.5896
	0.4471
	0.1
	0.1
	0.2766
	0.2459

Figure 5.17 Optimum values of the design variables

The same problem when solved using the ANSYS first order optimization gave a minimum value of the objective function *TVOL* = 1087.7 inch^3. The convergence plot for the ANSYS First order optimization and the optimum values of the design variables are shown in Figure 5.18 and Figure 5.19 respectively.

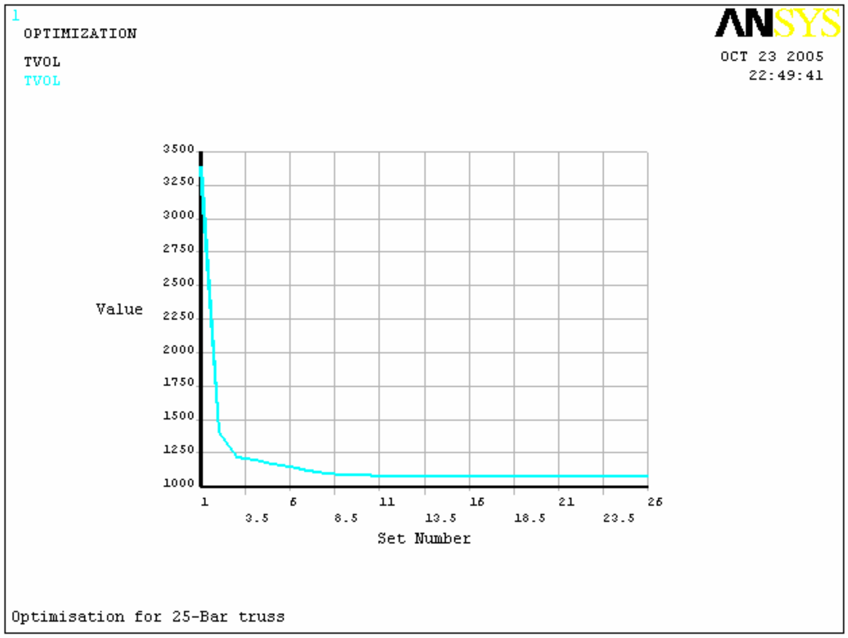


Figure 5.18 Function convergence plot in ANSYS for 25-Bar truss structure with 7 element groups

Iteration set # 9

<table>
<tr><td rowspan="7">Xopt (in^2)</td><td>0.90089</td></tr>
<tr><td>0.39732</td></tr>
<tr><td>0.44299</td></tr>
<tr><td>0.6042</td></tr>
<tr><td>0.1</td></tr>
<tr><td>0.27233</td></tr>
<tr><td>0.3</td></tr>
</table>

Figure 5.19 Optimum values of the design variables

As can be observed in the MQR optimization process there are two design variables out of seven that take the lower bound values permitted where as in ANSYS first order optimization there is only one design variable that has taken the lower bound value. The results of objective function from both optimization processes show close agreement with one another.

The results for the truss problem using the ANSYS DesignXplorer goal driven optimization are shown in Figure 5.20. Its seen here that the minimum value of the objective function is 1031.1 inch^3 .This value compares well with the one obtained using the MQR optimization process. However it took an initial sample size of 10,000 in DesignXplorer goal driven optimization to generate this result. This shows that MQR shows better accuracy with less number of samples.

Design Point 1

This design was rated BEST.

TABLE 7

Parameter values

Name	Value	Rating
A1	0.86585	−
A2	0.41975	−
A3	0.60869	−
A4	0.10373	−
A5	0.14729	−
A6	0.31549	−
A7	0.20348	−
SMAX	39924	−
TVOL	1031.1	★★

Reference Design Point 1

This design was not rated.

TABLE 6

Parameter values

Name	Value	Rating
A1	0.86585	−
A2	0.41975	−
A3	0.60869	−
A4	0.10373	−
A5	0.14729	−
A6	0.31549	−
A7	0.20348	−
SMAX	37202	−
TVOL	1031.1	★★

Figure 5.20 Results from DesignXplorer goal driven optimization for 25-bar truss problem -7 design variables

The tradeoff plots corresponding to the first and last sample generation of the goal driven optimization are shown in Figure 5.21 and Figure 5.22 respectively. As described in the previous problem the design points represented by pyramid or triangular blocks are infeasible points. The data points marked as blue blocks are considered as the best Pareto points within the design space. From the first generated sample set, the Pareto optimal points are selected and an optimization based on Genetic algorithm is performed to finally select the best candidate design. The results of the best candidate and its reference design were shown in Figure 5.20.

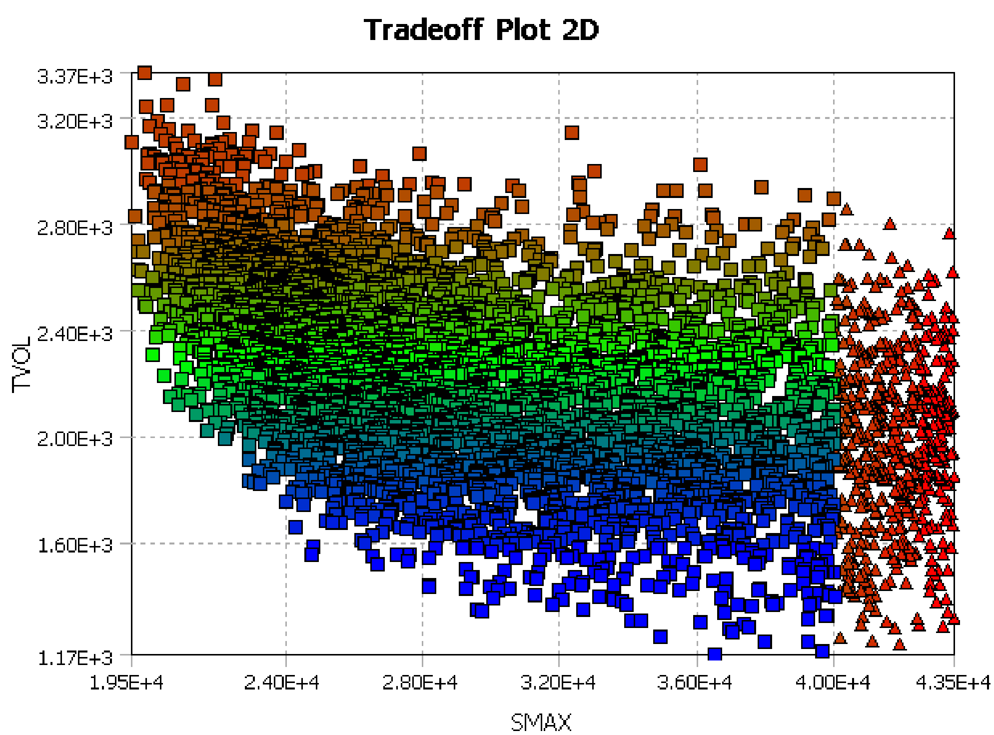


Figure 5.21 Tradeoff plot for first sample generation -25- Bar Truss

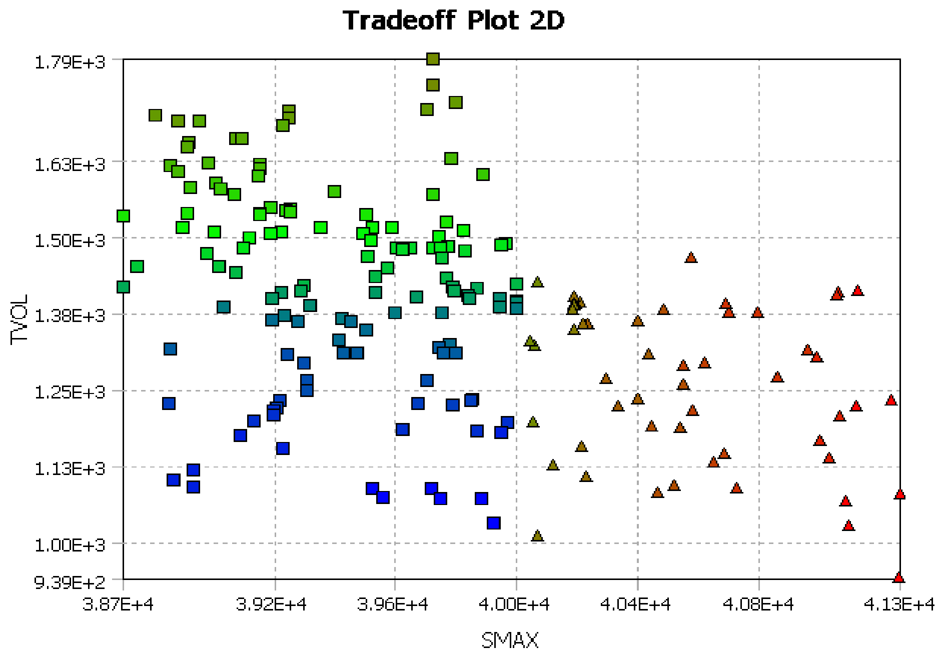


Figure 5.22 Tradeoff plot for last sample generation -25- Bar Truss

5.4 Plate with an Elliptical Hole

The rectangular plate containing an elliptical hole is shown in Figure 5.23. The plate is subjected to a tensile stress of 800 psi. The main objective is to find the size of the elliptical hole that can withstand the applied tensile stress. The plate is completely fixed along one of its shorter edges. The plate is considered to be made of Aluminum with material properties given as $EX = 1.0498\times10^7$ psi, Poisson's ration $PRXY = 0.3$ and weight density ρ of 0.1 lbf / in^3. The width of the plate is W = 100 inch, height of the plate H = 50 inch and thickness T = 1 inch.

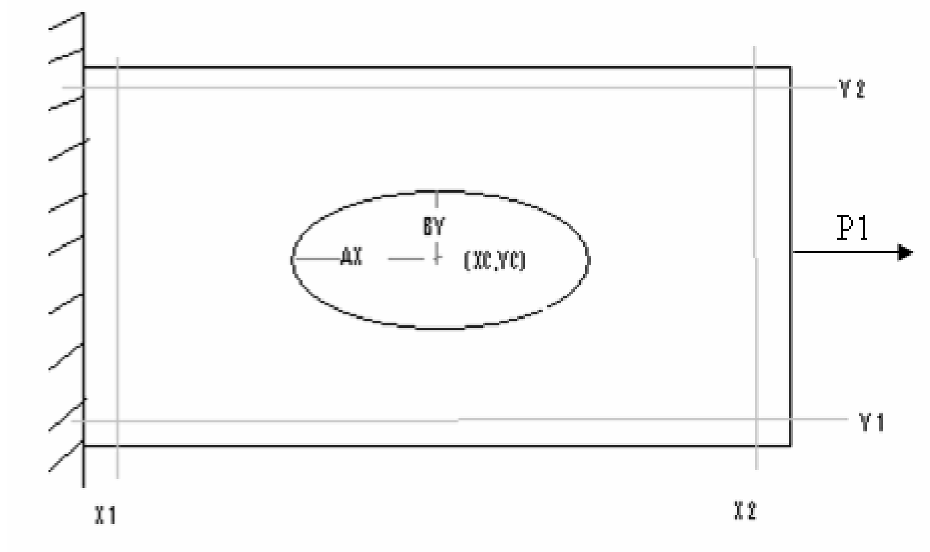


Figure 5.23 Plate with the elliptical hole subjected to tensile load

Using the MQR optimization process, two cases of the problem were studied. One where the design variables are the location of the centre of the ellipse (*XC, YC*) and the size of ellipse (*AX, BY*), i.e., there are four design variables. In the second case the center of the ellipse is fixed at the center of the plate and the only variables were (*AX, BY*). To solve the first case, a boundary has to be specified to constrain the hole within the plate during optimization. As observed in the Figure 5.23 the points (*X1, Y1*), (*X2, Y1*), (*X2, Y2*) and (*X2, Y2*) specifies the boundary within which the elliptical hole is allowed to move. In this test case we have specified a boundary where the walls of the boundary are 5 inches from the width and height of the plate i.e. *Xall* = *Yall* = 5 inches.

For both the test cases the objective of the problem is to minimize the weight of the plate such that the maximum stress developed as a result of the tensile pressure applied is within 14,000 psi i.e. $\sigma_{MAX} \leq \sigma_{ALL}$, where σ_{ALL} = 14,000 psi. Let's look at the problem formulation part of this test case. There are four design variables in the first test case. This involves proper selection of the design points. That means, the design points has to be chosen such that the elliptical hole is always within the specified boundary of the plate. This causes an additional set of constraints apart from the normal stress constraints to be included in the optimization problem. They are the input constraints that filter out those points that cause the elliptical hole to move out of the plate. The input constraint formulation is shown below.

$$X2 = W - Xall, \; Y2 = H - Yall$$
$$X1 = Xall, \; Y1 = Yall$$

Thus the input constraints are

$$\begin{aligned} XC + AX &\leq X2 \\ YC + BY &\leq Y2 \\ XC - AX &\geq X1 \\ YC - BY &\geq Y1 \end{aligned}$$

The values of the parameters that define the MQR optimization process are h = 1, r = 0.001, $fEPS$ = 0.01 and $gEPS$ = 0.01. The total number of samples defined for the DOE is NC = 150. The number of initial samples that is equal to the number of samples that define the center points is given by $N0$ =15. The bounds of the design variables in the order [*XC YC AX BY*] for the first case are lower bound LB = [5 5 1.5 1] inch and the upper bound UB = [95 45 45 20] inch. We know that for minimum weight the optimum value of the design variables should be [50 25 45 20] inches, provided the stress are within the allowable limit. It was observed here that with MQR optimization, the process converged to the minimum value of *TVOL* = 2364.5344 inch3 with the above mentioned optimum values of design variables in eight iterations as shown in Figure 5.24.

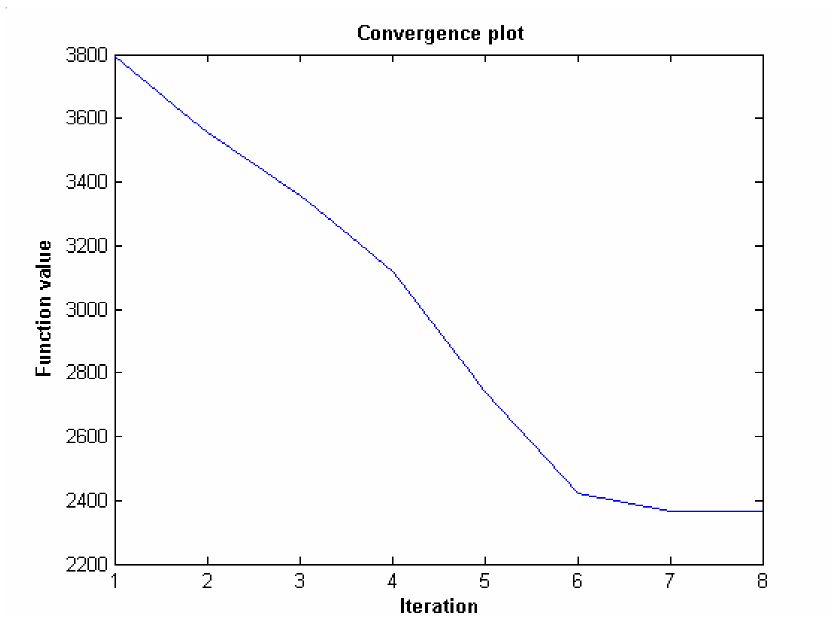


Figure 5.24 Convergence plot for the plate problem with four design variables

For the second case that involves two design variables namely the semi-major and semi-minor axes of the ellipse (*AX, BY*), the bounds were defined as *LB* = [1.5 1] inch and *UB* = [45 20] inch. The optimization process in this case converged to the minimum value of *TVOL* = 2364.5344 $inch^3$ in four iterations as shown in Figure 5.25.

The same problem was solved using the ANSYS First order optimization process and also using the ANSYS DesignXplorer goal driven optimization. Here only two design variables were considered for both the optimization processes. The convergence plot for the ANSYS First order optimization is shown in Figure 5.26. It took about 10 iterations for this process to converge to the objective function value of 2364.5344 inch^3.

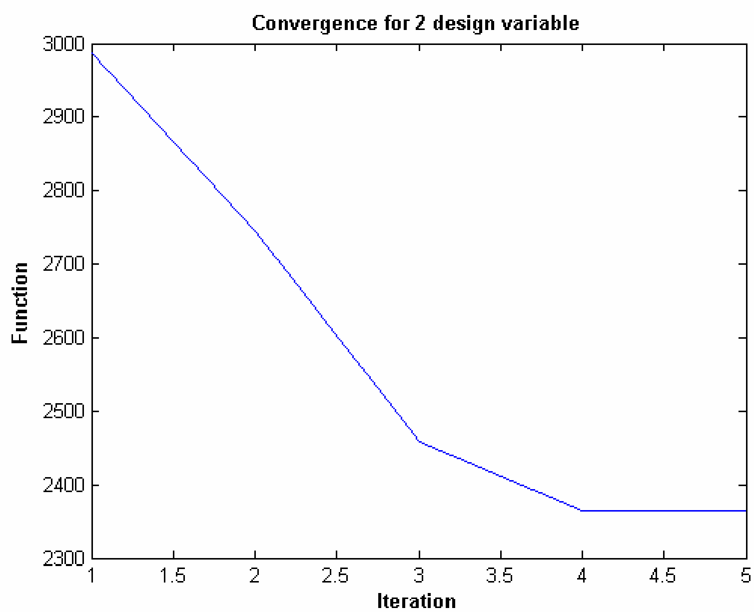


Figure 5.25 Convergence plot for the plate problem with two design variables

As can be observed from Figures 5.25 and 5.26, the results for both the MQR and the ANSYS First order optimization agree quite well with the true values. The finite element model for the plate with optimal design variables for the elliptical hole is shown in figure 5.27.

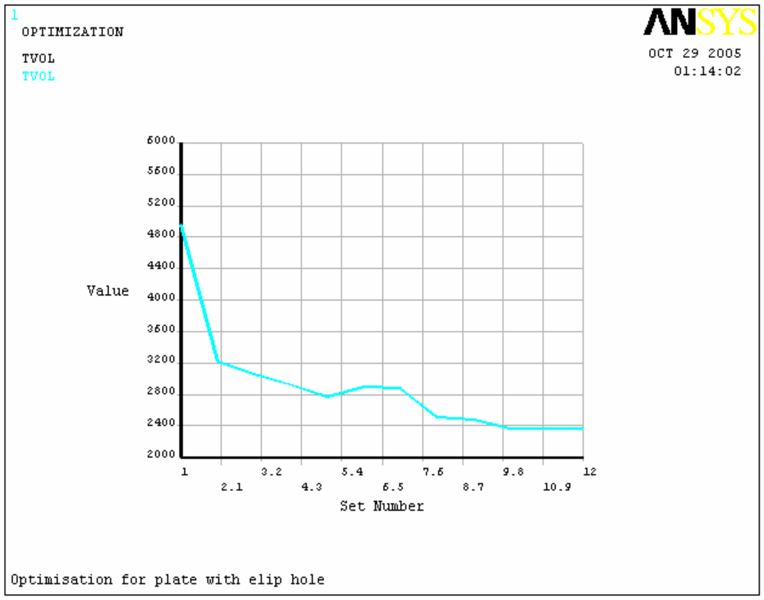


Figure 5.26 ANSYS Convergence plot for the plate problem-2 design variables

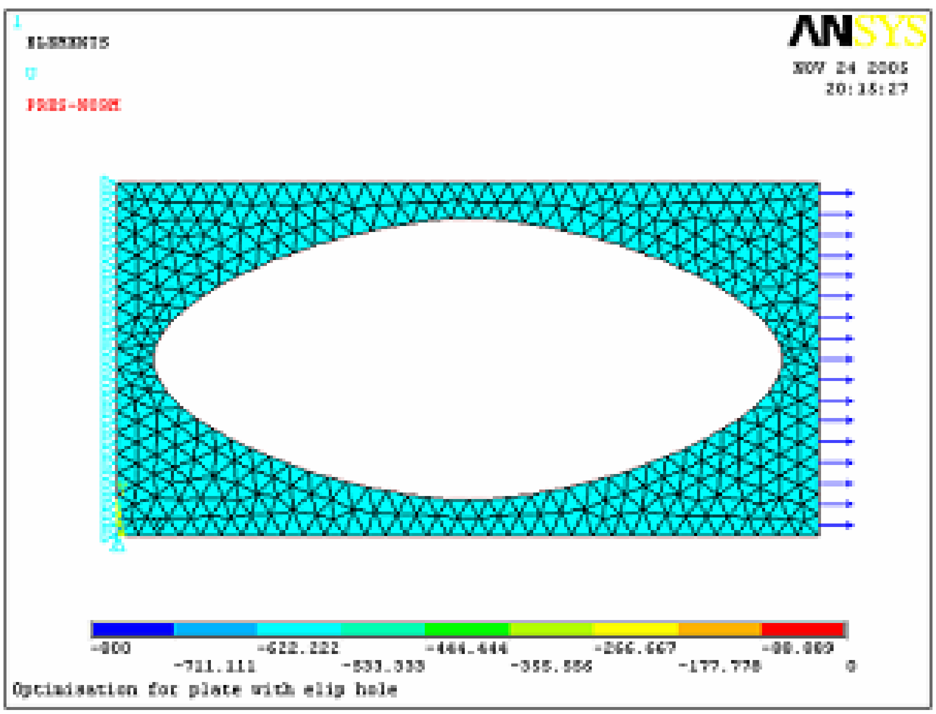


Figure 5.27 Finite element model of the plate with elliptical hole

The results obtained using the ANSYS DesignXplorer is shown in Figure 5.28. The results are found to agree quite well with those obtained from MQR and the First order optimization processes. The tradeoff plots generated as a result of the first sample generation and last sample generation of the ANSYS DesignXplorer goal driven optimization are shown in Figures 5.29 and 5.30.

Design Point 1

This design was rated BEST.

TABLE 7

Parameter values

Name	Value	Rating
AX	44.993	–
BY	19.995	–
VSMAX	10413	★
TVOL	2360.	★★★

Reference Design Point 1

This design was not rated.

TABLE 6

Parameter values

Name	Value	Rating
AX	44.993	–
BY	19.995	–
VSMAX	10054	★
TVOL	2365.6	★★★

Figure 5.28 Results from DesignXplorer Goal driven optimization

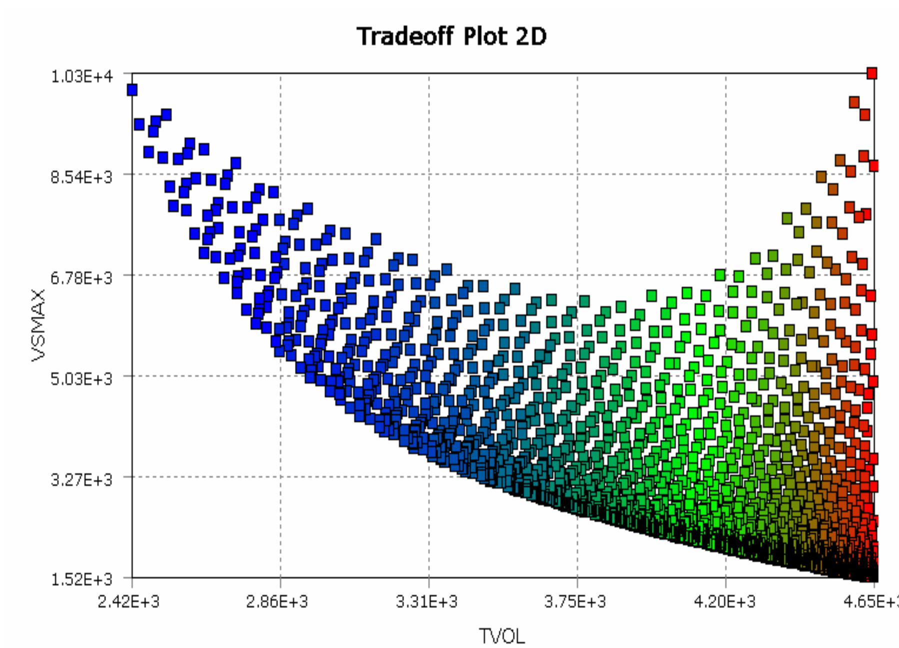


Figure 5.29 Tradeoff plot for first sample generation

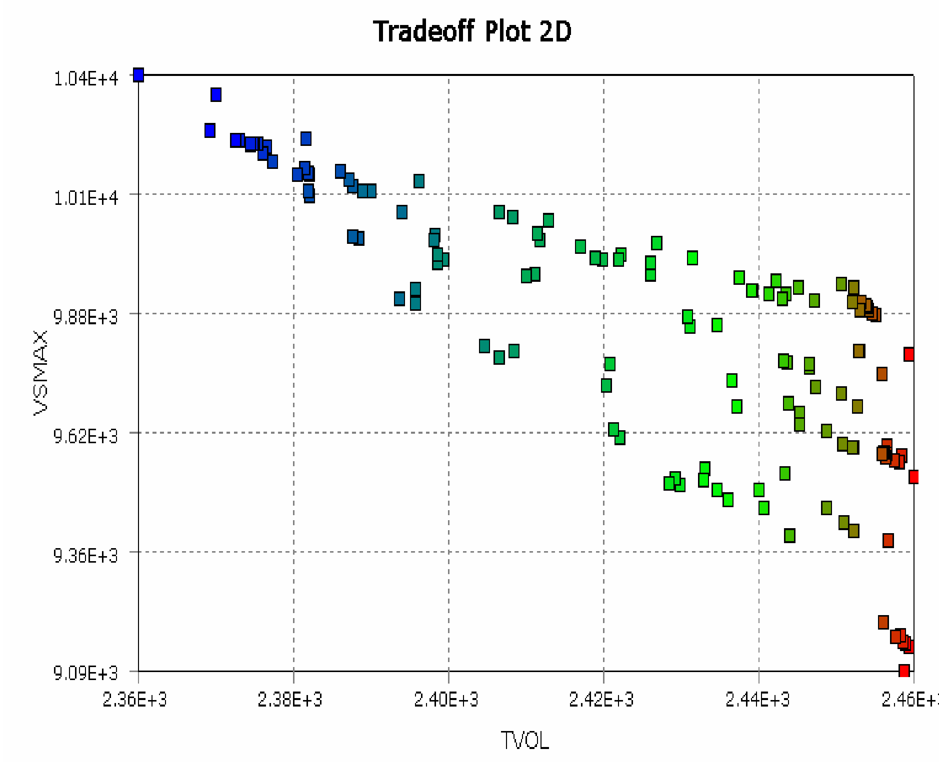


Figure 5.30 Tradeoff plot for last sample generation

The response plots generated during the response function evaluation in DesignXplorer study are shown in Figure 5.31 and 5.32. The first plot shows the variation of total volume of the plate with change in the input design variables whereas the second response plot indicates the variation of maximum stress as a function of the input design variables.

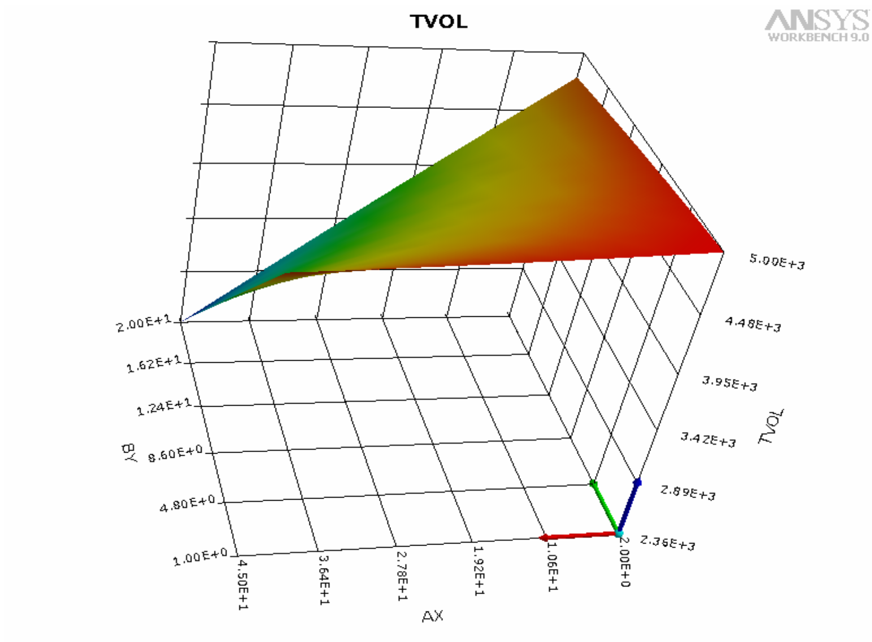


Figure 5.31 Response plot of Total volume 'TVOL'

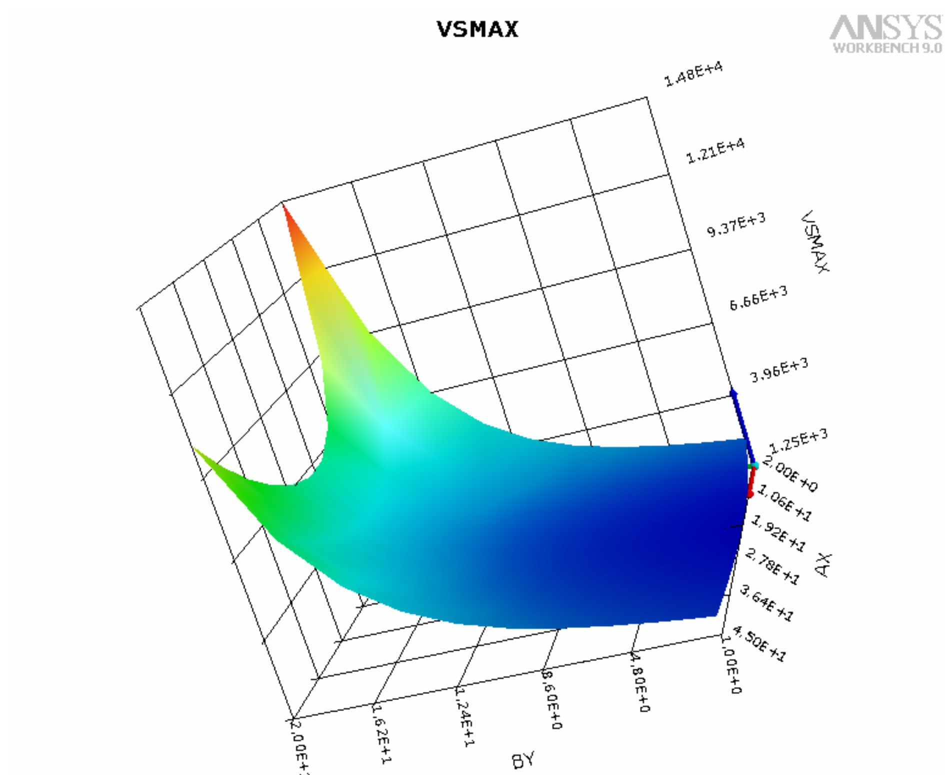


Figure 5.32 Response plot for maximum stress

5.5 Two Dimensional Vehicle Suspension

The objective of this problem is to design a 2D vehicle suspension as shown in Figure 5.33 for optimum ride comfort, when the vehicle goes over a speed bump. This study is based on a paper by *Deb and Saxena* [20] on finding the optimal suspension parameters using *genetic algorithm*. The characteristics that define a vehicle suspension are its spring constant 'k' of the coil spring and the damping constant 'c' of the damper. There are two suspensions one at the front end of the car and the other at the rear end. The tires of the car are modeled as springs of given stiffness value. There are three masses associated with this car model. The sprung mass which represents the mass of the car is indicated by 'm_s'. The un-sprung masses that represent the masses associated with the front and real axel is indicated by 'm_{fu}' and 'm_{ru}'. The objective is to find the optimal suspension parameters as the car runs over a speed bump, for minimum vertical displacement 'q_2' experienced by the passengers in the car such that the *jerk* or the rate of change of acceleration '$\dddot{q}_2$' is less than 18 m/s^3 [20]. The parameters 'k_{ft}' *and* 'k_{rt}' represent the front and rear tire stiffness as shown below. The parameters that define the front suspension are 'k_{fs}' and 'α_f' for the front spring and damper .Similarly for the rear suspension the parameters are 'k_{rs}' and 'α_r'. 'L' signifies the distance between the front and the rear axels and 'v' represents the velocity of the car. So as defined above the objective is to find the optimal suspension parameters k_{fs}, α_f, k_{rs} and α_r.

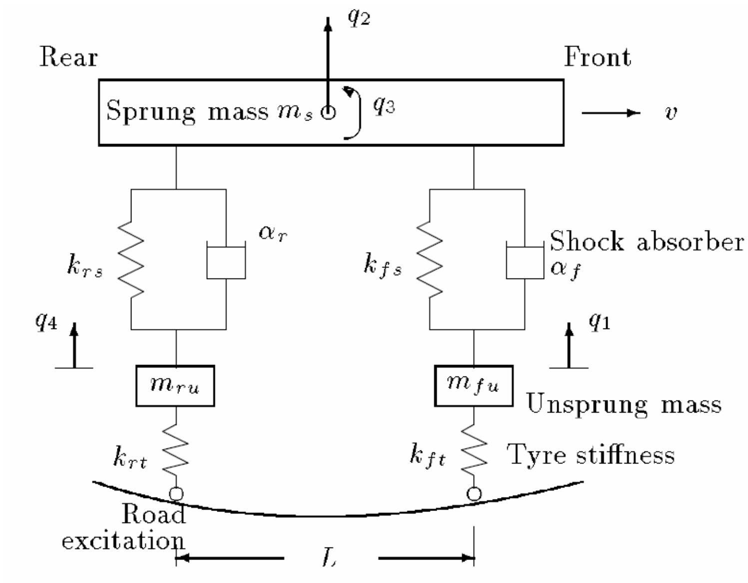


Figure 5.33 2D vehicle model [20]

In this experiment the speed bump is mathematically modeled as a half sinusoidal wave. The height and width of the speed bump is given as A = 70 mm and W = 500 mm. Thus the excitation caused by the speed bump on both the tires can be mathematically modeled by equations as shown below.

$$f_1(t) = A \ \sin\left[\frac{\pi t}{T}\right]$$

$$f_2(t) = A \ \sin\left[\frac{\pi(t - L/v)}{T}\right]$$

where $f_1(t)$ and $f_2(t)$ represent the excitations in the form of displacements on the front and rear tires of the car. 'T' represents the time taken by one tire to cross the bump. Graphically the excitations for the given values of *v*, *T*, *L* and *A* is shown in Figure 5.34.

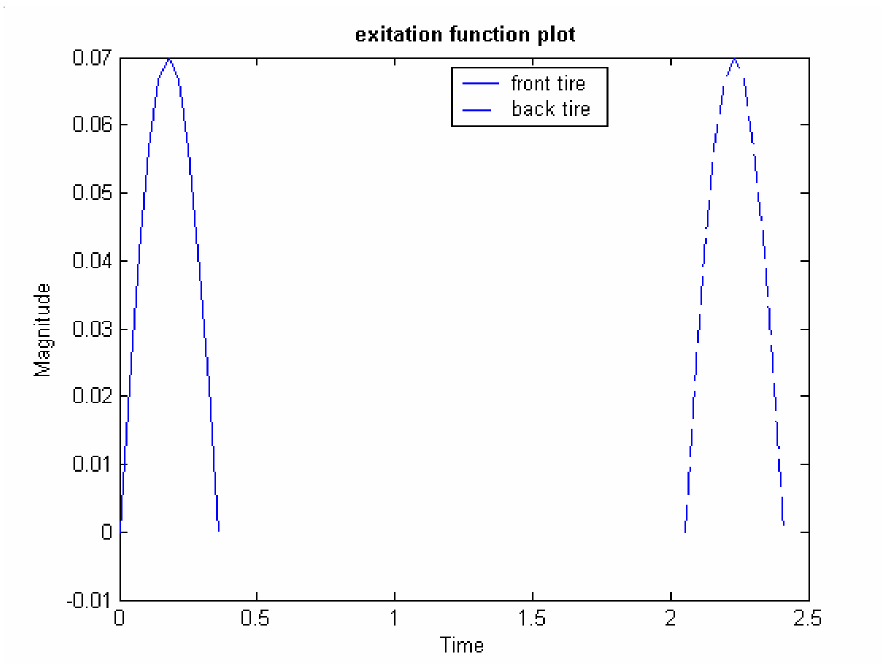


Figure 5.34 Excitation function plot

In the original work by *Deb* and *Saxena* [20], both the dampers and the rear suspension spring are considered to behave nonlinearly. The nonlinear characteristic of the rear suspension spring is shown in Figure 5.35. The dampers are also considered to behave in a similar fashion.

For the nonlinear spring the value of δ is given by $\delta_{k_{rs}} = 215\, mm$ and the relation between k_{rs}^{a}, k_{rs}^{b} and k_{rs}^{c} is given by $\frac{k_{rs}^{b}}{k_{rs}^{a}} = 1.28$ and $\frac{k_{rs}^{c}}{k_{rs}^{a}} = 1$.

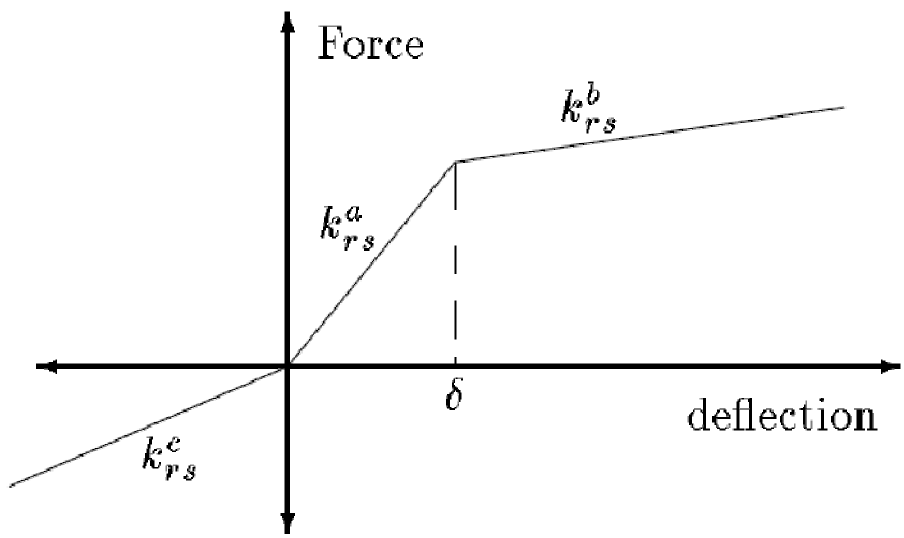


Figure 5.35 Nonlinear characteristic of the rear suspension spring

In case of the dampers the relationship between the damping constants are given as $\frac{\alpha_f^b}{\alpha_f^a} = 0.033$, $\frac{\alpha_r^b}{\alpha_r^a} = 0.257$,and $\frac{\alpha_f^c}{\alpha_f^a} = \frac{\alpha_r^c}{\alpha_r^a} = 1$ and values of δ in mm/sec are $\delta_{\alpha_f} = 50\, mm/\sec$ and $\delta_{\alpha_r} = 100\, mm/\sec$.

Let's now look into the aspect of vehicle modeling in ANSYS. Certain assumptions were made in ANSYS in the development of the vehicle model. The body or the chassis of the vehicle was modeled as a BEAM element with a very high Young's modulus of the order of 10^{11} psi and a Poisson's ratio of 0.3. The sprung and un-sprung masses were modeled using 3D MASS elements in ANSYS. For tires, rear damper and the front suspension spring-damper combination elements were used. However for the rear suspension spring a non-linear spring element was used. Thus in our approach non-linearity behavior was associated only with the rear suspension spring and all other elements were treated to behave linearly. A table summarizing the elements used to model 2D vehicle in ANSYS is shown below.

Table 5.1 Elements used in ANSYS for 2D vehicle suspension

Chassis of the car	BEAM 3 Element
Sprung Masses and Un-sprung masses	3D MASS 21 Element
Front suspension (Spring-Damper)	COMBI 14 Element
Rear suspension (Coil spring)	COMBI 39 Element
Rear suspension (Damper)	COMBI 14 Element
Front & Rear tire stiffness	COMBI 14 Element

Transient analysis was performed in ANSYS to find the vertical deflection of the sprung mass. A Central difference approximation for the numerical differentiation based on Taylor's series expansion was used to find the constraint function referred as *jerk*. The *jerk* which is defined as the rate of change of acceleration is given by the central difference formula.

$$y'''(t_0) \approx \frac{y_2 - 2y_1 + 2y_{-1} - y_{-2}}{2h^3}$$

where '*y*' represents the vertical deflection of the sprung mass as a function of time. An APDL file defining the problem parameters , the vehicle model, the solution process and the post-processing phase that also included the central difference scheme to find the constrain function was set up for the optimization process. The parameters already defined in the problem included the values for tire stiffness, the sprung and un-sprung masses ,the length 'l_1' between the front axel and mass center of the vehicle, the length 'L' between the front and rear axels, the polar moment of inertia '*J*' of the car and the velocity '*v*' of the car. The values of these parameters are given below as

m_s = 750 kg m_{ru}=115 kg m_{fu}= 50 kg

k_{rt} = 17 kg/mm k_{rt}= 15 kg/mm

l_1 = 1.50 m L = 2.85 m

J= 2.89 (10^4) kg.m^2

v = 5 kmph

Now lets look into the results obtained solving this problem using the MQR optimization, ANSYS First order optimization and ANSYS DesignXplorer goal driven optimization. The bounds of the design variables are given in the order

$\begin{bmatrix} k_f & \alpha_f & k_r^a & \alpha_r \end{bmatrix}$ as $LB = \begin{bmatrix} 40000 & 15000 & 25000 & 9000 \end{bmatrix}$ and $UB = \begin{bmatrix} 45000 & 20000 & 30000 & 12000 \end{bmatrix}$. The values of the parameters that define the MQR optimization process are '*h*' =1, '*r*'=0.001, *fEPS* =0.01 and *gEPS* =0.01. The total number of samples defined for the DOE is 'NC'=120. The number of initial samples that is equal to the number of samples that define the center points is given by 'N0'=20.

A modified MQR optimization code was used here as initially running the original code, it was found that the results of optimization largely depended on the starting point of the optimization algorithm in the design space. As a remedy to this problem, the optimization algorithm was iterated for a set of starting points whose value corresponded to the values of the samples of the main iteration.

Thus the modified MQR optimization process was found to converge in three iterations to a minimum value of objective function given by q_2 = 0.05371 meters with *jerk* =14.83 m / s^3. The convergence plot for the MQR process is shown in Figure 5.36. The optimal parameters of the front and rear suspension are $\begin{bmatrix} 44378 & 16148 & 28960 & 9734 \end{bmatrix}$.

The function convergence plot for the ANSYS First order optimization process is shown in Figure 5.37. It was found that the function converged to a minimum value of q_2= 0.05233 meter and *jerk* = 14.585 m / s^3. It can be observed that the results of MQR optimization process compare well with those obtained using ANSYS first order optimization process.

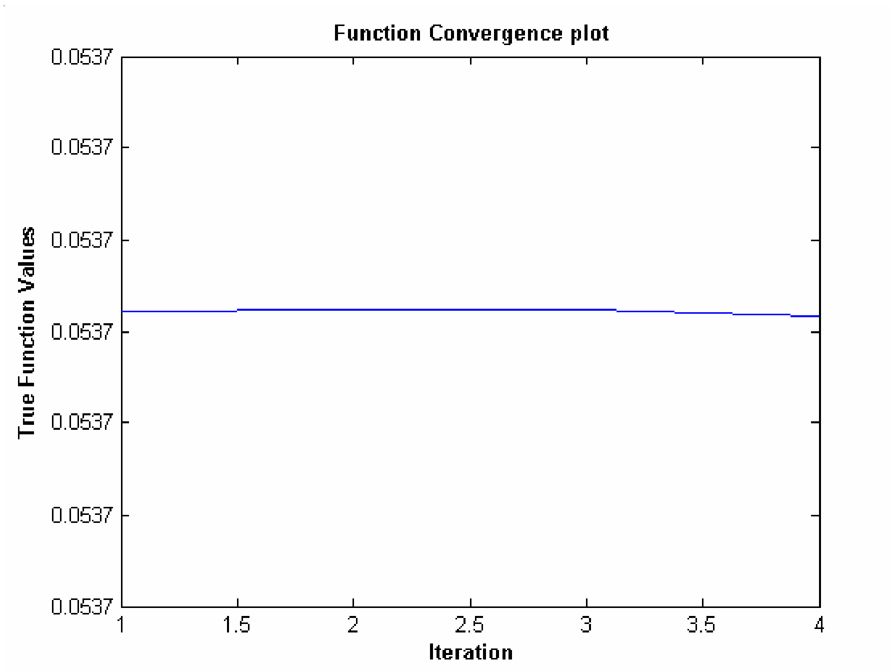


Figure 5.36 Function convergence plot

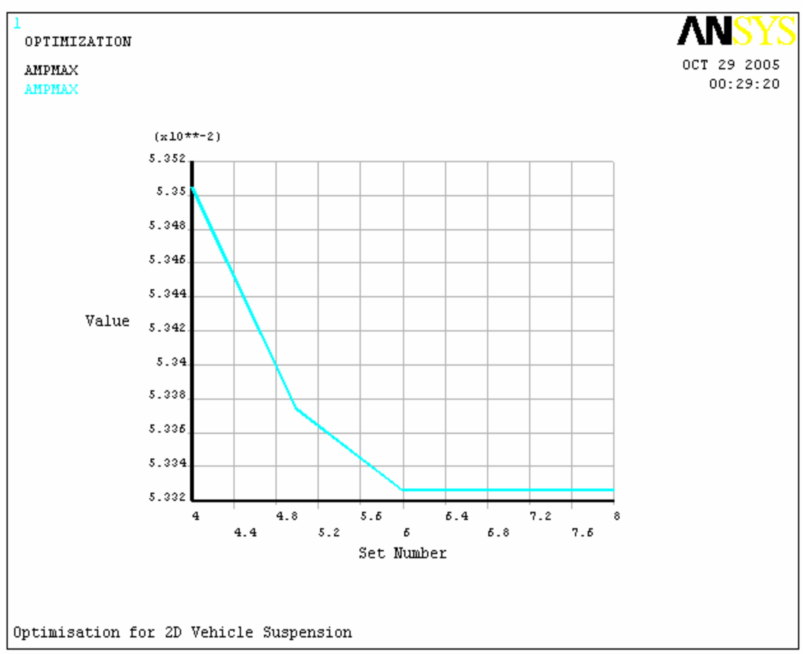


Figure 5.37 Function convergence - First order optimization

The results obtained using the ANSYS DesignXplorer is shown in Figure 5.38. The results are found to agree quite well with those obtained from MQR and the First order optimization processes.

Design Point 1

This design was rated BEST.

TABLE 7

Parameter values

Name	Value	Rating
Kf	44131	-
Af	15053	-
Kr	29559	-
Ar	10081	-
JERK	13.845	★★★
OBJF	5.2467e-002	★★★

Reference Design Point 1

This design was rated BEST.

TABLE 6

Parameter values

Name	Value	Rating
Kf	44131	-
Af	15053	-
Kr	29559	-
Ar	10081	-
JERK	13.84	★★★
OBJF	5.2466e-002	★★★

Figure 5.38 Results from ANSYS DesignXplorer goal driven optimization

The tradeoff plots generated as a result of the first sample generation and last sample generation of the ANSYS DesignXplorer goal driven optimization are shown in Figures 5.39 and 5.40.

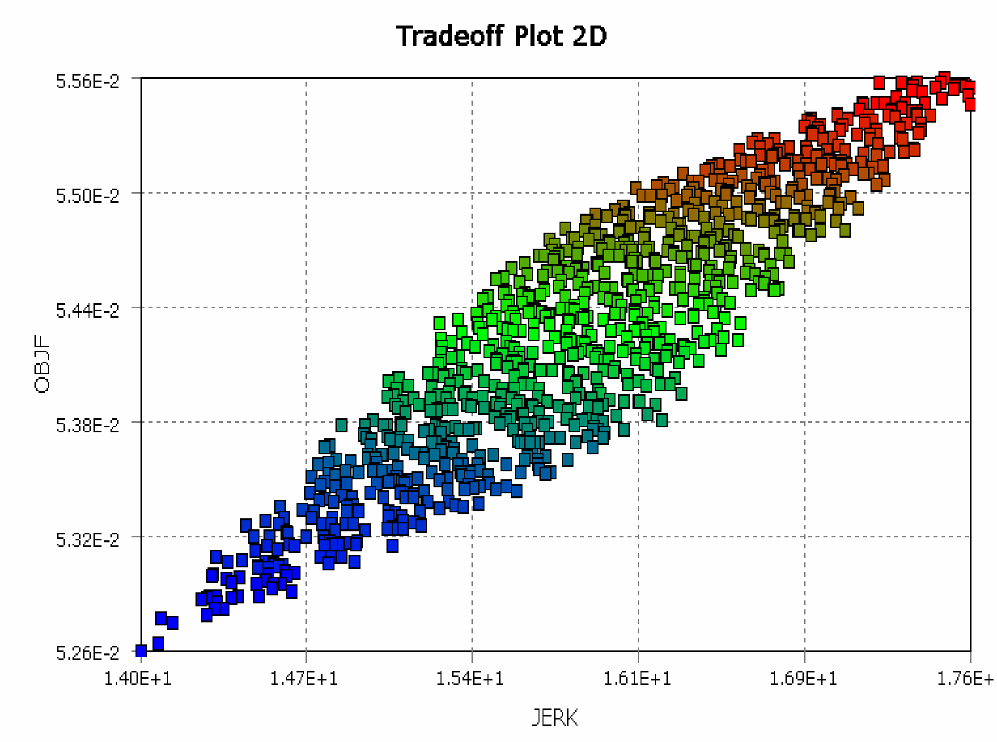


Figure 5.39 Tradeoff plate for the first generation of samples

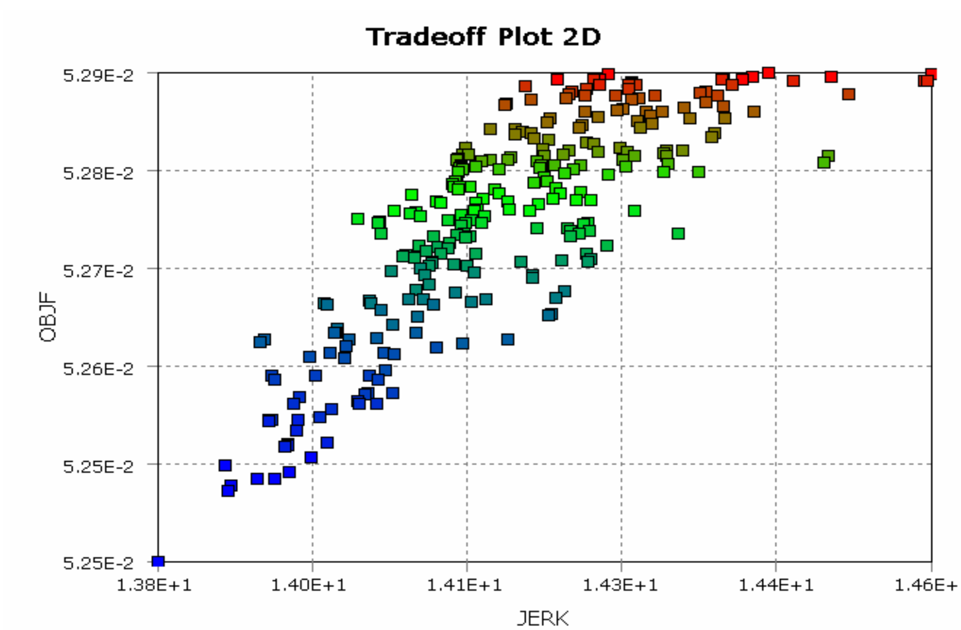


Figure 5.40 Tradeoff plot for the last generation of Pareto optimal points

The dynamic response plot of the displacement of sprung mass for the optimal design parameters as obtained from MQR optimization is shown in Figure 5.41. A comparison of this plot with the plot obtained from the study conducted by *Deb* [20] shows that there is a difference of about 10 mm in maximum deflection of the sprung mass. An explanation that could be described for this difference was the non-linear characteristics of the dampers that were considered in the original study by *Deb* [20]. On the other hand the dampers that involved in this study behaved linearly.

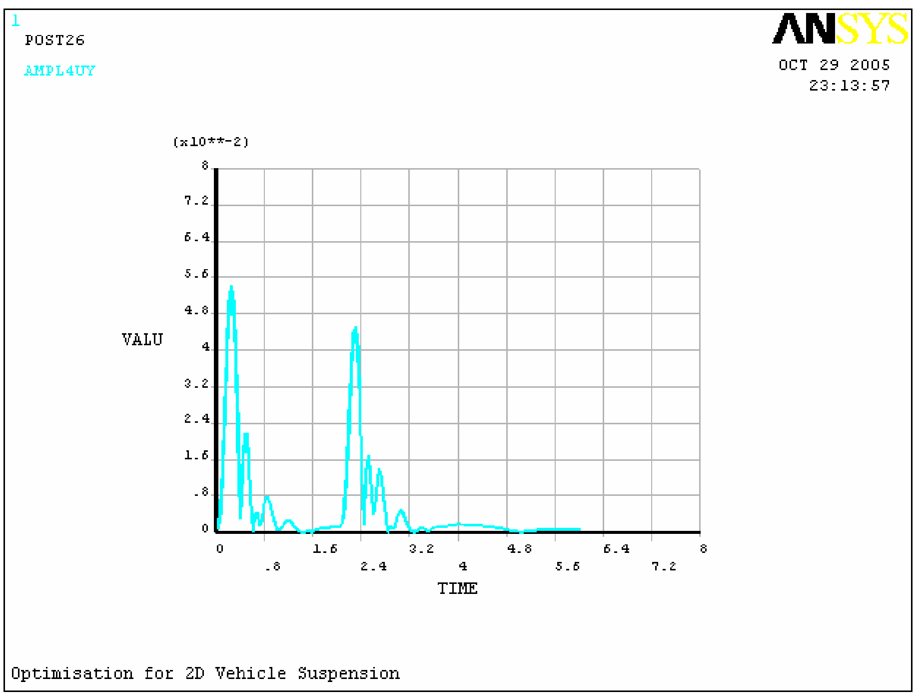


Figure 5.41 Dynamic response of the displacement of sprung mass

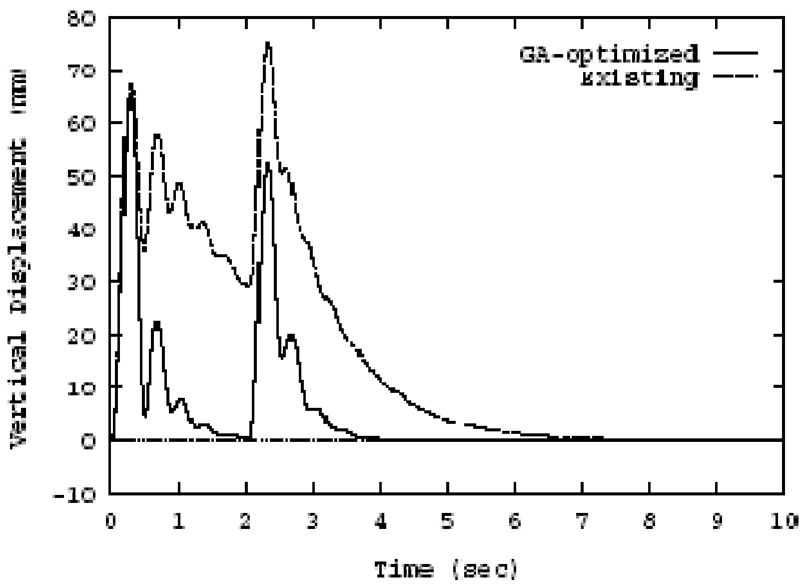


Figure 5.42 Dynamic responses as observed in reference [20]

5.5.1 Validation of ANSYS model

In this section we will verify whether our assumption of the ANSYS model is comparable with the mathematical model of the 2D vehicle suspension. For this test a linear model of the vehicle suspension in ANSYS is compared with the mathematical model that describes the dynamics of the suspension. A 2D vehicle suspension can be mathematically modeled by the following set of equations.

$$d_1 = q_1 - f_1(t)\ ,\ d_2 = q_2 + l_1 q_3 - q_1$$
$$d_3 = q_4 - f_2(t)\ ,\ d_4 = q_2 + l_2 q_3 - q_4$$

$$F_1 = k_{ft} d_1,\ \ F_2 = k_{fs} d_2,\ \ F_3 = \alpha_f \dot{d}_2,$$
$$F_4 = k_{rs} d_4,\ \ F_5 = \alpha_r \dot{d}_4,\ \ F_6 = k_{rt} d_3.$$

$$\left.\begin{aligned}
\ddot{q}_1 &= (F_2 + F_3 - F_1)/m_{fu} \\
\ddot{q}_2 &= -(F_2 + F_3 + F_4 + F_5)/m_s \\
\ddot{q}_3 &= [(F_4 + F_5)l_2 + (F_2 + F_3)l_1]/J \\
\ddot{q}_4 &= (F_4 + F_5 - F_6)/m_{ru}
\end{aligned}\right\} \tag{5.1}$$

The vertical deflection q_2 is found by converting this equation set (5.1) into state space form and then solving them using a fourth order Runge Kutta method in MATLAB. The deflection plot for the optimal suspension parameters for the suspension model is shown in Figure 5.43.

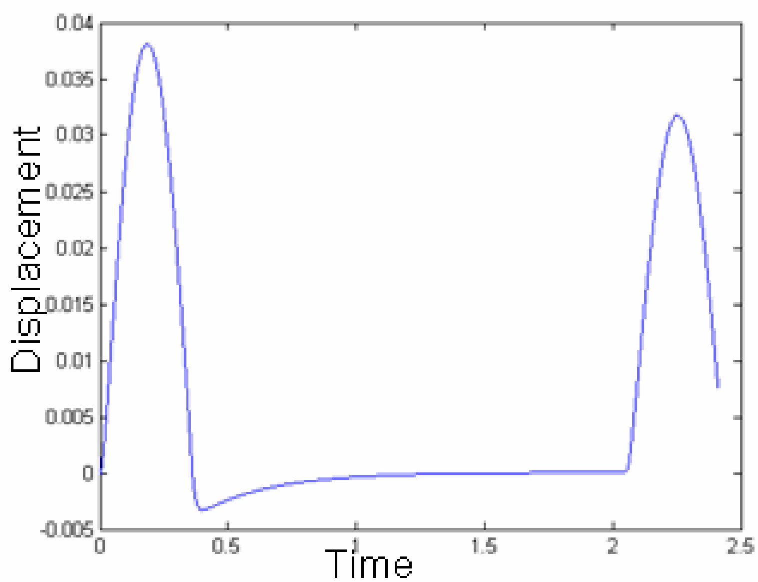


Figure 5.43 Deflection plot using fourth order Runge Kutta method in MATLAB

When comparing this response trend with Figure 5.41, it may be observed that the dynamics of the suspension are not well defined when problem is solved mathematically. The deflection plot as a result of fourth order Runge Kutta just reflects the movement of the tire over the speed bump but does not capture the vibrations that result after the tire crosses the bump. Hence to prove the model accuracy, the two dimensional vehicle model was reduced to a one dimensional model for the verification purpose.

For a one dimensional vehicle model only the front suspension parameters were considered. Half of the weight of the sprung mass was assumed to be concentrated over the front suspension. One dimensional vehicle model is as shown in Figure 5.44.

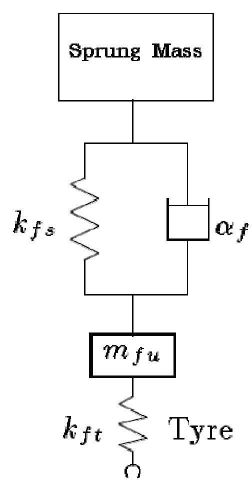


Figure 5.44 One dimensional vehicle model

Only the front tire excitation is considered while modeling the mathematical equations for the 1D vehicle model. Let y_1 be the displacement of the un-sprung mass and y_2 that of the sprung mass. The mathematical equations describing the dynamics of the 1D vehicle model can be explained as

$$
\begin{aligned}
&d_1 = y_1 - f_1(t) \quad \text{where } f_1(t) = A \sin\left[\frac{\pi t}{T}\right] \\
&d_2 = y_2 - y_1 \\
&F_1 = k_{ft} d_1, \quad F_2 = k_{fs} d_2, \quad F_3 = \alpha_f \dot{d}_2, \\
&\left.\begin{aligned}
\ddot{y}_1 &= (F_2 + F_3 - F_1)/m_{fu} \\
\ddot{y}_2 &= -(F_2 + F_3)/m_s
\end{aligned}\right\}
\end{aligned}
\qquad (5.2)
$$

The above set of equations is solved using the fourth order Runge Kutta technique by transforming them into first order differential equations using state space method. The deflection plot as obtained is shown in Figure 5.45. The deflection plot from ANSYS for the optimal values of front suspension parameters is shown in Figure 5.46. Comparing these two plots, we can conclude that our assumption of the vehicle model in ANSYS compares well with the mathematical model.

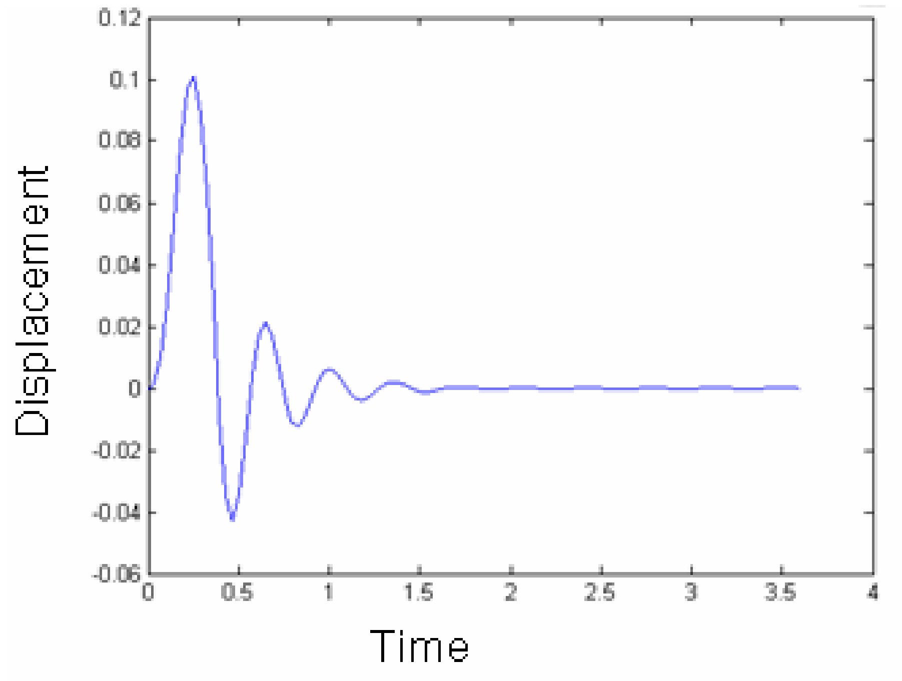


Figure 5.45 Deflection plot-mathematical model

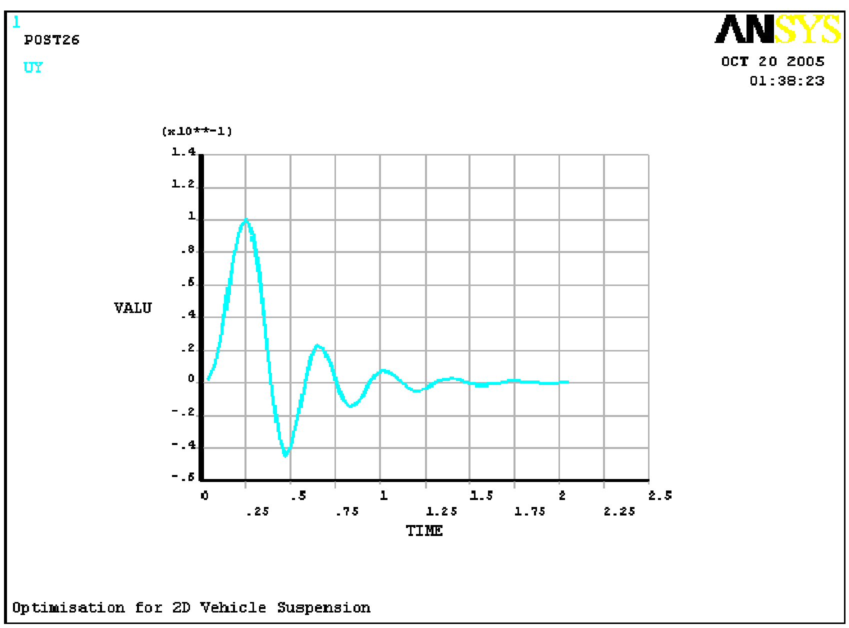


Figure 5.46 ANSYS- deflection plot for 1D vehicle model

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

The main objective of this work was to define a design automation process in MATLAB to solve computationally expensive design optimization problems using a non conventional optimization process that integrates a *DOE and response surface* modeling tool (MQR) and ANSYS a powerful finite element solver. The design automation process in MATLAB was successfully implemented by integrating the MQR based optimization process and ANSYS the finite element solver. Several well documented optimization problems were considered and the results from the proposed method was compared with the conventional optimization process in ANSYS based on first order optimization and also with a non-conventional process called ANSYS DesignXplorer goal driven optimization.

6.1.1 Conclusions

It was observed that the results from the MQR optimization process agree quite well with the first order optimization and the DesignXplorer goal driven optimization available in ANSYS. However the MQR optimization process performs really well when the number of design variables 'n' is between 5 and 10. The accuracy of the results fell drastically when the number of design variables exceeded ten. On comparison with the ANSYS DesignXplorer goal driven optimization, it was found that the accuracy of the MQR hardly depended on the sample size. On the other hand it was observed that the size of sample was a big factor for the accuracy of results obtained using DesignXplorer goal driven optimization. Further it was noticed that the number of function evaluations and computation time was comparatively less for the MQR optimization process. Hence MQR optimization process can be thought of as a good alternative to DOE based DesignXplorer goal driven optimization when the maximum the number of design variables are less than ten.

6.1.2 Recommendations

1. The problems considered in this study were size and dynamic response based optimization problems. The proposed method can be tried to solve complex optimization problems such as shape and topology based.
2. The algorithm can be modified for parallel implementation and can be experimented to solve large non-linear design problems efficiently.
3. A good alternative to solve certain complex optimization problems such as vehicle suspension problems would be to use *genetic algorithm* in place of sequential *quadratic programming* for the optimization process. Research have shown that *genetic algorithms* in most cases has found to converge to the global minimum in the function space as compared to local minimum as was observed in the 2D vehicle model.
4. Adaptive sampling and response modeling methods that samples each variable according to the contribution to the response should be investigated.

5. Other metamodeling techniques that can handle large number of design variables should be explored.

APPENDIX A

ANSYS APDL SAMPLE FILES

In this appendix, a sample ANSYS APDL file used during the MQR based optimization is shown. The parameter POPSIZE is given as a command line option to ANSYS. The finite element solution is initialized by MATLAB using the command [9]

"<drive>:\Program Files\Ansys Inc\V90\ANSYS\bin\<platform>\ansys90" -b -p ansysrf -POPSIZE #sets of design variables -i inputname -o outputname

by means of a system call.

A.1 9-Bar Truss

```
!************************************************!
!   ANsys Input for Test Problem- 9 Bar truss      !
!************************************************!

!Nine Element Groups

/CWD,C:\MATLAB6p5\work  !Change Ansys Directory To
                    !Matlab Work directory
!******DELETE PREVIOS ENTRIES OF FILES CREATED*****!

/FILNAM,TP-9
/PREP7

NUMDESV=2               !Number of design variables
NUMELEM=9               !Number Of element groups

!************************************************!
! Read The Area matrix generated in Matlab        !
!************************************************!

*DIM,XAREAS,,POPSIZE,NUMDESV
*VREAD,XAREAS(1,1),dvar,var,,JIK,NUMDESV,POPSIZE
(F18.13)
```

```
!************************************************!
!  Creating the model of 9-bar truss structure
!  Material is Steel,AISI 1020,low Carbon
!  ANSYS Element type LINK8 (3 D truss)is used
!************************************************!

ET,1,LINK1

*DO,I,1,NUMDESV       !Defining the real constant
  R,I,XAREAS(1,I)      !set for the design variables
*ENDDO

MP,EX,1, 2.972501e+07  !Elastic Moduluss in lbf/in^2
MP,PRXY,1,0.29         !Poissons Ratio

!****************DEFINING NODES******************!

N,1, 0, 0, 0
N,2,10, 0, 0
N,3,20, 0, 0
N,4,30, 0, 0
N,5,10,10, 0
N,6,20,10, 0

!**********DEFINING ELEMENTS BTW NODES*************!

*DO,I,1,3,1      ! Elements 1, 2, 3& 6 belong to real constant set 1
real,1
en,I,I,I+1
*ENDDO

real,1
en,6,5,6
!**************************************************!
real,2              ! Elements 4,5,7,8 & 9 belong to real constant set 2
en, 4, 1, 5

real,2
en,5,2,5

real,2
en,7,2,6
```

```
real,2
en,8,3,6

real,2
en,9,4,6

!********DEFINING BOUNDRY CONDITIONS & FORCES********!

D,1,ALL,0             !Node 1 is fixed completly

D,4, UY,0             !Node 2 is fixed in y direction
                      !free to move along x direction


F,5,fy,-5000          !defines forces acting on nodes
F,6,fx, 7000          !5&6
F,6,fy, 2000

FINISH

!************DEFINING ARRAYS FOR OUTPUT****************!
*GET,NCOUNT,NODE,,COUNT
*DIM,NDISPX ,ARRAY,NCOUNT,1
*DIM,NDISPY ,ARRAY,NCOUNT,1
*DIM,NDISPZ ,ARRAY,NCOUNT,1

*GET,ECOUNT,ELEM,,COUNT
*DIM,STRESS,array,ECOUNT,1

*DIM,DVAR,ARRAY,NUMDESV,1

*cfopen,results,txt

!****************************************************!
!          RUN LOOP FOR ALL CASES
!****************************************************!
*DO,I,1,POPSIZE,1

/PREP7

*DO,J,1,NUMDESV
 R,J,XAREAS(I,J)
*ENDDO
```

```
*VGET,DVAR(1,1),RCON,1,CONSTANT,1
*VWRITE,DVAR(1,1)
(E22.14)

FINISH

!***************SOLUTION PHASE*************************!
/SOLU
ANTYPE,STATIC
SOLVE
FINISH

!*************POST PROCESSING PHASE********************!
!Writing files results.txt,estress.var,noddisp.var &    !
!Tvolume.var to the work directory in Matlab           !              !
!****************************************************!
/POST1

SET,LAST
ETABLE,Stress,LS,1

*VGET,NDISPX(1,1),node,1,U,X
*VGET,NDISPY(1,1),node,1,U,Y
*VGET,NDISPZ(1,1),node,1,U,Z

*VWRITE,NDISPX(1,1)
(E22.14)

*VWRITE,NDISPY(1,1)
(E22.14)

*VWRITE,NDISPZ(1,1)
(E22.14)
*VGET,STRESS(1),ELEM,1,ETAB,Stress

*VWRITE,STRESS(1)
(E22.14)

ETABLE,Volume,VOLU
SSUM

*get,TVOL,SSUM,,ITEM,Volume
```

```
*VWRITE,TVOL
(E22.14,2X)

*ENDDO        ! END of DO LOOP for POPULATION CASES

*cfclos

*DIM,DATA,,30,POPSIZE
*VREAD,DATA(1,1),results,txt,,IJK,30,POPSIZE
(E22.14)

*MWRITE,DATA(1,1),pdata,var,,IJK,30,POPSIZE
(E22.14,2X)

FINISH

/DELETE,TP-9,EMAT
/DELETE,TP-9,ESAV
/DELETE,TP-9,FULL
/DELETE,TP-9,RST
/DELETE,TP-9,MNTR
/DELETE,TP-9,PVTS
/DELETE,TP-9,BCS
/DELETE,dvar,VAR

/EXIT,NOSAVE
```

A.2 25-Bar Truss

```
!************************************************!
!   ANsys Input for Test Problem- 25 Bar truss     !
!************************************************!

!7 Element Groups

/CWD, C:\MATLAB6p5\work  !Change Ansys Directory To
                         !Matlab Work directory

!******DELETE PREVIOS ENTRIES OF FILES CREATED*****!


/FILNAM,TP-25
/PREP7
```

```
/DELETE,NEdata,VAR
/DELETE,pdata,VAR


NUMDESV=7                          !Number of design variables
NUMOBJ =1                          !Number of objective functions
NUMELEM=25                         !Number Of element
MASSDENSITY= 2.620360e-04  !Units:lbf-sec^2/in^3


!************************************************!
! Read the Area matrix generated in Matlab
!************************************************!

*DIM,XAREAS,,POPSIZE,NUMDESV
*VREAD,XAREAS(1,1),dvar,var,,JIK,NUMDESV,POPSIZE
(E22.14)

!************************************************!
!  Creating the model of 25-bar truss structure
!  Material is Alluminum 2014-T6.(ETBX.com)
!  ANSYS Element type LINK8 (3 D truss)is used
!************************************************!

ET,1,LINK8

*DO,I,1,NUMDESV        !Defining the real constant
  R,I,XAREAS(1,I)        !set for the design variables
*ENDDO

MP,EX,1,  1.049800e+07 !Elastic Moduluss in lbf/in^2
MP,PRXY,1,0.33           !Poissons Ratio

!****************DEFINING NODES*****************!

N, 1, -37.5,   0.0,200
N, 2,  37.5,   0.0,200
N, 3, -37.5, -37.5,100
N, 4,  37.5, -37.5,100
N, 5,  37.5,  37.5,100
N, 6, -37.5,  37.5,100
N, 7,-100.0,-100.0,  0
N, 8, 100.0,-100.0,  0
```

```
N, 9, 100.0, 100.0,  0
N,10,-100.0, 100.0,  0

!**********DEFINING ELEMENTS BTW NODES**************!
! Need To Change depending on the number of design
!variables in the problems
!**************************************************!

real ,1
en,1,1,2
real,2
en,3,1,4
en,4,1,5
en,6,2,3
en,9,2,6

real,3
en,2,1,3
en,5,1,6
en,7,2,4
en,8,2,5

real,4
en,10,3,4
en,11,4,5
en,12,5,6
en,13,6,3

real,5
en,15,7,6
en,25,10,3
en,19,8,5
en,20,9,1

real,6
en,16,7,4
en,17,8,3
en,21,9,6
en,23,10,5

real,7
en,14,7,3
en,18,8,4
```

```
en,22,9,5
en,24,10,6

!********DEFINING BOUNDRY CONDITIONS & FORCES********!
D, 7, UX,, , 10, 3            !Node 7 through 10 is fixed on
D, 7, UY,, , 10, 3            !all 3 directions.
D, 7, UZ, , ,10, 3



F,1,fy, 20000        !defines forces acting on nodes
F,1,fz, -5000        !1&2
F,2,fy,-20000
F,2,fz, -5000

FINISH

!************DEFINING ARRAYS FOR OUTPUT*****************!

*GET,NCOUNT,NODE,,COUNT
*DIM,NDISPX ,ARRAY,NCOUNT,1
*DIM,NDISPY ,ARRAY,NCOUNT,1
*DIM,NDISPZ ,ARRAY,NCOUNT,1

*GET,ECOUNT,ELEM,,COUNT
*DIM,STRESS,array,ECOUNT,1

*DIM,DVAR,ARRAY,NUMDESV,1

!**************************************************!
!Write Element Count and Node Count data to NEdata.var
!**************************************************!

*cfopen,NEdata,var
*GET,NCOUNT,NODE,,COUNT
*VWRITE,NCOUNT
(E22.14)
*GET,ECOUNT,ELEM,,COUNT
*VWRITE,ECOUNT
(E22.14)
*cfclos
```

```
!*****************************************************!
!Open results txt file to write solutions data to matlab
!*****************************************************!

*cfopen,results,txt

!*****************************************************!
!           RUN LOOP FOR ALL CASES
!*****************************************************!

*DO,I,1,POPSIZE,1

/PREP7

*DO,J,1,NUMDESV
 R,J,XAREAS(I,J)
*ENDDO


*VGET,DVAR(1,1),RCON,1,CONSTANT,1
*VWRITE,DVAR(1,1)
(E22.14)

FINISH

!**************SOLUTION PHASE************************!

/SOLU
ANTYPE,STATIC
SOLVE
FINISH

!**************POST PROCESSING***********************!
/POST1



SET,LAST
ETABLE,Stress,LS,1

*VGET,NDISPX(1,1),node,1,U,X
*VGET,NDISPY(1,1),node,1,U,Y
```

```
*VGET,NDISPZ(1,1),node,1,U,Z


*VWRITE,NDISPX(1,1)
(E22.14)

*VWRITE,NDISPY(1,1)
(E22.14)

*VWRITE,NDISPZ(1,1)
(E22.14)


*VGET,STRESS(1),ELEM,1,ETAB,Stress


*VWRITE,STRESS(1)
(E22.14)

ETABLE,Volume,VOLU
SSUM

*get,TVOL,SSUM,,ITEM,Volume


*VWRITE,TVOL
(E22.14,2X)

*ENDDO           ! END of DO LOOP for POPULATION CASES

*cfclos

!*****************************************************!
!Write solutions for all population set to pdata.var file!
!*****************************************************!
rows= NUMDESV + NCOUNT*3 + ECOUNT + NUMOBJ
*DIM,DATA,ARRAY,rows,POPSIZE
*VREAD,DATA(1,1),results,txt,,IJK,rows,POPSIZE
(E22.14)
*MWRITE,DATA(1,1),pdata,var,,JIK,POPSIZE,rows
(100(E22.14,2X))
```

```
FINISH

/DELETE,results,TXT
/DELETE,TP-25,EMAT
/DELETE,TP-25,ESAV
/DELETE,TP-25,FULL
/DELETE,TP-25,RST
/DELETE,TP-25,MNTR
/DELETE,TP-25,PVTS
/DELETE,TP-25,BCS
/DELETE,dvar,VAR

/EXIT,NOSAVE
```

A.3 Plate with an elliptical hole

```
!*************************************************!
!   ANSYS Input for Test Problem- Plate with hole  !
!*************************************************!

/CWD,C:\MATLAB6p5\work  !Change Ansys Directory To
                        !Matlab Work directory

!******DELETE PREVIOS ENTRIES OF FILES CREATED*****!


/FILNAM,TPlate
/PREP7


NUMDESV=2                          !Number of design variables
NUMOBJ =1                          !Number of objective functions
MASSDENSITY= 2.620360e-04 !Units:lbf-sec^2/in^3

!******Geometric dimensions****************************!

W  =100              !Width of the rectangular plate
H  =50               !Height of the restangular plate
T  =1                !Thickness of the rectangular plate
P1 =0
P2 =-800
```

```
!**************************************************!
!  Material is Alluminum 2014-T6.(ETBX.com)
!  ANSYS Element type LINK8 (3 D truss)is used
!**************************************************!

ET, 1, PLANE2
KEYOPT, 1, 3, 3              ! PLANE STRESS ELEMENT WITH THICKNESS

! Define Real constants
R, 1, T

MP, EX, 1, 1.049800e+07   ! Elastic Moduluss in lbf/in^2
MP, PRXY, 1, 0.33            !Poissons Ratio


!*******************************************************!
! Read the Design Variable matrix generated in Matlab
! Vector containing the input design variables
!*******************************************************!

*DIM, XDVAR,, POPSIZE, NUMDESV
*VREAD, XDVAR (1, 1),dvar,var,,JIK,NUMDESV,POPSIZE
(E22.14)


XC=W/2
YC=H/2

AX =XDVAR (1, 1)
BY =XDVAR (1, 2)

!***********Create geometry****************************!
BLC4, 0, 0, W, H

K, 5,    XC, YC           ! Defines the centre of the Ellipse
K, 6, XC+AX, YC           ! AX is the semi major axis and BY is
K, 7,    XC, YC+BY        ! Semi minor axis.
K, 8, XC-AX, YC
K, 9,    XC, YC-BY

BSPLINE, 7, 8, 9,,,, BY,0,0, BY,0,0
BSPLINE, 7, 6, 9,,,,-BY,0,0,-BY,0,0
```

```
LGLUE, 5, 6

AL, 5, 6

ASBA, 1, 2

! APPLYING CONSTRAINTS
DL, 4,  ,UX,0
DK, 1,UY, 0
SFL, 2, PRES, P2
SFL,3,PRES,P1

! AREA MESHING
SMRTSIZE, 3
AMESH, ALL
Finish

!*************************************************************!
! Open results txt file to write solutions data to Matlab
!*************************************************************!

*cfopen, results, txt

!*************************************************************!
!            RUN LOOP FOR ALL CASES                        !
!*************************************************************!

*DO, I, 1, POPSIZE, 1

/PREP7

!**For the prob where constraints are just the semi major & minor axis of ellipse**!

XC=W/2
YC=H/2

AX =XDVAR (I, 1)
BY =XDVAR (I, 2)

*VWRITE, AX
(E22.14)
*VWRITE, BY
(E22.14)
```

```
ACLEAR, ALL
ADELE, ALL
LDELE, 5, 6
KDELE, 5, 9

!*************Define the size of elliptical hole****************!

K, 5,    XC, YC           ! Defines the centre of the Ellipse
K, 6, XC+AX, YC           ! AX is the semi major axis and BY is
K, 7,    XC, YC+BY        ! semiminor axis.
K, 8, XC-AX, YC
K, 9,    XC, YC-BY

BSPLINE, 7, 8, 9,,,, BY,0,0, BY,0,0
BSPLINE, 7, 6, 9,,,,-BY,0,0,-BY,0,0

LGLUE, 5, 6

AL, 1, 2, 3, 4
AL, 5, 6

ASBA, 1, 2

! AREA MESHING
AMESH, ALL
AREFINE, ALL,2
FINISH

*GET, ECOUNT, ELEM,,COUNT

!***************SOLUTION PHASE**********************************!

/SOLU
ANTYPE, STATIC
SOLVE
FINISH

!*********************POST PROCESSING*************************!
/POST1
SET, LAST

ETABLE, SEQV, S, EQV             !Create a table of element stress values.
ESORT,ETAB ,SEQV ,0 ,1 ,
```

```
*GET,VSMAX,SORT,,MAX

*VWRITE,VSMAX
(E22.14)

ETABLE, Volume, VOLU
SSUM

*get, TVOL, SSUM, ,ITEM ,Volume


*VWRITE, TVOL
(E22.14, 2X)

*ENDDO              ! END of DO LOOP for POPULATION CASES

*cfclos

!*******************************************************!
!Write solutions for all population set to pdata.var file!
!*******************************************************!
rows = NUMDESV + 1 + NUMOBJ
*DIM, DATA, ARRAY, rows ,POPSIZE
*VREAD, DATA (1, 1),results ,txt ,, IJK ,rows ,POPSIZE
(E22.14)
*MWRITE, DATA (1, 1),pdata ,var ,,JIK ,POPSIZE ,rows
(100(E22.14, 2X))


FINISH

/DELETE,results,TXT
/DELETE,TPlate,EMAT
/DELETE,TPlate,ESAV
/DELETE,TPlate,FULL
/DELETE,TPlate,RST
/DELETE,TPlate,MNTR
/DELETE,TPlate,PVTS
/DELETE,TPlate,BCS

/EXIT, NOSAVE
```

A.4 2 Dimensional Vehicle Suspension

```
!***********************************************!
!   ANsys Input for 2D Vehicle Suspension Problem  !
!***********************************************!

!9 Elements in Total

/CWD, C:\MATLAB6p5\work         ! Change ANSYS directory to
                                ! MATLAB Work directory

!******DELETE PREVIOS ENTRIES OF FILES CREATED*****!

/FILNAM, 2DVS
/PREP7

NUMDESV=4          ! Number of design variables
NUMOBJ =1          ! Number of objective functions

!************************************************!
! Define Elements that model the 2D vehicle Suspension           !
! The Chasse of the car is modeled using BEAM3 elements         !
! MASS21 elements are used to signify rigid mases               !
! Combination element (Spring14) is used to model shock         !
!-absorber and tire stiffness                                    !
!************************************************!

ET, 1, BEAM3
ET, 2, MASS21, , , 3
ET, 3, COMBIN14, , , 2
ET, 4, COMBIN14, , , 2
ET, 5, MASS21, , , 4
ET, 6, MASS21, , , 4
ET, 7, COMBIN14, , , 2
ET, 8, COMBIN14, , , 2
ET, 9, COMBIN39, , , 2

!************************************************!
!    Define the real constant set for elements      !
!************************************************!
R, 1, 1, 1, 1
R, 2, 730, 2.89e4     ! Mass of sprung mass=730 kg &
                      ! Torsional Inertial Izz=2.89e4
```

```
!*************************************************!
!  Read The matrix generated in Matlab that contains !
!  the input design variables for spring & damper    !
!*************************************************!

*DIM, XDVAR, , POPSIZE, NUMDESV

*VREAD, XDVAR (1, 1), dvar, var, ,JIK, NUMDESV, POPSIZE
(E22.14)

Kf =XDVAR (1, 1)       ! Kf & Af=spring & damping constant
Af =XDVAR (1, 2)       !of front suspension.
Kar =XDVAR (1, 3)      ! Kr & Ar=spring & damping constant
Ar =XDVAR (1, 4)       !of rear suspension.

Kbr=0.215*Kar

R,3,Kf,Af
R,4,,Ar

R,5, 50                          !Front unsprung mass = 50kg
R,6,115                          !rear unsprung mass  =115Kg

R, 7, 147000                     ! Front tire stiffness=15000 kg/m
R, 8,166600                      !rear tire stiffness =17000 kg/m
R,9,-1,-Kar,-0.5,-0.5*Kar,0.0,0.0        !Non Linear Spring Data
RMORE,0.215,Kbr,0.5,0.5*1.28*Kar,1,1.28*Kar

!**********************************************!
! Properties of Beam : Random value is assumed            !
! EX=10e11 & Poissons ratio PRXY =0.3                     !
!**********************************************!
MP, EX, 1, 10e11       ! Elastic Moduluss in lbf/in^2
MP, PRXY, 1, 0.3       ! Poissons Ratio

!****************DEFINING NODES*****************!
N, 1, 0.00, 0, 0
N, 2, 0.00, 1, 0
N, 3, 0.00, 2, 0
N, 4, 1.35, 2, 0
N, 5, 2.85, 2, 0
N, 6, 2.85, 1, 0
N, 7, 2.85, 0, 0
```

```
!**********DEFINING ELEMENTS BTW NODES**************!
TYPE, 8
REAL, 8
E, 1, 2

TYPE, 6
REAL, 6
E, 2

TYPE, 4
REAL, 4
E, 2, 3

TYPE, 9
REAL, 9
E, 2, 3

TYPE, 1
REAL, 1
E, 3, 4

TYPE, 2
REAL, 2
E, 4

TYPE, 1
REAL, 1
E, 4, 5

TYPE, 3
REAL, 3
E, 5, 6

TYPE, 5
REAL, 5
E, 6

TYPE, 7
REAL, 7
E, 6, 7
```

```
!********DEFINING BOUNDRY CONDITIONS & FORCES********!

D, 1, UX, , , 7, 6,        !Node 1 & 7 constrained along x axis
D, 2, UX, , , 6,2          !Nodes 2,4 & 6 constrained along x axis
D, 2, ROTZ, , , 6, 4     !Rotation about Z axis constrained for nodes 2&6

FINISH

*cfopen, results, txt

!****************************************************!
!          RUN LOOP FOR ALL CASES
!****************************************************!

*DO, I, 1, POPSIZE, 1

/PREP7

Kf =XDVAR (I, 1)        ! Kf & Af =spring & damping constant
Af =XDVAR (I, 2)        ! Of front suspension
Kar =XDVAR (I, 3)       ! Kr & Ar=spring & damping constant
Ar =XDVAR (I, 4)        ! of rear suspension
Kbr =0.215*Kar

*VWRITE, Kf             ! Write the design variables to results file
(E22.14)
*VWRITE,Af
(E22.14)
*VWRITE,Kar
(E22.14)
*VWRITE,Ar
(E22.14)

R,3,Kf,Af
R,4,,Ar

R,9,-1,-Kar,-0.5,-0.5*Kar,0.0,0.0  !Non Linear Spring Data
RMORE,0.215,Kbr,0.5,0.5*1.28*Kar,1,1.28*Kar

FINISH
```

```
!***************SOLUTION PHASE**************************!

/SOLU

D, 1, UY,
D, 7, UY, 0.021631

ANTYPE, TRANS
!*
TRNOPT, FULL
LUMPM, 0
!*
SOLCONTROL, OFF

NSUBST, 1
OUTRES, ERASE
OUTRES, NSOL, ALL
TIME, 0.036
LSWRITE, 1,

D, 7, UY, 0.041145
TIME, 0.072
LSWRITE, 2,

D, 7, UY, 0.056631
TIME, 0.108
LSWRITE, 3,

D, 7, UY, 0.066574
TIME, 0.144
LSWRITE, 4,

D, 7, UY, 0.07
TIME, 0.18
LSWRITE, 5,

D, 7, UY, 0.066574
TIME, 0.216
LSWRITE, 6,

D, 7, UY, 0.056631
TIME, 0.252
LSWRITE, 7,
```

```
D, 7, UY, 0.041145
TIME, 0.288
LSWRITE, 8,

D, 7, UY, 0.021631
TIME, 0.324
LSWRITE, 9,

D, 7, UY, 0
TIME, 0.36
LSWRITE, 10,
!*
NSUBST, 20
TIME, 1.08
LSWRITE, 11,
NSUBST, 27
TIME, 2.052
LSWRITE, 12,

D, 1, UY, 0.021631
NSUBST, 1
TIME, 2.088
LSWRITE, 13,

D, 1, UY, 0.041145
TIME, 2.124
LSWRITE, 14,

D, 1, UY, 0.056631
TIME, 2.16
LSWRITE, 15,

D, 1, UY, 0.066574
TIME, 2.196
LSWRITE, 16,

D, 1, UY, 0.07
TIME, 2.232
LSWRITE, 17,

D, 1, UY, 0.066574
TIME, 2.268
```

```
LSWRITE, 18,

D, 1, UY, 0.056631
TIME, 2.304
LSWRITE, 19,

D, 1, UY, 0.041145
TIME, 2.34
LSWRITE, 20,

D, 1, UY, 0.021631
TIME, 2.376
LSWRITE, 21,

D, 1, UY, 0
TIME, 2.412
LSWRITE, 22,

LSSOLVE, 1, 22, 1,

FINISH
!***************POST PROCESSING************************!
/POST26

NSOL, 2, 4, U, Y
ABS, 3, 2,  , , AMPL4UY

*GET, AMPMAX, VARI, 3, EXTREM, VMAX

!**************CONSTRAINT FUNCTION EVALUATION***********!

*GET, TVMAX, VARI, 3, EXTREM, TMAX

DLT=0.036
T2=TVMAX+2*DLT
T1=TVMAX+DLT
TM1=TVMAX-DLT
TM2=TVMAX-2*DLT

*GET, F2, VARI, 3, RTIME, T2
*GET, F1, VARI, 3, RTIME, T1
*GET, FM1, VARI, 3, RTIME, TM1
*GET, FM2, VARI, 3, RTIME, TM2
```

```
ACLRT= (F2-2*F1+2*FM1-FM2) / (2*DLT*DLT*DLT)
*IF, ACLRT, LT, 0, THEN
JERK=-1*ACLRT
*ELSE
JERK=ACLRT
*ENDIF

!****RECORD OBJECTIVE FUNCTION AND CONSTRAINT FUNCTION***!

OBJF=AMPMAX

*VWRITE, JERK
(E22.14)

*VWRITE, OBJF
(E22.14, 2X)

*ENDDO                    ! END of DO LOOP for POPULATION CASES

*cfclos
!*****************************************************!
! Write solutions for all population set to pdata.var file
!*****************************************************!
rows =NUMDESV + 1 + NUMOBJ
*DIM, DATA, ARRAY, rows, POPSIZE
*VREAD, DATA (1,1), results, txt, , IJK, rows, POPSIZE
(E22.14)
*MWRITE, DATA (1, 1), pdata, var, , JIK, POPSIZE, rows
(100(E22.14, 2X))

FINISH

/DELETE, results, TXT
/DELETE, 2DVS, EMAT
/DELETE, 2DVS, ESAV
/DELETE, 2DVS, FULL
/DELETE, 2DVS, RST
/DELETE, 2DVS, MNTR
/DELETE, 2DVS, PVTS
/DELETE, 2DVS, BCS

/EXIT, NOSAVE
```

REFERENCES

[1]. S. Rao, Engineering Optimization – Theory and Practice. NY: John Wiley & Sons, Inc., 1996.

[2]. Robert D. Cook, Finite Element Modeling For Stress Analysis NY: John Wiley & Sons, Inc., 1995.

[3]. R. J. Yang, S. Kodialam, L. Gu and C.-H. Tho, "Multidisciplinary Design Optimization in a Scalable, High Performance Computing Environment," Structural and Multidisciplinary Optimization, vol. 26, pp. 256-263, Feb. 2004.

[4]. T. W. Simpson, A. J. Booker, D. Ghosh, A. A. Giunta, P. N. Koch and R.-J. Yang, "Approximation Methods in Multidisciplinary Analysis and Optimization-A Panel Discussion," Structural and Multidisciplinary Optimization, vol. 27, pp. 302-313, July. 2004.

[5]. Ashok D Belligundu and Tirupathi R. Chandrupatla, Optimization Concepts And Application in Engineering. Prentice Hall India

[6]. Schittkowski, K., "On the convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangian Line Search function," Optimization, Mathematische operationsforschung and Statistik, vol. 14, 197-216, 1983.

[7]. Pshenichny, B. N. and Danilin, Y.M., Numerical Methods in External Problems, Mir Publishers, Moscow, second printing 1982.

[8]. Holland J.H., Adaptation in natural and artificial system, Ann Arbor, The University of Michigan Press, 1975.

[9]. ANSYS V 9.0 Documentation Manual, Advanced Analysis Techniques Guide-Design Optimization-Optimization Techniques

[10]. Schmit L. A., "Structural design by systematic synthesis", Proceedings, 2nd conference on electronic computation, ACSE, New York, pp. 105-122, 1960.

[11]. S-Y. Chen, "Integrating ANSYS with Modern Numerical Optimization Technologies", ANSYS Solutions Magazine, Published in Winter Issue 2001.

[12]. W.J.Roux, Nielen Stander and R.T. Haftka, "Response Surface Approximation for structural Optimization". 1998 International Journal of numerical methods in engineering, vol. 42. , Issue 3, pp 517-534.

[13]. R. Jin, W. Chen and T. W. Simpson, "Comparative study of metamodeling techniques under multiple modeling criteria," Journal of structural and Multidisciplinary Optimization, vol. 23, No. 1, 2001, pp. 1-13.

[14]. Powell M. J. D., 1987 Radial Basis Function for Multivariable Interpolation: A Review. In: Mason, J. C.: Cox, M. G. (eds), Algorithms For Approximation. Oxford University press.

[15]. Hongbing Fang, "Multi-objective Crashworthiness Optimization with Radial Basis Function," AIAA 2004-4487

[16]. R.L Hardy, "Theory and Application of the Multiquadric-bi harmonic method". Comp. Math. Application, vol. 19, n. 8/9, pp. 163-208 1990.

[17]. E. J. Kansa, “Multiquadrics – A scattered data approximation scheme with applications to computational fluid dynamics,” I. Comput. Math. Appl. 19, 127-145.

[18]. B. P. Wang, “Parameter Optimization in Multiquadric Response Surface Approximations,” Structural Multidisciplinary Optimization, vol. 26, pp. 219-223.

[19]. Anthony A. Giunta, “Overview of Modern Design of Experiment Methods for Computational Simulations,” AIAA 2003-0649.

[20]. Deb K. and Saxena V., “Car Suspension design for Comfort using Genetic Algorithm”.

[21]. T. W. Simpson, J. D Peplinski and P. N. Koch “Metamodels for Computer based Engineering Design,” Engineering with Computers (2001), vol. 17, pp. 129-150.

[22]. Mark J. L. Orr, “Introduction to Radial Basis Function networks,” April 1996.

BIOGRAPHICAL INFORMATION

Ajaykumar Menon was born on June 2nd, 1979 in Thrissur, India, the eldest son of M. Vijaykumar and Amminikutty Vijaykumar. He enrolled in Lal Bahadur Shastri College of Engineering, Kerala, India in 1998 and earned his B. Tech. (Bachelor of Technology) in Mechanical Engineering in June 2002.In his senior year; he undertook an internship at the Apollo Tyres Ltd in India. He began his career in the field of design engineering at Forbes Marshall Ltd., Pune, India, where he has was employed as an Engineer at their valves division from May 2003 to April 2004.

Ajaykumar Menon had started his M.S. (Master of Science) degree in Mechanical Engineering at The University of Texas at Arlington, Arlington, Texas in June 2004 and earned M.S. Degree in Mechanical Engineering in December 2005 with a 4.0 G.P.A. During his tenure as a master's student, he served as the Graduate Teaching Assistant of Dr. Wen S. Chan, Dr. John Kebrle and Dr. Zhen Xue Han. His thesis was based on the development of a design automation code in MATLAB that incorporated a finite element tool (ANSYS) and a modern optimization concept based on design of experiments and response surface modeling (MQR). His current research interests are finite elements, structural dynamics, structural optimization and computer aided design.