

Multi Disciplinary Design Optimization of Vertically Landing Stage Recovery of Launch Vehicle

A project report submitted
in partial fulfillment for the award of the degree of

Bachelor of Technology

in

Aerospace Engineering

by

Dasu Deva Karthik Lakshman



**Department of Aerospace Engineering
Indian Institute of Space Science and Technology
Thiruvananthapuram, India**

May 30, 2022

Certificate

This is to certify that the project report titled ***Multi Disciplinary Design Optimization of Vertically Landing Stage Recovery of Launch Vehicle*** submitted by **Dasu Deva Karthik Lakshman**, to the Indian Institute of Space Science and Technology, Thiruvananthapuram, in partial fulfillment for the award of the degree of **Bachelor of Technology in Aerospace Engineering** is a bona fide record of the original work carried out by him/her under my supervision. The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Pankaj Priyadarshi
(Guide)
Scientist/Engineer 'H' and Head
ACDD/AERO Divison
Vikram Sarabhai Space Center

Dr. Raveendranath. P
(Internal Guide)
Adjunct Professor
Department of Aerospace Engineering
Indian Institute of Space Science and
Technology

Dr. Aravind Vaidyanathan
Professor and Head of the Department
Department of Aerospace Engineering
Indian Institute of Space Science and
Technology

Place: Thiruvananthapuram
Date: May 30, 2022

Declaration

I declare that this project report titled ***Multi Disciplinary Design Optimization of Vertically Landing Stage Recovery of Launch Vehicle*** submitted in partial fulfillment for the award of the degree of **Bachelor of Technology** in **Aerospace Engineering** is a record of the original work carried out by me under the supervision of **Dr. Pankaj Priyadarshi and Dr. Raveendranath. P.**, and has not formed the basis for the award of any degree, diploma, associateship, fellowship, or other titles in this or any other Institution or University of higher learning. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

Place: Thiruvananthapuram

Dasu Deva Karthik Lakshman

Date: May 30, 2022

(SC18B010)

Acknowledgements

I acknowledge the guidance and support of my advisors, Dr. Pankaj Priyadarshi and Dr. Raveendranath P. throughout the project duration. My interactions with them are a source of inspiration. I would like to express my sincere gratitude towards Dr. Pankaj Priyadarshi sir for giving this opportunity to work on unique and one of its kind reusable stage recovery problem. I was fortunate enough to have quality discussions with him. I acknowledge Dr. Sudeesh Chethil for the support and for providing genuine finite element software with high performance computing. I would also like to thank my parents, friends for their constant support and encouragement.

Dasu Deva Karthik Lakshman

Abstract

Vertical Take-off Vertical Landing(VTVL) Launch Vehicle Stage recovery is very challenging and popular aspect in reusable launch vehicle design. Reusable launch vehicle provides cost-effective and environment friendly access to the space. Design of vertical landing stage recovery is a complex system engineering problem in which many subsystems work together to bring the reusable stage to near zero body rates and velocity. This requires deep throttling of engine which may cause severe instabilities. A novel *Spear In Sand* recovery method which do not require deep throttling is proposed in VSSC in which the launch vehicle stage attached with deployable spears is allowed to land in a fully prepared sand bed with non-zero velocities and body rates. The present work deals with the three dimensional simulation and analysis of the spear penetration in sand using finite element method and this needs to be integrated with trajectory to form a multi-disciplinary analysis and design optimization problem. It is computationally expensive to directly link the finite element analysis with trajectory, instead a metamodel or surrogate model for the finite element analysis is built using *Radial Basis Function Neural Network (RBFNN)* and *Gradient Based Adaptive Sampling*. A python code is developed for RBFNN and Gradient Based Adaptive Sampling. Multi-Disciplinary Design Optimization is done using the obtained metamodel and trajectory, and the results are compared with the traditional sequential optimization method. *Differential Evolution* is used as the optimization solver for both the optimization methods.

Contents

List of Figures	xi
List of Tables	xv
List of Algorithms	xvii
1 Introduction	1
1.1 Vertically Landing Re-Entry trajectory	2
1.2 Novel Spear in Sand Approach for Landing	3
1.3 Work Flow	4
2 Review of Related Literature	5
2.1 Scope of the current work	8
3 Landing Dynamics Module	9
3.1 Finite Element Analysis of Spear In Sand Penetration	9
3.2 Two Dimensional Axi-symmetric Analysis	15
3.3 Two Dimensional Plane Strain Analysis	19
3.4 Three Dimensional Analysis	22
4 Metamodel Based Design Optimization	30
4.1 Introduction	30
4.2 Metamodels	31
4.3 Design of Experiment(DoE)	34
4.4 Metamodel Validation	39
4.5 Invalid Sampling Points	41
4.6 Test Functions	41
4.7 Metamodel for the Finite Element Analysis	53

5 Re-Entry Trajectory Module	54
5.1 Equations of Motion	54
5.2 Methodology	56
5.3 Re-Entry Trajectory Optimization	57
5.4 Multi-Disciplinary Design Optimization	58
5.5 Problem Statement	59
6 Results and Discussions	62
6.1 Metamodel for the Finite Element Analysis	62
6.2 Optimization	64
6.3 Comparison of results	65
7 Conclusions	68
7.1 Future Scope	69
Bibliography	69

List of Figures

1.1	Vertically Landing Stage Recovery of SpaceX Falcon 9 First Stage[1]	1
1.2	Toppling and falling of Falcon 9 First Stage[2]	2
1.3	Typical Launch Profile of SpaceX Falcon 9[3]	3
1.4	Concept of Vertically Landing Reusable Stage Recovery on sand bed	4
3.1	Mohr-Coulomb Yield Criterion[4]	11
3.2	Yield Surfaces of Mohr-Coulomb and Linear Drucker-Prager Plasticity Models in Three-Dimensional Deviatoric Space[5]	12
3.3	Analysis using ordinary meshing(left) and ALE Auto-Adaptive Remeshing(right)[4]	14
3.4	Two Dimensional Axi-Symmetric Finite Element Model with boundary conditions, fields and axis	15
3.5	Meshing for the Two Dimensional Axi-Symmetric Finite Element Model .	16
3.6	Displacement Contour(left), Revolved view of Displacement Contour(right)	17
3.7	Depth of Penetration vs Time using Mohr-Coulomb Plasticity Model(blue) and Drucker-Prager Plasticity Model(red)	17
3.8	Mesh Convergence Study	19
3.9	Two Dimensional Plane Strain Finite Element Model with boundary conditions and fields	20
3.10	Meshing for the Two Dimensional Plane Strain Finite Element Mode	20
3.11	Displacement Contour of Two Dimensional Plane Strain ANalysis	21
3.12	Depth of Penetration vs Time using Mohr-Coulomb Plasticity Model(blue) and Drucker-Prager Plasticity Model(red)	21
3.13	Depth of penetration vs time for 2D Axi-Symmetric Analysis(red) and 2D Plane Strain Analysis(blue) using Drucker-Prager Plasticity Model	22
3.14	Deformation in Lagrangian Analysis(top) and Eulerian Analysis(bottom)[6]	23
3.15	Validation of Freely Falling Penetrometer using 2D Axi-symmetric(left) and 3D CEL analyses(right)	25

3.16	Validation Study and Mesh Dependency Study	25
3.17	Normalized Depth of Penetration vs Normalized time	27
3.18	Reusable Stage without Landing Legs	27
3.19	Landing Legs with CG position	28
3.20	3D CEL Finite Element Model and 3D meshing	29
4.1	The concept of metamodeling for the case of two design variables[7]	31
4.2	Various Sampling Categories[8]	34
4.3	Two Dimensional Grid Sampling	35
4.4	Latin Hypercube Sampling	35
4.5	General flowchart of an adaptive sampling approach[8]	36
4.6	Peaks Function	43
4.7	Contour of Peaks Function	43
4.8	Sampling Points obtained from a) Grid Sampling, b) Latin Hypercube Sampling and c) Adaptive Sampling	44
4.9	Easom Function	45
4.10	Contour of Easom Function	45
4.11	Sampling Points obtained from a) Grid Sampling, b) Latin Hypercube Sampling and c) Adaptive Sampling	46
4.12	a) Fonesca Fleming Function f_1 and b) Fonesca Fleming Function f_2	48
4.13	a) Contour of Fonesca Fleming Function f_1 and b) Contour of Fonesca Fleming Function f_2	48
4.14	Sampling Points obtained from a) Grid Sampling, b) Latin Hypercube Sampling and c) Adaptive Sampling	49
4.15	a) Viennet Function f_1 , b) Viennet Function f_2 and c) Viennet Function f_3 .	51
4.16	a) Contour of Viennet Function f_1 , b) Contour of Viennet Function f_2 and c) Contour of Viennet Function f_3	51
4.17	Sampling Points obtained from a) Grid Sampling, b) Latin Hypercube Sampling and c) Adaptive Sampling	52
5.1	Geometry of the Reusable Stage	55
5.2	Process of Differential Evolution	60
6.1	Leave one out cross validation error vs functional evaluation for adaptive sampling	63
6.2	Results obtained using traditional Sequential Optimization Method	66

6.3	Objective Function vs Number of Iterations using Differential Evolution . . .	66
6.4	Results Obtained from Multi-Disciplinary Design Optimization	67
6.5	Objective Function vs Number of Iterations using Differential Evolution . .	67

List of Tables

3.1	Known Parameters of the spear from the previous work	9
3.2	Maximum Depth of Penetration for both Plasticity Models	16
3.3	Estimated Parameters of Sand	18
3.4	Experimental Conditions for validation study[9]	24
3.5	Properties of Reusable Stage(courtesy of VSSC)	27
4.1	Radial Basis Functions	32
4.2	Comparsion of Sampling Techniques for Peaks Function	42
4.3	Comparsion of Sampling Techniques for Peaks Function	45
4.4	Comparsion of Sampling Techniques for Fonseca Fleming Function	47
4.5	Comparsion of Sampling Techniques for Viennet Function	50
5.1	Boundary Conditions	58
5.2	Range of Design Variables	58
6.1	Leave-one-out Cross Validation Error for all basis funtions for $c=0.1$	62
6.2	Comparison of sampling techniques	63
6.3	Optimized Design Variables and terminal conditions	65
6.4	Accelerations and depths of penetration obtained from metamodel	65

List of Algorithms

5.1	Differential Evolution	61
-----	------------------------	----

Chapter 1

Introduction

Recovery of initial spent stages of a launch vehicle and reusing them is very crucial to the cost effective access to space. Many space agencies have already tried and have been trying to recover and reuse the spent stages effectively to cut the costs. There are numerous methods researchers have proposed for the recovery of spent stages, out of which the Vertical Take-off Vertical Landing(VTVL) recovery method is the most successful and reliable method, which is currently used by leading space agencies like *SpaceX*, *Blue Origin*, etc. In Vertical Take-off Vertical Landing(VTVL) recovery method, launch vehicle take-off vertically and after stage separation, the reusable stage performs two or three maneuvers and retropropulsive burns and lands vertically to a predetermined location either on earth or in ocean(drone ship), as shown in Fig. 1.1.



Figure 1.1: Vertically Landing Stage Recovery of SpaceX Falcon 9 First Stage[1]

The most challenging part is that the velocity and body rates must be near to zero at touchdown for safe landing without damages the stage. The reusable stage must be per-

fectly vertical in order to avoid the toppling tendency after landing. Even though if the above two conditions met, if there are any disturbances in the landing site like waves induced motion to the drone ship, heavy winds while landing on earth, etc., the reusable stage would topple and falls down(as shown in Fig. 1.2) which cannot be reused any more.



Figure 1.2: Toppling and falling of Falcon 9 First Stage[2]

The reusable stage consists of various subsystems like trajectory, propulsion, landing leg deployment dynamics, structures, aerodynamics control surfaces, etc. and all these subsystems are required to work hand in hand to recover the reusable stage optimally. The interactions of these subsystems plays a major role in designing of reusable launch vehicles. The design of reusable launch vehicles is a *Multi-Disciplinary Design Optimization Problem* in which one must consider all the disciplines/subsystems to get the design. The major disciplines used in the current work are Re-entry Trajectory, Landing Dynamics which are discussed in the sections. 1.1 and 1.2.

1.1 Vertically Landing Re-Entry trajectory

The typical launch profile of vertically landing reusable launch vehicle is similar to the SpaceX Falcon-9 launch profile shown in Fig. 1.3. The re-entry trajectory of reusable stage starts after its separation from upper stage(s). After the separation, the reusable stage undergoes *flip maneuver* in which the angle of attack change from near zero to near 180° using the aerodynamic grid fins and roll control thrusters. After that a boostback burn followed by an entry burn are burned to steer the reusable stage towards land site and to reduce the velocity of the stage. Then the reusable stage is allowed to fall freely using the

aerodynamic drag to decelerate and guide the reusable stage to the precise land site. Finally, a landing burn is burned to bring the velocity to zero and to make the vehicle vertical.

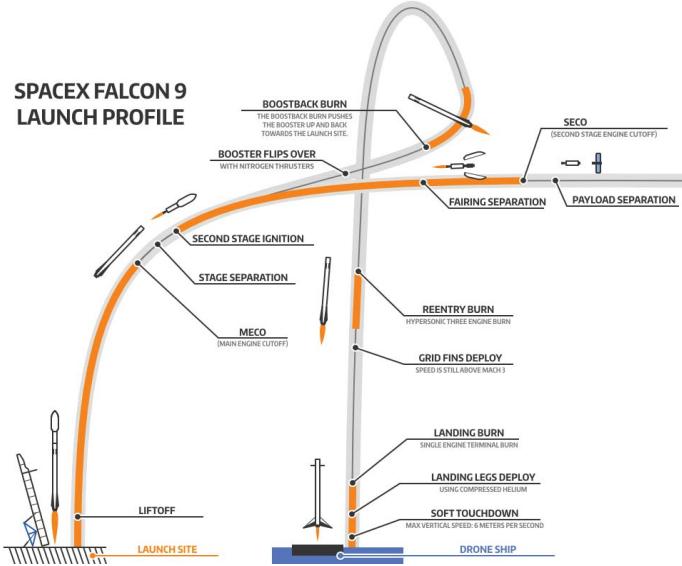


Figure 1.3: Typical Launch Profile of SpaceX Falcon 9[3]

1.2 Novel Spear in Sand Approach for Landing

The Vertical Take-off Vertical Landing recovery method comes under the propulsive recovery methods in which the engine need to burn multiple times to recover the stage. Leading space agencies like *SapceX* uses only one merlin engine out of nine for deep throttling to balance the thrust to weight ratio, *Blue Origin* uses retropropulsion and hovers before the landing. Deep throttling can cause severe combustion instabilities inside the engine and may lead to catastrophic failure of the mission. An Intense closed loop guidance between propulsion and aerodynamic surfaces is required to make velocities and body rates in all directions equal to zero at the time of touchdown. Propulsive recovery method in the context of present ISRO launch vehicles is not feasible because of large thrust to weight ratio due to presence of only one vikas engine. An alternate method, known as *Spear In Sand Method* has been proposed in VSSC[10](shown in Fig. 1.4) which can eliminate the need for deep throttling and hovering. In this method, the reusable stage is decelerated to minimum possible non-zero velocity and non-zero body rates without using deep throttling or hovering and instead of landing on a rigid launch pad, the stage lands on a designed sand bed in which the residual kinetic energy in the reusable stage is dissipated in the form of

penetration inside the sand[11]. And also this method allows non-zero attitude and attitude rates of the stage at touchdown which eliminates the intense closed loop guidance. The depth of penetration in the sand provides anchoring to the stage which do not topple when there are any disturbances due to environment.

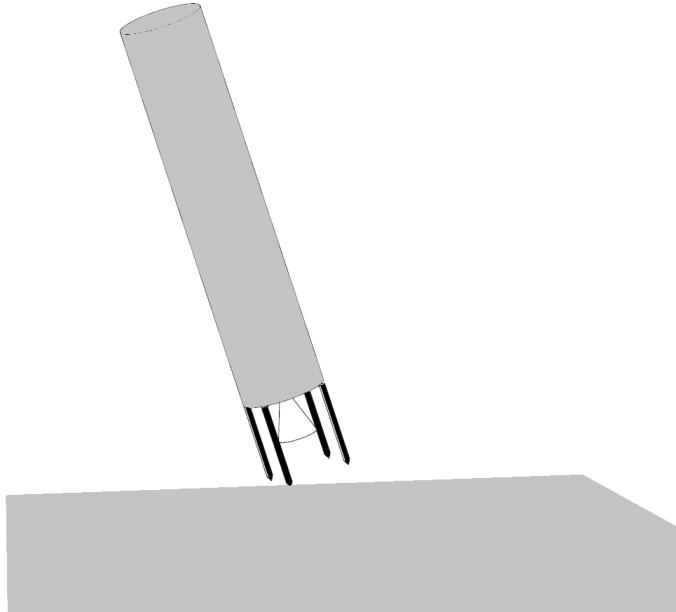


Figure 1.4: Concept of Vertically Landing Reusable Stage Recovery on sand bed

1.3 Work Flow

The current work deals with the Multi-Disciplinary Design Optimization of Vertically Landing Stage Recovery of Launch vehicle. This report is organised as follows, Chapter. 2 provides a brief summary of literature related to Re-Entry Trajectory Optimization, Dynamics of Spear inside the sand and Optimization Methods. Chapter. 3 elaborates the approaches, methods for the simulation of Landing Dynamics of spear in sand using finite element method. Chapter. 4 describes the various sampling methods and algorithms for generating sampling points to build accurate metamodels, test cases are also presented to compare the various sampling methods. Chapter. 5 describes the methodology, formulation and optimization of Vertically Landing Re-Entry Trajectory. This chapter also presents the Multi-Disciplinary Design Optimization Method and optimization solver used for the current work. The results, analyses and discussions for the re-entry trajectory, metamodel, sampling methods, and optimization are presented in Chapter. 6. The conclusions and future work in Chapter. 7 concludes the report.

Chapter 2

Review of Related Literature

This chapter reviews the major works which are directly and indirectly related to the current work. As described earlier, the stage recovery in the current work is comprises of two major disciplines: Re-Entry Trajectory and Landing Dynamics of spear in sand.

Vertical Landing re-entry trajectory optimization is gaining popularity day by day due to its complexity, highly non-linear and non-convex nature. So far, researchers optimized vertical landing re-entry trajectory on near vacuum areas like moon, etc which eliminates the highly non-linear drag effects in the trajectory. *Anglim et. al*[3] included the non-linear effects of drag in the trajectory optimization, the entire trajectory is divided into three phases, boostback burn, freefall and landing burn, and rather than optimizing the trajectory using control variables directly, the author assumed bang-off-bang thrust control with predetermined attitude or thrust angle, so that the design variables now are only time lengths of the three phases. The optimization problem is framed as minimum fuel and minimum terminal error problem in which summation of squares of L2 norms of difference between required terminal conditions and obtained terminal conditions along with negative of terminal mass is taken as the objective function. Heuristic Method(PSO) is used to solve the optimization problem.

Liu et. al[12] proposed a new re-entry trajectory optimization method in which both aerodynamic controls and thrust controls are used as the design variables at the same time and the author aimed at onboard real time re-entry trajectory optimization for precise landing by convexifying the highly non-linear, non-convex NLP problem and non convex constraints to solve the optimization problem using convex optimization.

Zhao et.al[13] proposed new optimization method for solving re-entry trajectory problem, the author considered three dimensional point mass trajectory and a total of four trajectory phases, *Attitude Adjusting Phase*, *Powered Descending phase*, *Aerobraking Phase* and *Vertical Landing Phase*, here the trajectory is optimized directly by optimizing the

control variables, a linear time profile is assumed for angle of attack, bank angle in powered descent and vertical landing phases and the constants in the linear profile are taken as design variables and the time lengths required for each trajectory phase, and throttling parameters in powered descending, vertical landing phases are also considered as the design variables. Finally, the optimization problem is handled using a clustering-based adaptive optimization technique.

There are no direct references available for the current work on spear in sand penetration, but cone penetration tests(CPT) and freely falling penetrometer(FFP) tests to measure the soil properties in geotechnical engineering are very similar to the current spear in sand penetration. Numerical modelling of these tests using finite element analysis gained popularity after the development of computational technology because of their sophisticated and challenging nature. Most of the researchers modelled penetrometer as rigid as body and soil as perfectly elastic and plastic material. Researchers have used various plasticity models used for modelling soil. *Susila et. al*[14] simulated the cone penetration in normally consolidated sand with friction as two-dimensional axi-symmetric problem in ABAQUS Finite Element Software using explicit integration scheme and auto adaptive remeshing to handle the large mesh distortion, the author used Drucker-Prager plasticity to model the soil behaviour. Freely falling penetrometer tests are more analogous to the current work of spear in sand penetration than ordinary cone penetration tests because in CPT the penetrometer is pushed continuously into the soil whereas FFP are allowed to fall under gravity but methodologies used to simulate both tests are same.

Nazem et. al[9] simulated the freely falling cone penetrometer into uniform clay without friction as two-dimensional axi-symmetric problem, the same methodology which is used for CPT is used here and the behaviour of soil is modelled using much simple Tresca Plasticity model. The author also demonstrated the influence of soil strain hardening on penetration, and the simulated and experimental data were compared.

So far, the CPT and FFP tests are simulated as two-dimensional axi-symmetric problems rather than actual three dimensional problems, *Falla et. al*[15] used Coupled Eulerian Lagrangian(CEL) method to model cone penetration tests in three-dimensions in which soil is modelled as eulerian body and penetrometer is modelled as rigid body and the soil behaviour is modelled using Tresca Plasticity Model, the author validated the application of coupled eulerian lagrangian method to CPT with standard existing literature, the author conducted a parametric study on the cone bearing factor and the clay rigidity index to determine their relationship, and the results are in good accord with the experimental results.

Ghosh et. al[16] simulated the freely falling circular plate penetrometer into uniform clay with out friction using Coupled Eulerian Lagrangian(CEL) method, the author investigated the influence of mass, velocity, strain rate hardening and other parameters on the penetration response and also the author validated the application of Coupled Eulerian Lagrangian(CEL) method to freely falling circular plate penetrometer with standard studies.

Wang et. al[17] studied the large displacement finie element analysis in geotechnical engineering in which the author chose three different methods, Coupled Eulerian Lagrangian(CEL), implicit Remeshing and Interpolation Technique by Small Strain (RITSS) and Efficient Arbitrary Lagrangian Eulerian(EALE), and to simulate the standard geotechnical problems such as CPT, Buckling of Pipeline, Consolidation under Surface footing and Freely Falling Cone Penetration Test. Though there is no validation for free falling cone penetration test, the author showed that CEL method performed better for dynamic analyses than other methods.

Integration of finite element software with optimization solver is not possible for design optimization, instead a metamodel or surrogate model to the software can be built, which approximates the blackbox simulation with known analytical function, to get the output which is nearly equal to the output from the software for a given input. But the surrogate model requires training dataset containing the inputs and outputs from the software to predict the output with utmost accuracy. The generation of the input data points is known as *Sampling*. The type of analytical function chosen also effects the accuracy of the metamodel. There are many methods of sampling, but *Adaptive Sampling* gaining popularity now a days because of its ability to produce accurate metamodel with less number of datapoints, *Liu et. al*[8] reviewed the present day Adaptive Sampling methods in the context of metamodels, and presented what type of adaptive sampling method required for a given metamodel.

Radial Basis Function Neural Networks(RBFNN) are primarily introduced for interpolation between the datapoints, the RBFNN are simple, analytical and their gradient information is readily available. Because of this reason *Gradient Based Adaptive Sampling* is the ideal choice when the metamodel is RBFNN. *Yao et. al*[18] used the information of first order gradient to sample the new points near the regions where large non-linearity occurs to gather the best information of the function. *Wei et. al*[19] has presented a new approach for sampling new locations that employs second order gradient information.

Martins et. al[20] reviewed several architectures of Multi-Disciplinary Design Optimization methods, all existing MDO architectures fall under two categories, *Monolithic* and *Distributed* architectures. In monolithic architectures, all the disciplines are combined to

form a global optimization problem with only one objective function, whereas Distributed architectures have more than one objective function, one at the global level and the other at the subsystem/discipline level. *Balesdent et. al*[21] surveyed MDO methods in context of launch vehicle design, the author mentioned that more than 70% of the existing literature used monolithic architecture(Multi-Disciplinary Feasible Method) for the design of launch vehicle.

2.1 Scope of the current work

The current work aims to analyze the design of vertically landing reusable stage recovery using the *Novel Spear In Sand* approach and optimize the stage recovery using Multi-Disciplinary Design Optimization.

Re-entry trajectory is divided into a total of three phases, both control variables and time lengths of the phases are considered as design variables. The state space equations are integrated using forward euler method.

Three-Dimensional Landing Dynamics of spear in sand is simulated using Finite Element Analysis to obtain the reaction forces and depth of penetration time histories.

Using metamodel as functional approximation for the black box FEA simulation is the rudimentary concept for the current work. A python code is developed for the metamodel training and gradient based adaptive sampling. Multi-Disciplinary Design Optimization is performed using Re-Entry Trajectory and the obtained Metamodel and Differential Evolution is used as the optimization solver.

Chapter 3

Landing Dynamics Module

As discussed earlier, the residual kinetic energy in the reusable stage is dissipated in the form of penetration in sand. The dynamics and interaction of spear in sand are most important for the design of spears. There are many analytical methods for penetration of objects in sand and soils most of them are empirical relations or derived by using simplified models. Finite Element Analysis(FEA) can implement any kind of constitutive model with ease and helps us to simulate complex real life scenarios with high degree of accuracy to make the optimal design. In the current work, the spear in sand phenomena is simulated using Finite Element Method.

3.1 Finite Element Analysis of Spear In Sand Penetration

From the previous experimental and analytical work[cite ncmdao], the properties of the spear are known and are shown in Table. 3.1.

Table 3.1: Known Parameters of the spear from the previous work

Parameter	Value
Geometry	Conical
Apex Angle	60°
Diameter	0.15m
Length	1.275m
Design Depth of Penetration	0.275m
Velocity	8 m/s
Buckling Load	8072 kN
Load on Spear	2073.032kg

Using the properties of the spear and experimental conditions like velocity and depth of penetration, an inverse analysis is done in **ABAQUS** Finite Element Software to find out

the sand properties and the obtained properties are used in the further analyses.

3.1.1 Sand Constitutive Model

In geotechnical engineering, plasticity models are widely used as constitutive models for analyses using soil/sand. There are various plasticity models and different authors used different plasticity models. There are various plasticity models ranging from simpler *Tresca Model* to advanced *Extended Cam Clay Model*. The number of parameters of the plasticity models increase as the complexity of the model increases. The Drucker-Prager Plasticity Model was chosen as the constitutive model for sand in this study, and the results were compared to the standard Mohr-Coulomb Model. The models are described below,

3.1.1.1 Mohr-Coulomb Plasticity Model

Mohr-Coulomb Plasticity Model is the most extensively used constitutive model in geotechnical engineering to simulate plastic deformations of soil. The main advantage of the Mohr-coulomb plasticity model is that its parameters can be acquired immediately from laboratory studies such as Tri-axial testing. The mohr-coulomb model assumes linear dependency between shear stress and normal stress at the time of yielding[4].

$$\tau = c - \sigma \sin\phi \quad (3.1)$$

The normal stress, σ , is negative in compression, the material's cohesiveness is c , and the angle of internal friction is ϕ . The yield criteria for the Mohr-coulomb model is obtained by drawing the Mohr circles for stresses at yielding in plane of extreme principal stresses. The line passes through these circles gives the Mohr-Coulomb yield criteria(shown in Fig. 3.1). From Mohr's Circle,[4]

$$\tau = s \cos\phi \quad (3.2)$$

$$\sigma = \sigma_m + s \sin\phi \quad (3.3)$$

From above three equations we can simplify all three to one equation,

$$s + \sigma_m \sin\phi - c \cos\phi = 0 \quad (3.4)$$

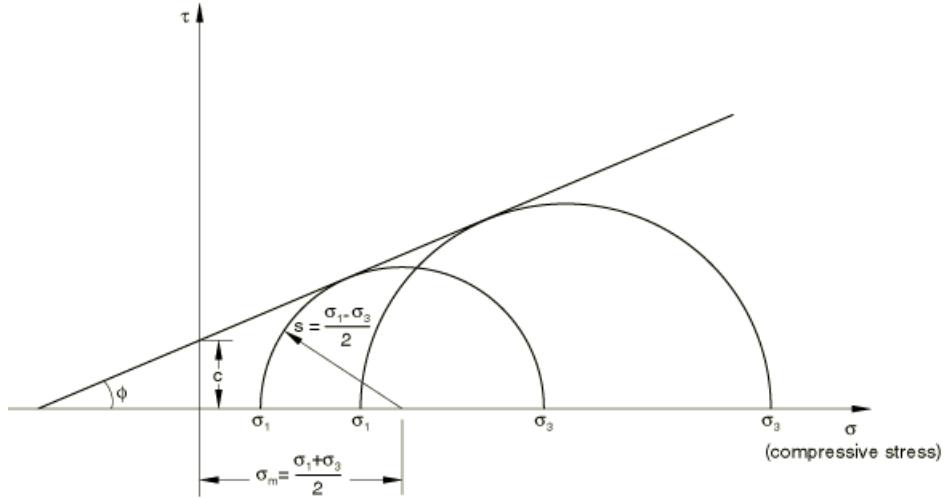


Figure 3.1: Mohr-Coulomb Yield Criterion[4]

where,

$$s = \frac{1}{2}(\sigma_1 - \sigma_3) \quad (3.5)$$

and

$$\sigma_m = \frac{1}{2}(\sigma_1 + \sigma_3) \quad (3.6)$$

The equation(3.4) is the *mohr-coulomb yield criteria*. The major drawback of this model is that the mohr-coulomb yield criteria is independent of intermediate principal stresses, but it holds for most of the soils.

3.1.1.2 Drucker-Prager Plasticity Model

Drucker-Prager Plasticity Model unlike Mohr-Coulomb Plasticity Model considers intermediate principal stresses. It is frequently used to model frictional materials with pressure-dependent yield behaviour, such as granular materials, soils, and sand. There are several yield criteria in the Drucker-Prager model such as linear, hyperbolic, and exponent which are based on the shape of the yield surface in the p-t plane. In the current work, **Linear Drucker-Prager Plasticity Model** is used for the further analyses. The Linear Drucker-Prager Plasticity yield criteria can be written as,[4]

$$F = t - p \tan\beta - d = 0 \quad (3.7)$$

where, p is the equivalent pressure stress

$$p = -\frac{1}{3} \text{trace}(\sigma) \quad (3.8)$$

and q is the Mises equivalent stress

$$q = \sqrt{\frac{3}{2}(S : S)} \quad (3.9)$$

$$S = \sigma + pI \quad (3.10)$$

S is the stress deviator, In addition, the linear model employs the third deviatoric stress invariant,

$$r = \left(\frac{9}{2} S \cdot S : S \right)^{\frac{1}{3}} \quad (3.11)$$

and

$$t = \frac{1}{2}q \left[1 + \frac{1}{K} - \left(1 - \frac{1}{K} \right) \left(\frac{r}{q} \right)^3 \right] \quad (3.12)$$

where β is the p-t plane slope of the linear yield surface, often known as the 'friction angle,' and K is the ratio of the yield stress in triaxial tension to the yield stress in triaxial compression, making the yield criterion a function of intermediate stress. And d stands for material cohesion,

$$d = \left(1 - \frac{1}{3} \tan \beta \right) \sigma_c \quad (3.13)$$

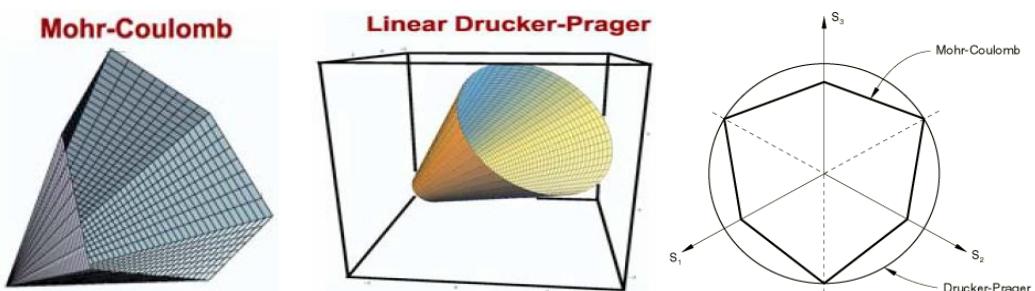


Figure 3.2: Yield Surfaces of Mohr-Coulomb and Linear Drucker-Prager Plasticity Models in Three-Dimensional Deviatoric Space[5]

where, σ_c is the uniaxial compression yield stress, hardening is defined by using this parameter. The advantage of this model is that the yield surface of Drucker-Prager Plasticity Model in deviatoric plane is smoother than Mohr-Coulomb Plasticity Model because of its dependence on intermediate principal stresses(shown in Fig. 3.2). Generally, geotechnical materials like soils/sand have small dependence on the intermediate stresses, therefore Drucker-Prager Plasticity model is chosen for the current work. Mohr-Coulomb Plasticity Model requires three parameters(c , ϕ , ψ) and Drucker-Prager Plasticity Model requires four parameters(σ_c , β , ψ^* , K). As said earlier, these parameters are estimated using the values in Table. 3.1 and are shown in the later sections.

3.1.2 Solution Method

The non-linear differential equations of any sort can be solved using two standard finite element solution methods, *implicit* and *explicit*. Both approaches are supported by ABAQUS. When the number of nodes is considerable, the implicit finite element solution method iteratively solves non-linear differential equations using the Newton-Raphson method, which is computationally demanding. The explicit finite element method is primarily introduced to solve dynamic problems, which uses explicit central difference integration method[14]

$$\dot{u}_{(i+\frac{1}{2})} = \dot{u}_{(i-\frac{1}{2})} + \frac{\Delta t_{(i+1)} + \Delta t_{(i)}}{2} \ddot{u}_{(i)} \quad (3.14)$$

$$u_{(i+1)} = \Delta t_{(i+1)} \dot{u}_{(i+\frac{1}{2})} \quad (3.15)$$

where the increment is i , the displacement is u , and the time increment is Δ_t .

The explicit finite element method uses diagonal (lumped) mass matrices which reduces the computational effort by a great amount. The acceleration($\ddot{u}_{(i)}$) is obtained by

$$\ddot{u}_{(i)} = M^{-1}(F_{(i)} - I_{(i)}) \quad (3.16)$$

M stands for the mass matrix, $F_{(i)}$ for the applied load vector, and $I_{(i)}$ for the inertial force vector. The value of Δt is determined by the conditional stability criteria which states that the time increment must be approximately equal to the time the elastic wave propagate to every element in the model. Explicit finite element method requires less computational resources and also faster than the implicit method. Explicit finite element solution method is chosen for the current work.

3.1.3 Auto-Adaptive Remeshing

For the analyses which include large deformations within less time, the mesh distorts heavily resulting in inaccurate solutions and sometimes premature termination of analysis. Auto-Adaptive remeshing techniques are employed in those scenarios to preserve the quality of mesh throughout the analysis. **Arbitrary Lagrangian Eulerian(ALE)** method is one such remeshing technique and is widely used to preserve the mesh quality throughout the analysis. In ALE method, the material inside the domain is independent of the mesh which is eulerian analysis and the material at the boundaries of the domain is fixed to the mesh which is Lagrangian analysis, hence the name.

The ALE method comes under the *r-adaptivity* scheme in which the mesh changes by relocating the nodes without actually deleting and adding the degrees of freedom, the difference between ordinary meshing and adaptive meshing are shown in Fig. 3.3. In ABAQUS ALE remeshing is performed at an user defined frequency and number of remeshing sweeps, the quality of the mesh increases as the frequency decreases and remeshing sweeps increases because the as the frequency decreases the remeshing occurs more frequently with more remeshing sweeps but at the cost of computation time, and also results might not vary distinctly, so optimal frequency and number of remeshing sweeps are to be chosen. In the current analysis, the number of remeshing sweeps are chosen as '8' with a frequency of '2' and the algorithm used is *conventional smoothing algorithm*.

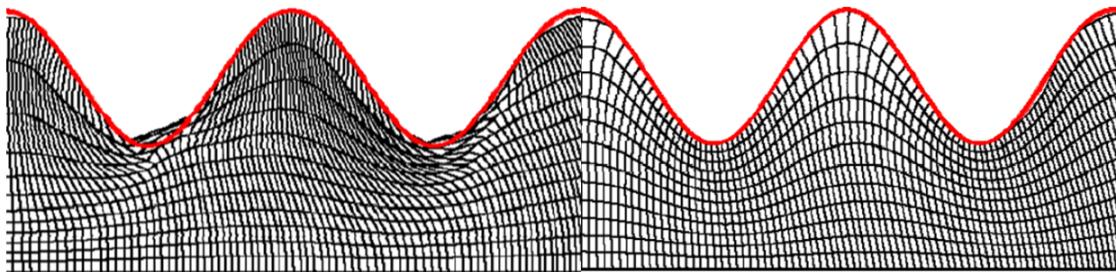


Figure 3.3: Analysis using ordinary meshing(left) and ALE Auto-Adaptive Remeshing(right)[4]

3.2 Two Dimensional Axi-symmetric Analysis

Two Dimensional Axi-Symmetric analysis is done to find out the sand parameters from the values and experimental conditions from Table. 3.1. Fig. 3.4 shows the Two Dimensional Axi-Symmetric Finite Element Model with boundary conditions, fields and axis. The spear is modelled as Discrete Rigid Body. The sand is modelled as linearly elastic and perfectly plastic following Drucker-Prager and Mohr-Coulomb yield criteria. *Master-Slave* type contact with friction($\mu = 0.414$) is modelled between spear and sand.

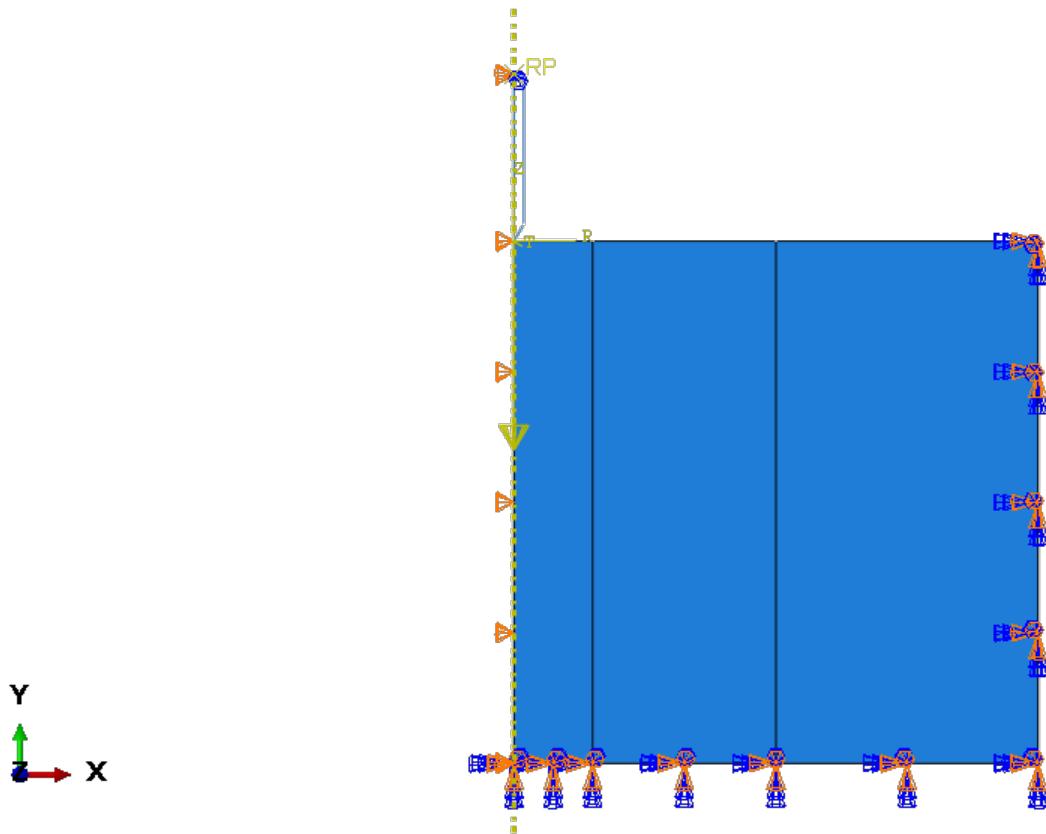


Figure 3.4: Two Dimensional Axi-Symmetric Finite Element Model with boundary conditions, fields and axis

The size of the domain is taken as 4m x 4m and the meshing is done as shown in Fig. 3.5. *Three-noded linear axisymmetric triangular elements(CAX3)* are used for modelling sand and *discrete elements* are used for modelling spear. The density of the mesh near the spear is made thicker than the mesh far away from the spear. A total of 1,44,000 elements are used in meshing of the sand. The type of analysis is done here is *Axi-Symmetric Stress Analysis*

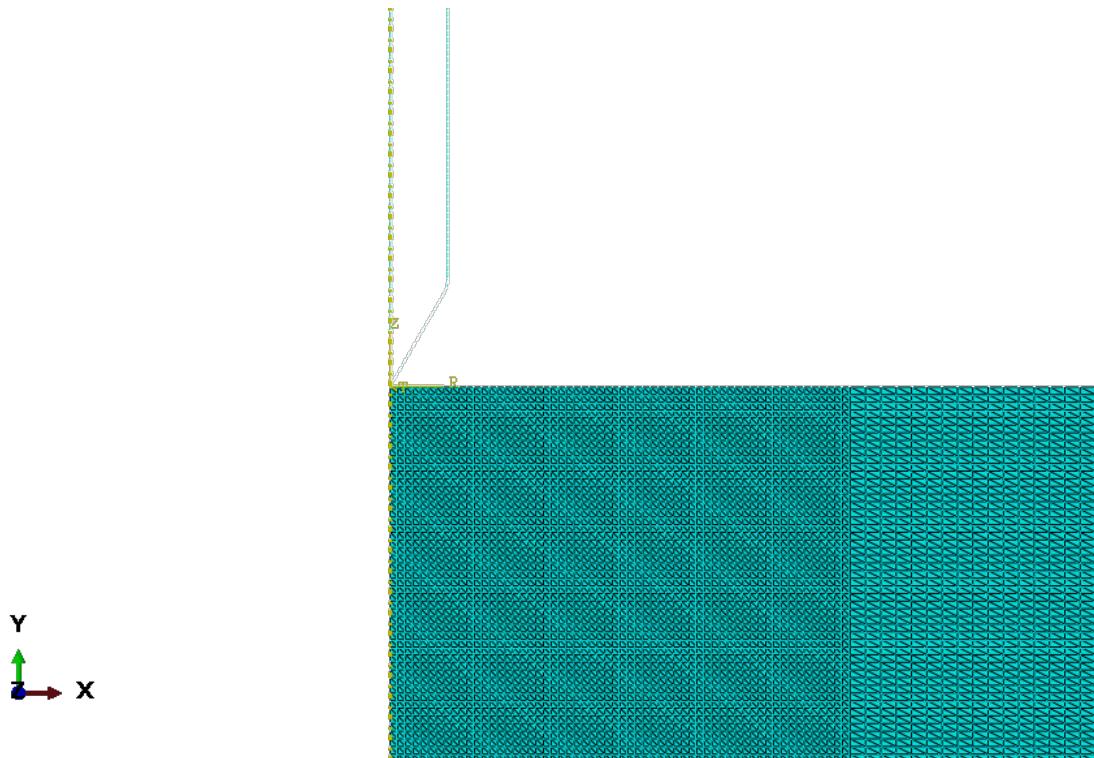


Figure 3.5: Meshing for the Two Dimensional Axi-Symmetric Finite Element Model

3.2.1 Results

Fig. 3.7 shows the depth of penetration vs time using two plasticity models Mohr-Coulomb and Drucker-Prager. It can be seen that the depth of penetration starts increasing and reaching a maximum value at which it comes to rest. It can be noticed that the difference of depth of penetration between Mohr-Coulomb and Drucker Prager is minimal which ensures that Finite Element Model is accurate, and the depth of penetration using Drucker-Prager model is lesser than the Mohr-Coulomb model this is due to the presence of intermediate principal stresses in Drucker-Prager model which causes additional resistance.

Table 3.2: Maximum Depth of Penetration for both Plasticity Models

Model	Mohr-Coulomb	Drucker-Prager
Maximum		
Depth of penetration	0.2877m	0.2758m

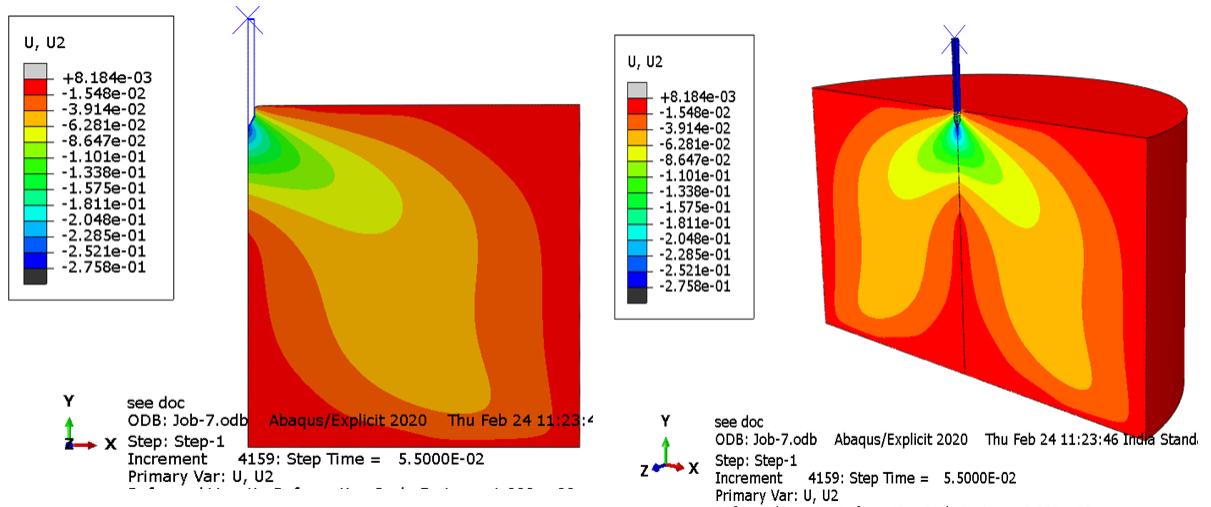


Figure 3.6: Displacement Contour(left), Revolved view of Displacement Contour(right)

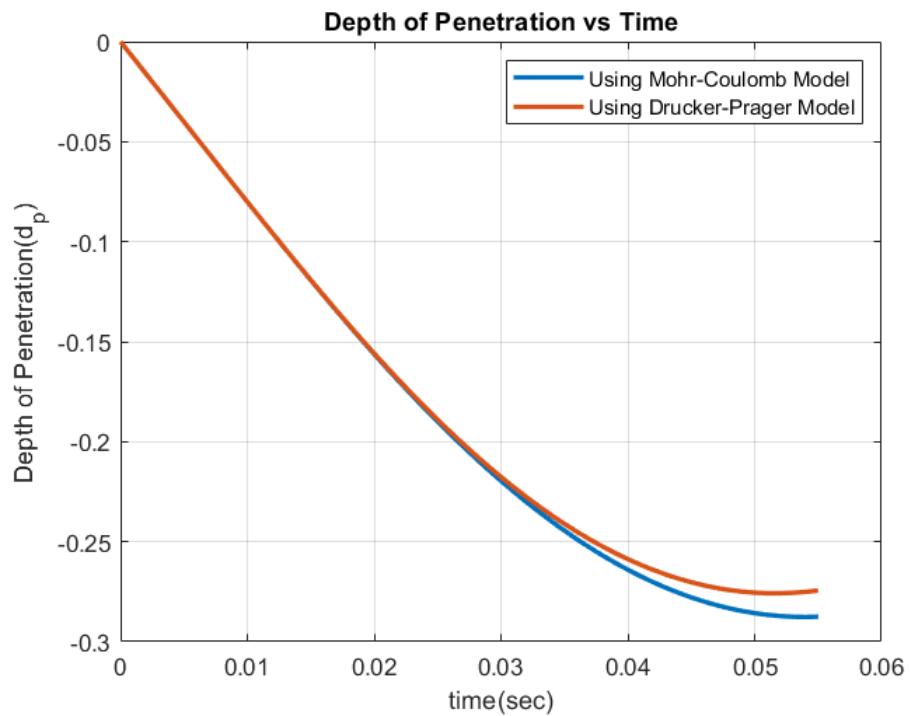


Figure 3.7: Depth of Penetration vs Time using Mohr-Coulomb Plasticity Model(blue) and Drucker-Prager Plasticity Model(red)

3.2.2 Sand Properties

Table. 3.3 shows the estimated parameters of sand from the Two Dimensional Axi-Symmetric Analysis using the known values and experimental conditions of the spear. From the parameters obtained, the type of sand required for the design is **Stiff Clayey Sand**[cite].

Table 3.3: Estimated Parameters of Sand

Elastic Properties of Sand	
Density	1800 Kg/m^3
Young's Modulus	30 MPa
Poisson's Ratio	0.3
Drucker-Prager Parameters	
Friction Angle(β)	50.57°
Dilation Angle(ψ^*)	6.08°
Compressive Hardening(σ_c)	332.404 KPa
Flow Stress Ratio(K)	0.778
Mohr-Coulomb Parameters	
Internal Friction Angle(ϕ)	30.37°
Dilation Angle(ψ)	3°
Cohesion Stress(c)	95.244 Kpa

3.2.3 Mesh Convergence Study

Fig. 3.8 shows the mesh convergence study for the Two Dimensional Axi-Symmetric Analysis using the sand properties from the Table. 3.3. The depth of penetration is normalized with required depth of penetration($d_p = 0.275$), it can be seen that the depth of penetration increasing with number of elements but the slope of the curve is decreasing which ensures the convergence. The amount of displacement increased for the case with 900000 elements is less than 3% of the required value, which we can safely assume as negligible for engineering design. Therefore, we can use the above obtained sand parameters for further analyses.

The major drawback of Two-Dimensional Axi-Symmetric Analysis is that, it can only perform analysis in vertical direction(i.e, Flight Path Angle = 90°), In order to perform inclined analysis in two dimensions, instead of Axi-symmetric analysis, a plane strain analysis is performed and the results are shown in further sections.

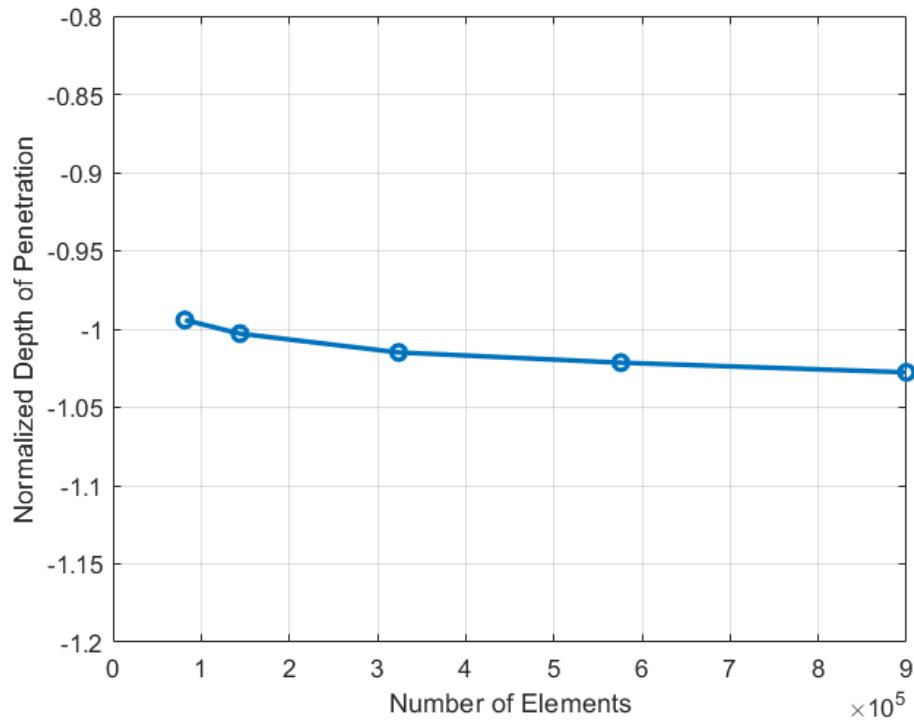


Figure 3.8: Mesh Convergence Study

3.3 Two Dimensional Plane Strain Analysis

Two Dimensional Plane Strain Analysis is performed using the same parameters and methodology as described in the Axi-Symmetric Analysis. The changes that made in the analysis are, type of analysis is changed to *Two Dimensional Plane Strain Analysis* and the type of elements used are *Three-noded linear plane strain triangular elements*. Also, in Axi-Symmetric analysis, only a quarter of the load is taken into account because the analysis occurs in only one quadrant due to rotating axis of symmetry, but in plane strain analysis, half of the load is taken into account due to planar symmetry. The finite element model and mesh are shown in Fig. 3.9 and 3.10.

3.3.1 Results

Fig. 3.11 and Fig. 3.12 shows the displacement contours and Depth of Penetration vs Time for Two Dimensional Plane Strain analysis. The two dimensional axi-symmetric and two-dimensional plane strain analyses are compared in the below section.

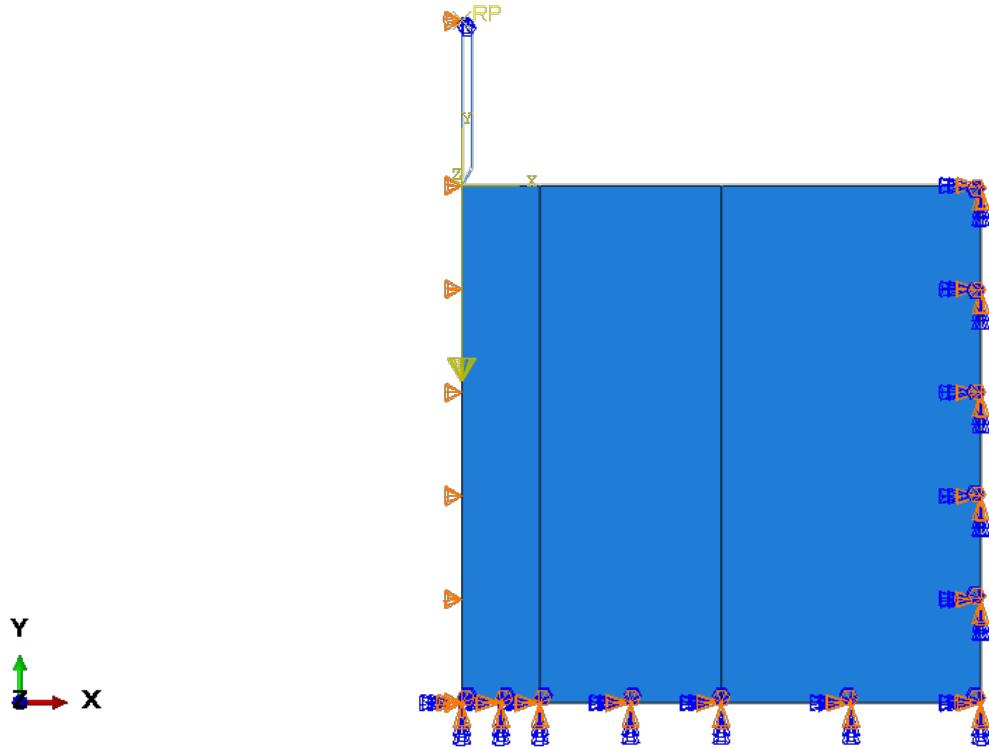


Figure 3.9: Two Dimensional Plane Strain Finite Element Model with boundary conditions and fields

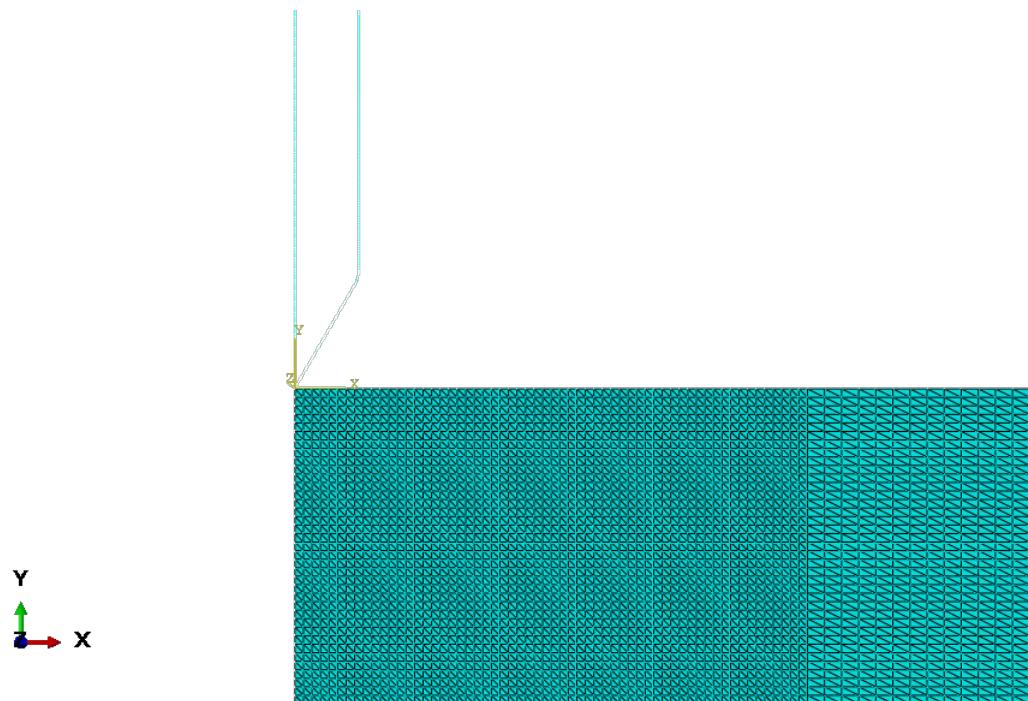


Figure 3.10: Meshing for the Two Dimensional Plane Strain Finite Element Model

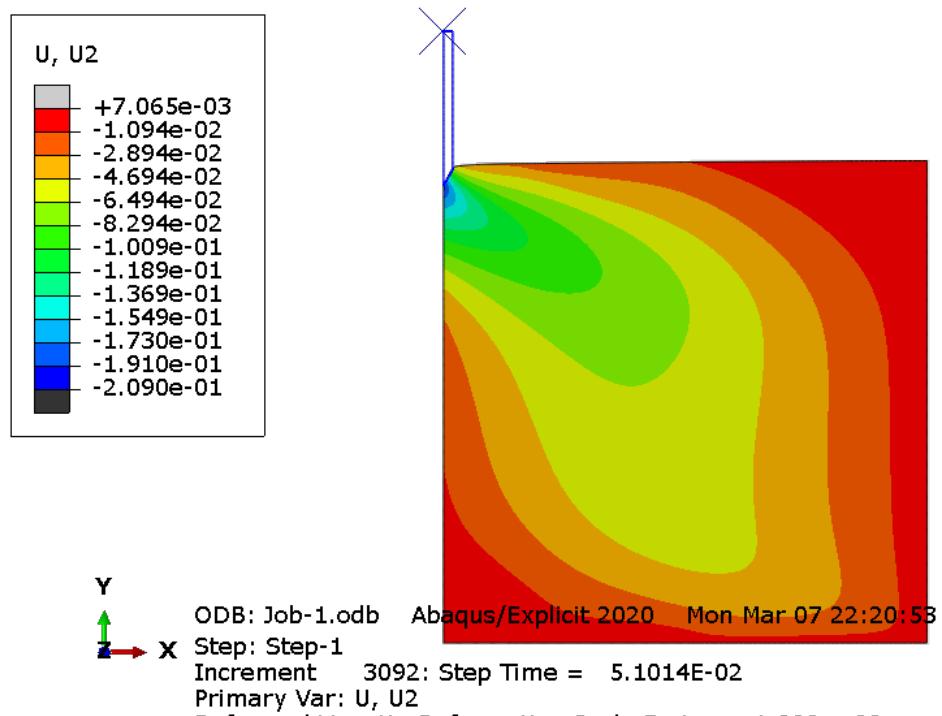


Figure 3.11: Displacement Contour of Two Dimensional Plane Strain ANalysis

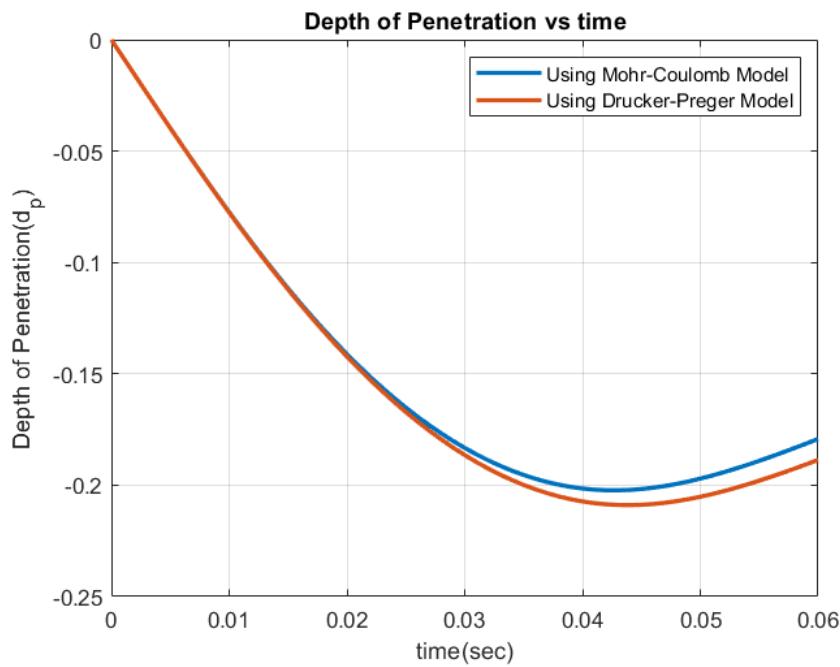


Figure 3.12: Depth of Penetration vs Time using Mohr-Coulomb Plasticity Model(blue) and Drucker-Prager Plasticity Model(red)

3.3.2 2D Axi-Symmetric vs 2D Planar Analysis

Fig. 3.13 shows the Depth of penetration vs time for 2D Axi-Symmetric Analysis(red) and 2D Plane Strain Analysis(blue) using Drucker-Prager Plasticity Model. It can be seen that the maximum depth of penetration in plane strain analysis is lesser than the axi-symmetric analysis, this is because the plain strain analysis considers the spear as *wedge* whereas the axi-symmetric analysis considers spear as cone. The surface of Wedge is greater than cone given apex angle and length(or diameter), as the surface area increases the drag force increases as a result the depth of penetration decreases. And also the difference of depth of penetration between both the analyses is huge(greater than 24%), so Two Dimensional Plane Strain analysis cannot replace Two Dimensional Axi-Symmetric Analysis to perform inclined spear penetration analyses.

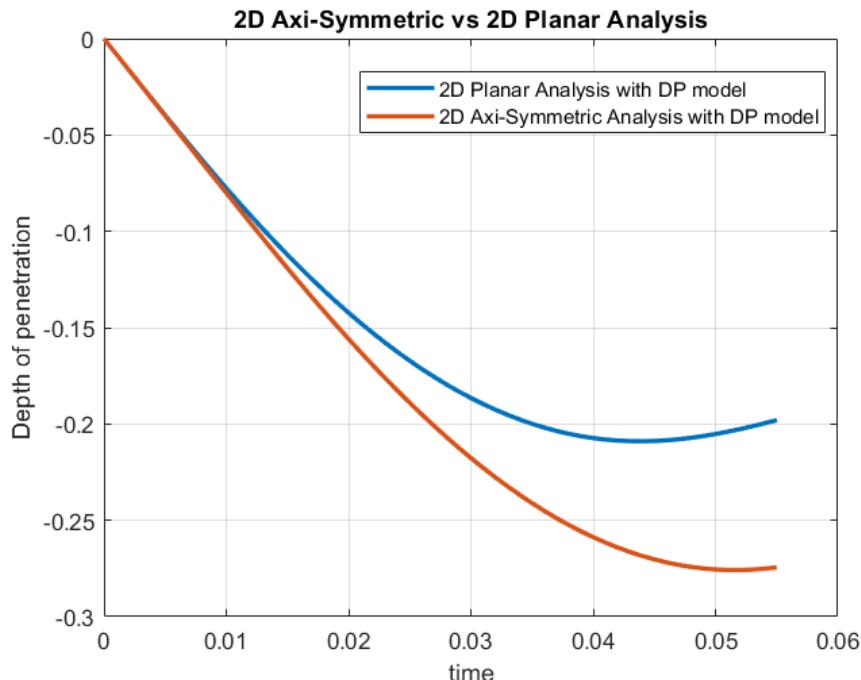


Figure 3.13: Depth of penetration vs time for 2D Axi-Symmetric Analysis(red) and 2D Plane Strain Analysis(blue) using Drucker-Prager Plasticity Model

3.4 Three Dimensional Analysis

In real life scenario, the reusable stage might have non-zero body rate and might be descending with pitch angle other than 90° . And also there is a chance that optimal landing

can occur with non-zero body rates and inclined pitch angles. Since the analysis with body rates and inclined pitch angles cannot be done using Axi-Symmetric and Planar analyses, Three Dimensional Analysis is inevitable. The same problem of mesh distortion arises in Three-Dimensional Analysis because of large deformation in small amount of time. Using ALE Auto-Adaptive Remeshing in three dimensions increases the computational time exponentially. In ALE, the material is independent of nodes inside the domain which is eulerian approach but fixed to nodes on boundaries which is lagrangian approach, instead one can think of modelling the entire domain using eulerian approach in which material is independent of nodes all over the domain, that is what the Coupled Eulerian Lagrangian (CEL) technique in ABAQUS offers, and it is briefly explored in the sections following.

3.4.1 Coupled Eulerian Lagrangian(CEL) Method

The Coupled Eulerian Lagrangian(CEL) analysis use both eulerian and lagrangian approaches in the same analysis. A lagrangian body is one that is more rigid than the others, while the others are eulerian bodies in the analysis. Unlike, lagrangian bodies where the elements and nodes are fixed to the material, the elements of the eulerian body(s) are not fixed to the material but are fixed in space and the material moves between the elements(as shown in Fig. 3.14).

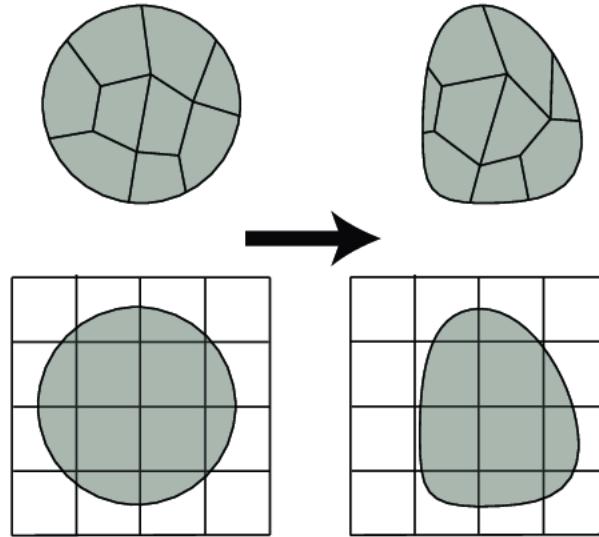


Figure 3.14: Deformation in Lagrangian Analysis(top) and Eulerian Analysis(bottom)[6]

Since the elements are fixed in space, there is no mesh distortion in CEL approach.

A parameter called Eulerian Volume Factor(EVF) whose value varies from 0 to 1 is introduced in the CEL approach to monitor the material flow in side the elements, it takes value 0 when there is no material inside the element and 1 when the element is filled with material. EVF plays a crucial role in numerical integration. Generally, the material tends to flow out of the domain in CEL analysis, so a void space is created above the domain to account for the spillage and void is meshed along with domain but material is not assigned to it. CEL approach supports only explicit integration scheme and detailed mathematical framework of CEL can be found in [22].

3.4.2 Validation of CEL approach to cone penetration analysis

CEL approach is gaining popularity since its inception in geotechnical community, almost all geotechnical field tests are being simulated using this approach. Various authors simulated standard field tests like cone penetration test(CPT), Freely Falling Plate Penetrometer, etc. But literature is not available for *freely falling cone penetrometer* test which is very similar to our current work. A validation study have been done to validate CEL approach to freely falling cone penetrometer test and the obtained results are compared to Nazem et. al[9] standard results. Nazem et. al used 2D Axi-symmetric analysis with ALE auto-adaptive remeshing for simulating freely falling cone penetrometer into frictionless clay, the same analysis is done in 3D using CEL approach(shown in Fig. 3.15), the Plasticity model used here is Tresca Plasticity model($\tau = s_u$) with strain hardening effect.

$$s_u = s_{u,ref} \left[1 + \lambda \log \left(\frac{\dot{\gamma}}{\dot{\gamma}_{ref}} \right) \right] \quad (3.17)$$

The experimental conditions are shown in Table. 3.4

Table 3.4: Experimental Conditions for validation study[9]

Parameter	Value
Diameter of the Spear	0.04m
Apex Angle	60°
Impact Velocity	10 m/s
Mass	0.2 Kg
$s_{u,ref}$	5kPa
$E_u/s_{u,ref}$	200
λ	0.2
$\dot{\gamma}_{ref}$	0.01/hour

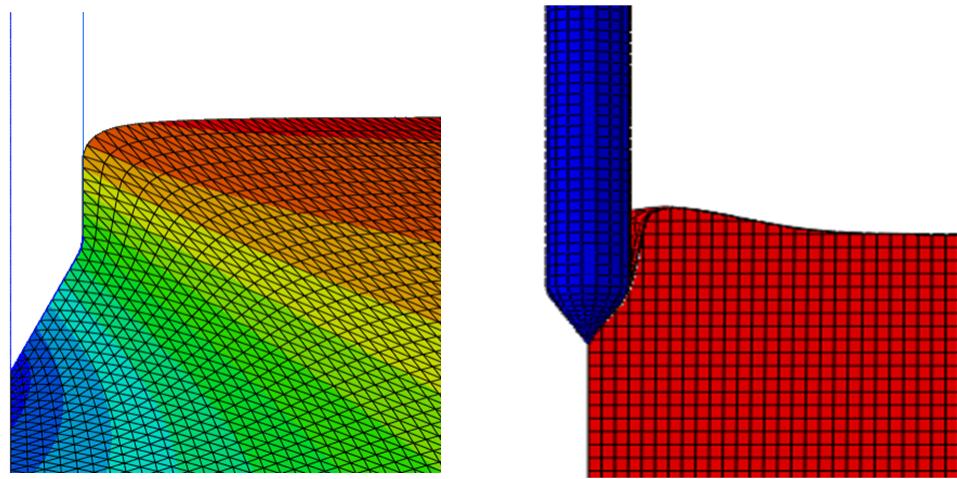


Figure 3.15: Validation of Freely Falling Penetrometer using 2D Axi-symmetric(left) and 3D CEL analyses(right)

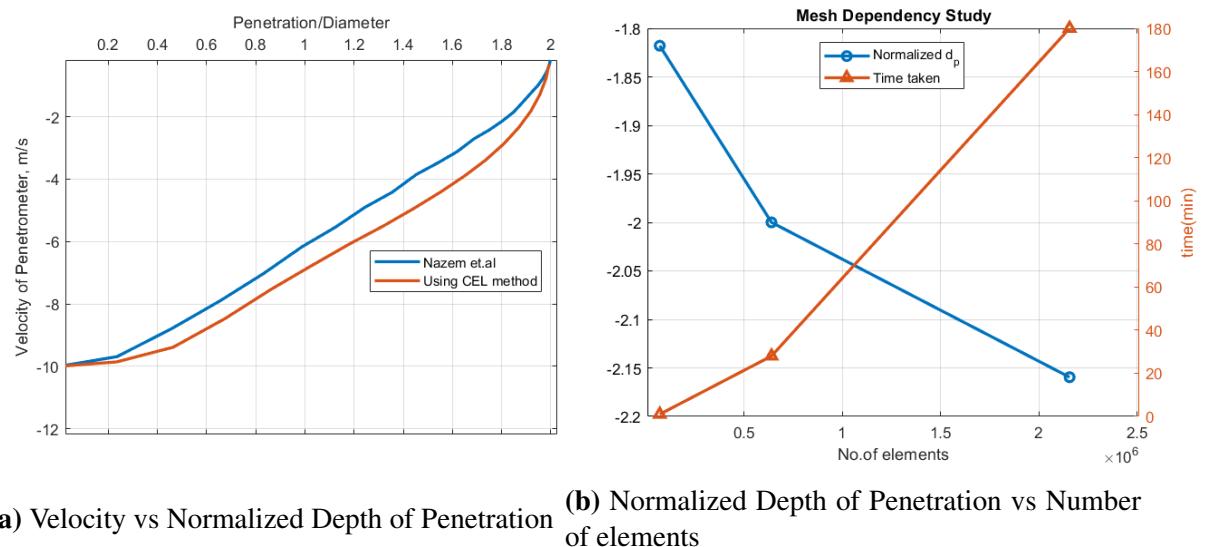


Figure 3.16: Validation Study and Mesh Dependency Study

Fig. 3.16(left) shows the Depth of Penetration vs Velocity plots of freely falling cone penetration test into clay using 3D Coupled Eulerian Lagrangian(CEL) method and 2D ALE method(Nazem et. al), although there is slight difference between the intermediate values, the final depth of penetration obtained from the both the methods are equal and this proves the applicability of 3D CEL method to the freely falling penetration problems. Mesh Dependency Study have been done to the 3D CEL method and is shown in Fig. 3.16(right). It can be seen that as the number of elements increases the depth of penetration is increasing, the reason is not known clearly but one possibility is that the accuracy of contact interface between lagrangian body and eulerian bodies is not as great as lagrangian body with lagrangian body because in CEL contact interface is determined using Eulerian Volume Fraction(EVF) in each element[17]. This is the main disadvantage of Coupled Eulerian Lagrangian method where element size is also a parameter one needs to find out.

3.4.3 Spear In Sand Penetration Analysis using Coupled Eulerian Lagrangian(CEL) Analysis

As discussed earlier, element size is also a parameter in CEL analysis. First the size of the element required for the current analysis is found out by reproducing the Depth of penetration vs Time plot from 2D Axi-Symmetric Analysis using CEL method and later the obtained element size is used for the further analyses. The methodology used in the 3D CEL method is same as the 2D Axi-Symmetric Analysis but the contact in 3D CEL is formulated as the *General Contact*. The best mesh seed size obtained is 0.157 and the number of elements used are 1,48,257 elements. The non-dimensional depth of penetration vs time are shown in Fig. 3.17. It can be seen that both curves are very similar with slight difference(error < 2%) in the final displacement, which one can safely assume as negligible.

3.4.4 Finite Element Model of Reusable Stage Recovery in Sand for 3D CEL method

Since it turned out that using 3D analysis is inevitable, the full reusable stage is considered for the analysis and the properties of the reusable stage are listed in Table. 3.5 and the reusable stage is shown in Fig. 3.18.

The reusable stage with landing legs is modelled as Discrete Rigid Body in the analysis and the forces, stresses acting on the landing legs of stage are found out using FEA analysis.

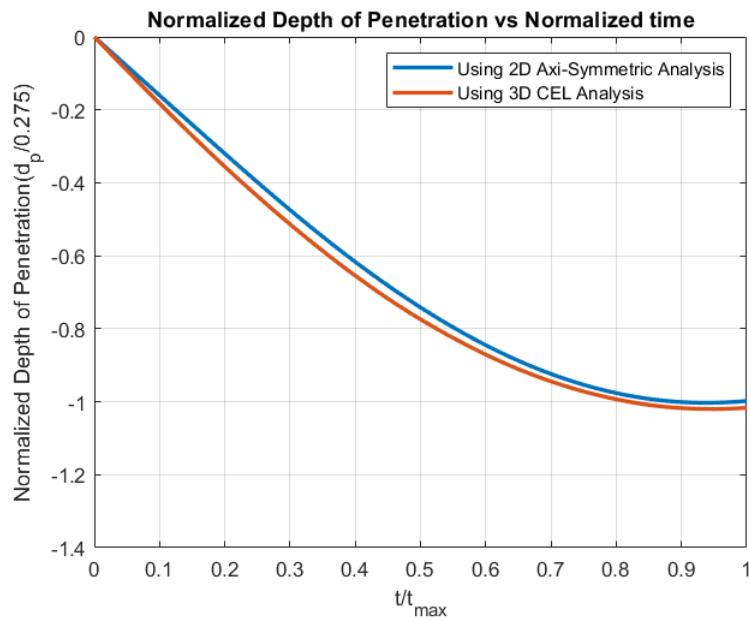


Figure 3.17: Normalized Depth of Penetration vs Normalized time

Table 3.5: Properties of Reusable Stage(courtesy of VSSC)

Parameter	Value
Mass	8350 Kg
Diameter	2.1 m
Length	16.23 m
CG position along axis from bottom(y_{CG})	6.678m
x_{CG}	0.04m
z_{CG}	0.02m
$I_{yy,CG}$	8000 Kgm ²
$I_{xx,CG} = I_{zz,CG}$	400000 kgm ²
$I_{xy,CG} = I_{yz,CG}$	1000 kgm ²
$I_{xz,CG}$	100 Kgm ²
Nozzle Length	1.15 m
Nozzle Diameter	1 m

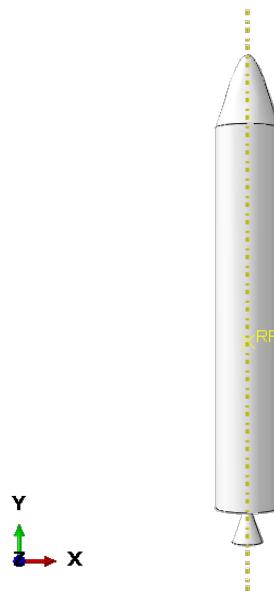


Figure 3.18: Reusable Stage without Landing Legs

Only landing legs are in contact with sand throughout the analysis, therefore to reduce the computational expenditure the reusable stage is modelled as point mass with all the above properties and only legs are modelled as 3D Rigid Bodies as shown in Fig. 3.19.

The size of the domain is taken as 8mx8mx8m and on top of it 8mx1mx8m space is given as void space. *Explicit* method is used as solution method and the type of analysis used is *Coupled Eulerian Lagrangian* Analysis and the type of elements used for modelling sand(Eulerian Body) and Landing Legs(Discrete Rigid Body) are *Eight-Noded Linear Eulerian Brick(EC3D8R)* and *Four-Noded 3-D Bilinear Rigid Quadrilateral(R3D4)*. Three-Dimensional CEL Finite Element Model with boundary conditions, fields and mesh are shown in Fig. 3.20.

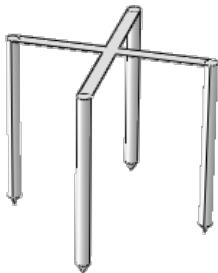


Figure 3.19: Landing Legs with CG position

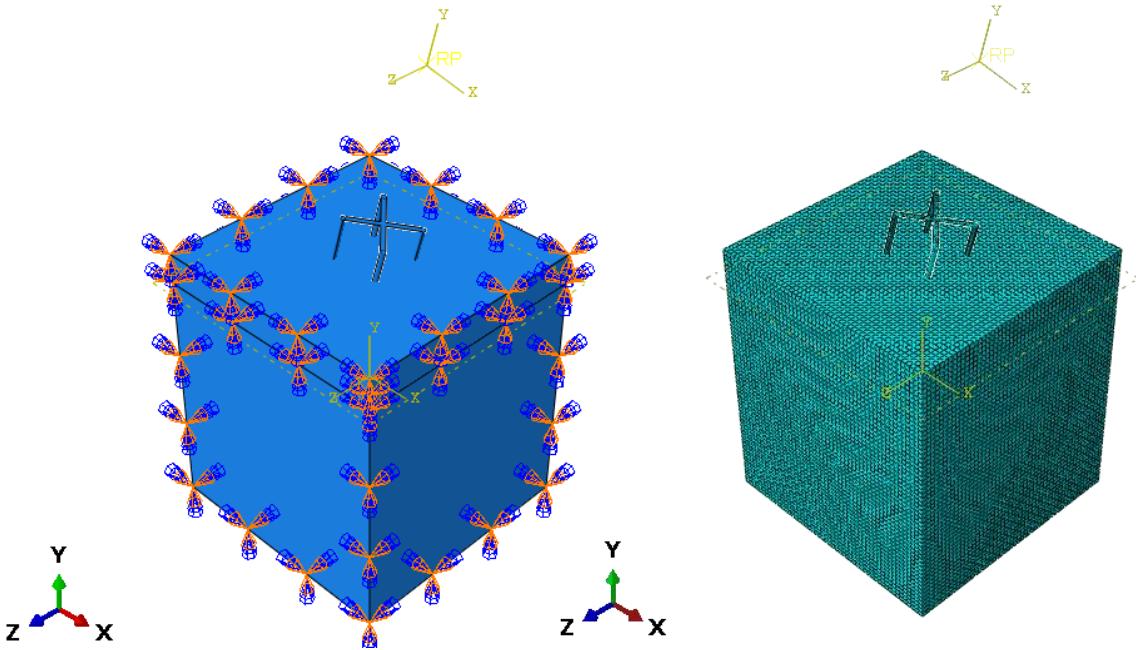


Figure 3.20: 3D CEL Finite Element Model and 3D meshing

In the current work, only four legs are used and the stage is assumed to be landing on two legs initially when the pitch angle is not equal to 90° (as shown in the above figure). Although the analysis is three dimensional, the forces and fields are defined only in XY plane. With all the parameters fixed, the current analysis is dependent on three variables, magnitude of impact velocity(V), Pitch Angle(θ) and Pitch Rate($\dot{\theta}$) and gives the depth of penetration vs time, acceleration, reaction forces, etc which are used for design and optimization. Now it is similar to black box simulation or an unknown function, where it takes the input and gives the output. Directly using the above **ABAQUS** model in design optimization is not possible because, for e.g., if Particle Swarm Optimization(PSO) with population size as 100 and maximum number of iterations of 100 is used for design optimization, the total number of required simulations are more than 10000 and also dynamical, manually doing these simulations is nearly impossible. So, a metamodel or surrogate model is used for the design optimization and is discussed in the next chapter.

Chapter 4

Metamodel Based Design Optimization

4.1 Introduction

Due to their potential to accurately simulate real-world problems, computer-based numerical studies like as Computational Fluid Dynamics (CFD), Finite Element Analysis (FEA), and others have become a very essential part in aerospace design optimization. However, they comes with the huge computational cost. To overcome the computational cost and to make optimal design within less time, *Metamodels* also known as *Surrogate Models*, *Response Surface Models*, etc are introduced. Metamodels are the analytical representation or analytical expressions for the actual computational models constructed based on the dataset of inputs and their corresponding outputs obtained from computational models. In other words, Metamodel is a fitting function which takes the input and returns the output which is nearly equal or equal to the output of actual computational model. The concept of metamodeling is shown in the Fig. 4.1 Metamodelling consists of three important steps:

1. Design of Experiments(or Sampling): A set of input points is created which spread over the whole domain in this step.
2. Functional Evaluations(or outputs): In this step, the actual outputs for inputs obtained in teh above step are obtained from computational models.
3. Metamodel: In this step, a function is fitted for above generated inputs and outputs.

The fitted model can be used for further processes like Optimization(hence the name Metamodel Based Design Optimization), Uncertainty Analysis, Sensitivity Analysis, etc.

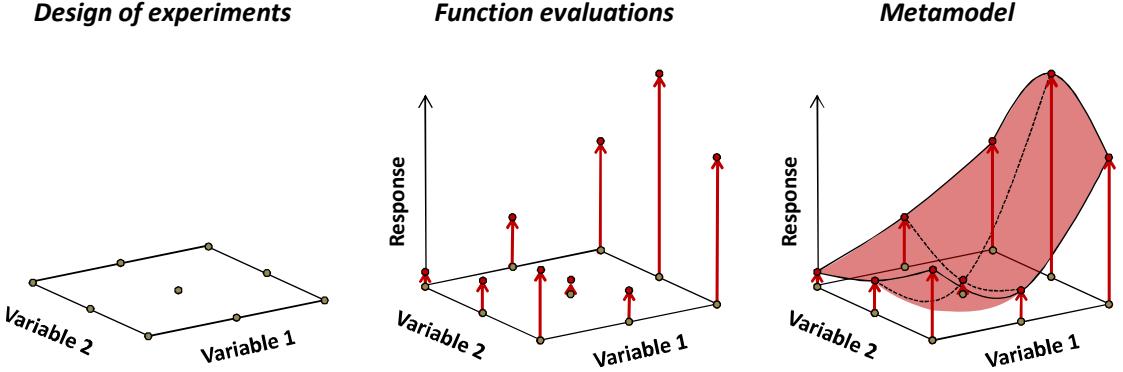


Figure 4.1: The concept of metamodeling for the case of two design variables[7]

4.2 Metamodels

As discussed in Chapter. 4, the final or detailed computational model takes the input and gives the output, mathematically, it takes a set of input variables(\mathbf{x}) and returns a set of output values(\mathbf{y}), one can think of the computational model as function, f , which maps the m real input variables to n real output variables,

$$\mathbf{f} : \mathbb{R}^p \rightarrow \mathbb{R}^q \quad (4.1)$$

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \quad (4.2)$$

But the function, f , is unknown which is similar to blackbox simulation. The metamodel assumes a known, analytical function, $s(\mathbf{x})$ for f using finite dataset $\{\mathbf{x}, \mathbf{y}\}$ which approximates the true function values as,

$$\hat{\mathbf{y}} = s(\mathbf{x}) \quad (4.3)$$

In general the predicted function values are not as same as true function values and contains some error, ε

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \hat{\mathbf{y}} + \varepsilon = s(\mathbf{x}) + \varepsilon \quad (4.4)$$

There are many standard metamodels like, *Polynomial Regression*, *kriging model*, *Radial Basis Function Neural Network(RBFNN)*, *Artificial Neural Networks(ANN)*, *Support Vector Regression(SVR)*, etc. Each of the metamodels have their own benefits and applicability. In the current work, **Radial Basis Function Neural Network(RBFNN)** is chosen as

metamodel.

4.2.1 Radial Basis Function Neural Network(RBFNN)

Hardy et. al[23] created the Radial Basis Function Neural Network (RBFNN) metamodel to fit the uneven topography contours of geographical data. RBFNN consists of three layers, *input layer*, *hidden layer*, and *output layer*. Unlike Artificial Neural Networks, in which non-linearity is achieved by using more hidden layers with linear basis functions, RBFNN uses only one hidden layer with non-linear basis functions. Given a set of N sampling points and their outputs, $[x_i, y_i]$ ($i=1,2,\dots,N$), where x_i being the p-dimensional vector and \hat{y}_i being its q-dimensional output vector, the standard RBFNN model can be written as,

$$\hat{y}_j = \sum_{i=1}^n \lambda_{ij} \Phi(||x - c_i||) = \Phi \cdot \lambda_j, \quad (j = 1, 2..q) \quad (4.5)$$

The Φ is radial basis function, λ_{ij} are the weights, n represents the total number of neurons in the hidden layer and c_i represents the centers of radial basis functions, and is of same dimension as input, the weights and centers of RBFNN can be obtained by training using the sample points also known as training of neural network. There are many types of radial basis functions are available and some of them are shown in Table. 4.1

Table 4.1: Radial Basis Functions

Radial Basis Function	Expression
Linear	$\Phi(r) = r$
Cubic	$\Phi(r) = (r + c)^3$
Thin-Plate Spline	$\Phi(r) = r^2 \log(cr)$
Multi-Quadric	$\Phi(r) = \sqrt{r^2 + c^2}$
Inverse Multi-Quadric	$\Phi(r) = 1/\sqrt{r^2 + c^2}$
Gaussian	$\Phi(r) = e^{-cr^2}$

where $r = ||x - c_i||$ and c is a constant from (0,1].

4.2.2 RBFNN Training

There are various methods for obtaining centers and weights, an analytical method is chosen for the current work, in which the centers are same as the sample data inputs ($c_i = x_i$) and the number of neurons is equal to the number of sampling points($n=N$), then using

equation(5.5),

$$\mathbf{y}_{N \times q} = \Phi_{N \times N} \boldsymbol{\lambda}_{N \times q} \quad (4.6)$$

where,

$$\mathbf{y}_{N \times q} = \begin{bmatrix} y_{1,1} & y_{1,2} & \dots & y_{1,q} \\ y_{2,1} & y_{2,2} & \dots & y_{2,q} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ y_{N,1} & y_{N,2} & \dots & y_{N,q} \end{bmatrix}$$

$$\Phi_{N \times N} = \begin{bmatrix} \Phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \Phi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \dots & \Phi(\|\mathbf{x}_1 - \mathbf{x}_N\|) \\ \Phi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \Phi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \dots & \Phi(\|\mathbf{x}_2 - \mathbf{x}_N\|) \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \Phi(\|\mathbf{x}_N - \mathbf{x}_1\|) & \Phi(\|\mathbf{x}_N - \mathbf{x}_2\|) & \dots & \Phi(\|\mathbf{x}_N - \mathbf{x}_N\|) \end{bmatrix}$$

$$\boldsymbol{\lambda}_{N \times q} = \begin{bmatrix} \lambda_{1,1} & \lambda_{1,2} & \dots & \lambda_{1,q} \\ \lambda_{2,1} & \lambda_{2,2} & \dots & \lambda_{2,q} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \lambda_{N,1} & \lambda_{N,2} & \dots & \lambda_{N,q} \end{bmatrix}$$

For a chosen radial basis function(Φ) and c , the only unknowns in the above equation is $\boldsymbol{\lambda}_{N \times q}$ which can be obtained by inverting the radial basis function matrix($\Phi_{N \times N}$) and multiplying with the sample output points, i.e.,

$$\boldsymbol{\lambda}_{N \times q} = \Phi_{N \times N}^{-1} \mathbf{y}_{N \times q} \quad (4.7)$$

The obtained weights completes the training of metamodel.

4.3 Design of Experiment(DoE)

The accuracy of the metamodel depends on two factors: (1) Selection of set of input points(Sampling), (2) Type of Fitting Function. The process of selection of finite number of input points(Sampling) over the design space in order to get the best information of the model is known as **Design of Experiments(DoE)**. There are many methods for sampling the design space, but all the methods fall under three categories as shown in Fig. 4.2.

The sampling techniques are divided into two major categories, known as *One-shot sam-*

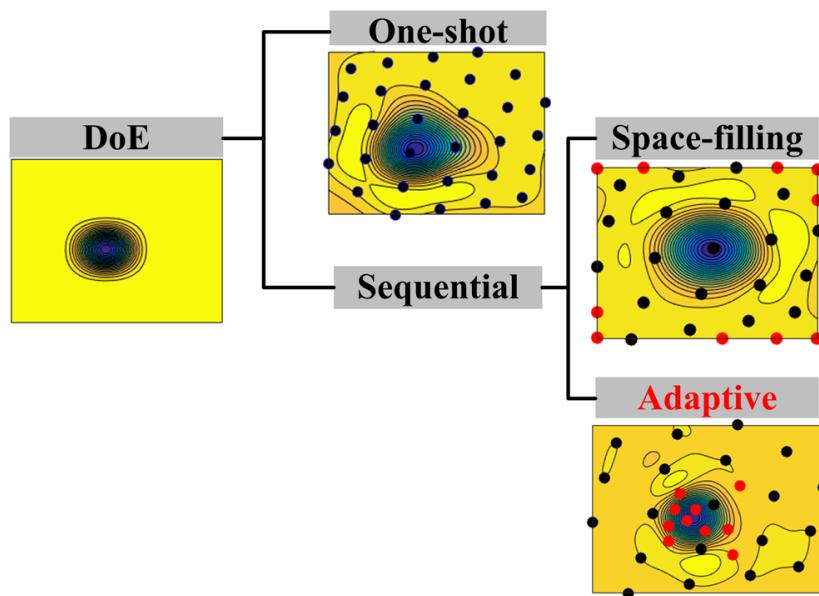


Figure 4.2: Various Sampling Categories[8]

pling and *Sequential Sampling*. In *One-shot sampling*, the sample size is known beforehand and the sampling is done at a single step. The key drawback of this strategy is that it is difficult to discover the ideal sample size for a function with no prior knowledge. To mitigate this problem *Sequential Sampling* is proposed, in which sampling is done sequentially. Sequential Sampling again subdivided into two categories, *Space-filling* and *Adaptive Sequential Sampling*. *Space-filling Sequential Sampling* uniformly distributes sample points across the full design space, whereas, *Adaptive Sequential Sampling* chooses the sample points based on the active learning from the previous sample points and metamodel generated from them.

As said earlier there are many methods for sampling, some of the methods used in the current work are described below.

4.3.1 Grid Sampling

Grid Sampling, comes under the one-shot sampling method, is the simplest sampling method in which the domain is partitioned into uniform square grids and the grid points(or nodes) are taken as sampling points as shown in the Fig. 4.3.

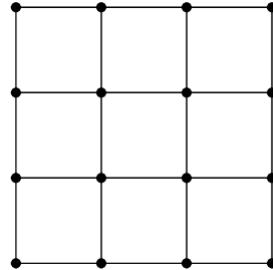


Figure 4.3: Two Dimensional Grid Sampling

4.3.2 Latin Hypercube Sampling

Unlike Grid Sampling, Latin Hypercube Sampling is a combination of random and stratified sampling in which given the number of sample points required(n), the interval in each dimension divided into n sub interval and a random value is chosen from each sub interval using uniform distribution and the obtained values in each interval each dimensions are randomly paired with values from other dimensions without replacement to form a sample set. The important point to note here is that if the domain is divided into n^2 boxes LHS places the sample points inside the boxes such a way that the next sampling point is not placed same column and row in which previous point is placed as shown in Fig. 4.4.

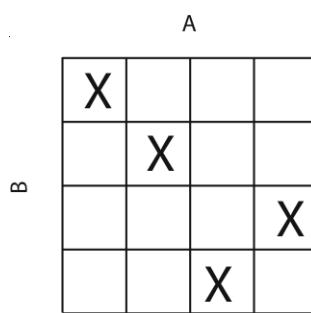


Figure 4.4: Latin Hypercube Sampling

4.3.3 Adaptive Sampling

Computational Models in Aerospace Design Optimization consumes huge amount of time (especially CFD simulations takes days for completion). *Adaptive Sampling* is designed for such cases where functional evaluations are very expensive to obtain. As said earlier, it uses active learning from previous sample points and metamodel to obtain new points such that an accurate metamodel is constructed with least number of points possible. The general adaptive sampling approach is shown in Fig. 4.5. The process of adaptive sampling starts

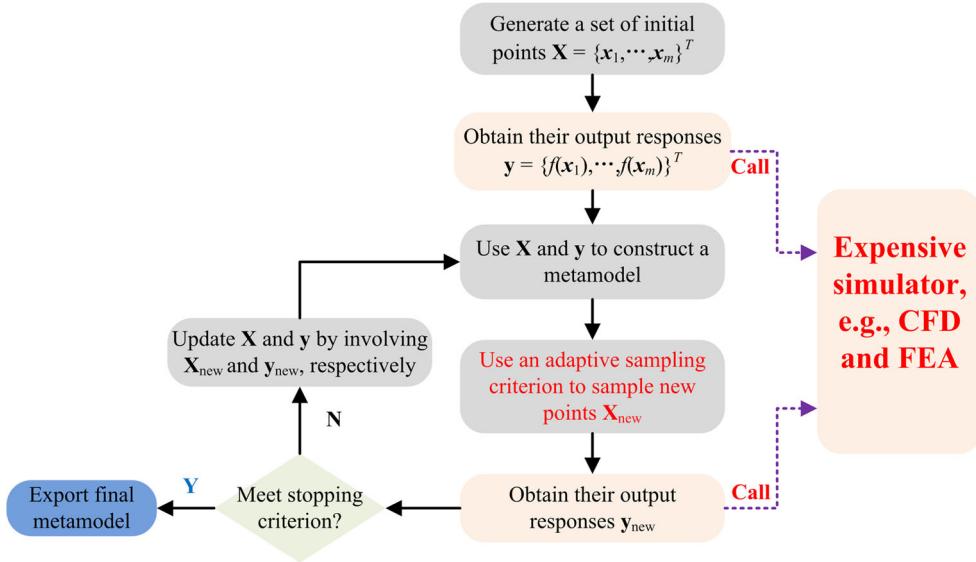


Figure 4.5: General flowchart of an adaptive sampling approach[8]

with a small number of sample points, obtained using grid or latin hypercube sampling, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}^T$ and their corresponding output responses from the computational models $\mathbf{y} = \{y_1, y_2, \dots, y_m\}^T$ to build an initial metamodel \hat{f} . Then using the adaptive sampling criterion, an additional global level optimization is performed over the design space to obtain \mathbf{x}_{new} and it is added to \mathbf{X} till the stopping criterion is met. The new point x_{new} is selected by maximizing the score function over the entire domain as[8],

$$x_{new} = \operatorname{argmax}_{x \in D} Score(local(x), global(x)) \quad (4.8)$$

where the $local(x)$ and $global(x)$ represent the local exploitation and global exploration terms. The local exploitation term plays a major role in adaptive sampling since it helps to find the points that give maximum information about the computational model. We can use prediction variance, cross-validation error, local non-linearity, local gradient and hes-

sian(curvature) of the initial metamodel \hat{f} to create local exploitation criterion.

There is a chance of piling up of new sample points at particular region if we use only local exploration criterion for finding the new points. The global exploration term helps to discover the new points that have not been discovered before. In this way we can explore the entire design space to find the points which gives maximum information about the computational models. Generally, distance based criterion *maximin* or *minimax* are used for global exploration. **Gradient Based Adaptive Sampling** is used in the current work due to the analytical tractability of local gradients and hessians of RBF Neural Networks.

4.3.3.1 Gradient Based Adaptive Sampling

Graphically, the function values changes drastically near the regions with large curvature(or large gradients) and these regions are most important for optimization. It is hard for metamodel generated using one-shot sampling methods to capture the curvature data exactly which results in large prediction errors. **Gradient Based Adaptive Sampling** make use of gradients(or curvature or geometry) of the metamodel to sample more points at the regions with large gradients(or curvature) to get more accurate metamodel with less number of points. In the current work, **Maximum Curvature and Minimum Point Distance based Sequential Sampling Method(MCMPDS)**[19] is employed for sampling new points and is described below.

4.3.3.1.1 Maximum Curvature and Minimum Point Distance based Sequential Sampling Method(MCMPDS)

Before getting into the details, it is important to get familiarize with the variables and notation of the method,

1. **Gradient of RBFNN:** Considering the single valued RBF Neural Network with 'p' dimensions, the gradient of the RBF Neural Network can be represented as follows,

$$\frac{\partial \hat{f}}{\partial \mathbf{x}} = \sum_{i=1}^N \frac{\lambda_i \phi'(r_i)}{r_i(\mathbf{x})} (\mathbf{x} - \mathbf{x}_i)^T \quad (4.9)$$

Where $r_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i\|$, $\phi'(r_i)$ is the partial derivative of $\phi(r_i)$ with respect to r_i [19]

2. **Hessian of RBFNN:** Again differentiating the above equation gives the Hessian of

the RBF Neural Network,

$$H(\mathbf{x}) = \frac{\partial^2 \hat{f}}{\partial \mathbf{x}^2} = \sum_{i=1}^N \frac{\lambda_i}{r_i(\mathbf{x})} \left(\phi'(r_i) \cdot \mathbf{I} + \left[\phi''(r_i) - \frac{\phi'(r_i)}{r_i(\mathbf{x})} \right] \frac{(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^T}{r_i(\mathbf{x})} \right) \quad (4.10)$$

3. **Curvature($\mathbf{K}(\mathbf{x})$):** For multi dimensional problem the curvature $\mathbf{K}(\mathbf{x})$ is defined as the square root of sum of squares of eigen values of the hessian matrix, for a RBF Neuaral Network with dimension 'p', $\mathbf{K}(\mathbf{x})$ can be written as,

$$\mathbf{K}(\mathbf{x}) = \sqrt{\sum_{i=1}^p k_i(\mathbf{x})^2} \quad (4.11)$$

4. **Minimum Distance($d_{min}(\mathbf{x})$):** It is the minimum distance between the point \mathbf{x} and rest of the sample points \mathbf{x}_i [19].

4.3.3.1.2 MCMPDS Algorithm : With the above knowledge, the MCMPDS method can be described as follows,

- **Step-1:** Initialize a small set of points($\{\mathbf{X}\}$) using one-shot sampling methods like grid sampling, Latin Hypercube Sampling, etc.
- **Step-2:** Find the outputs $\{\mathbf{y}\}$ for the initialized set of points using the computational points.
- **Step-3:** Fit the initial RBFNN metamodel for the inputs $\{\mathbf{X}\}$ and the outputs $\{\mathbf{y}\}$.
- **Step-4:** Check the stopping criteria. The post analysis methods like RSME need large number of points which is not suitable for our case. In the current work, two stopping criteria are selected,
 1. Maximum number of functional evaluations(M_1), and
 2. Maximum number of continuous *Invalid Sampling Points*(ISP)(M_2). The former criteria constrains the maximum number of functional evaluations and the later criteria ensures that the model is sufficiently accurate.

- The continuous invalid sampling points are the subsequent sampling points that meet the condition[19],

$$|y_i - \hat{y}_{i-1}| \leq \delta_a \text{ or } \frac{|y_i - \hat{y}_{i-1}|}{\bar{y}_i} \leq \delta_r$$

Where y_i is the output from the computational model for input \mathbf{x}_i , \hat{y}_{i-1} is the output from RBF Neural Network build without the sampling point \mathbf{x}_i , \bar{y}_i is the average of all existing responses, δ_a and δ_r are absolute and relative tolerances. If any one is satisfying then stop the iterations.

- Step-5:** Sequential Optimal Sampling. The new optimal point(\mathbf{X}_{new}) is obtained by solving the global optimization problem[19],

$$\begin{aligned} & \text{Maximize} \quad K(\mathbf{x}) \cdot d_{min}^D(\mathbf{x}) \\ & \text{W.r.t} \quad \mathbf{x} \\ & \text{Subject to} \quad K(\mathbf{x}) \geq 10^{-8} \end{aligned}$$

Where $K(\mathbf{x})$ is the curvature(local exploitation term) and $d_{min}^D(\mathbf{x})$ is the minimum distance points(global exploration term). where D is the exponent and is taken as D=2 for 2-dimensional domain and D=1 for 3-dimensional domain in the current work. Here maximizing the product $K(\mathbf{x}) \cdot d_{min}^D(\mathbf{x})$ ensures that the new sampling points are at maximum curvature and also spreading the entire domain. In the current work *Differential Evolution* is used for solving the above global optimization problem.

- Step-6:** Compute the output for the new point from the previous step and update the RBF Neural Network.
- Step-7:** Go to Step-3 for checking stopping criterion. The above algorithm is taken from the Ref. [19].

4.4 Metamodel Validation

As said earlier, the accuracy of the metamodel depends on sampling and type of metamodel and there are no universal measures for describing the goodness of the metamodel. But there are measures used to assess the accuracy of the metamodels by comparing them with others. Error based measures and Cross Validation measures and Invalid Sampling Points(ISP's) are few of the metrics used in the current work.

4.4.1 Error Measures

In error measures, the accuracy of the metamodel is evaluated by the error or residuals, the difference between functional value from computational models and metamodels.

4.4.1.1 Root Mean Square Error(RSME)

RSME is most widely used metric for calculating accuracy, but it requires huge data set for getting estimate of error.

$$RSME = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \quad (4.12)$$

where y_i is the output from computational model and \hat{y}_i is the output from the computational model.

4.4.2 Cross Validation

Cross Validation methods are mainly used when output from the computational models are very expensive to obtain. Instead of creating extra dataset for testing(or assigning some portion of training dataset for testing), they make use of only training data to validate the metamodel. Cross Validation error is computed by splitting the training dataset(outputs and inputs) into 'n' subsets randomly and then in each turn a subset is taken out and a metamodel is fitted to the remaining data and validated on chosen dataset. The procedure continues for all subsets and mean square error of all the subsets gives the Cross Validation Error.

4.4.2.1 Leave-One-Out Cross Validation

Leave-One-Out Cross Validation is a cross validation method in which the subset reduces to one point. This method is used when training dataset is usually smaller and output responses are very expensive to obtain. The cross validation error for training dataset of size n can be written as,

$$\epsilon_{CV} = \frac{1}{n} \sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2 \quad (4.13)$$

Where $y^{(i)}$ is the actual output from the computational model for the input x_i and $\hat{y}^{(i)}$ is the output obtained for the metamodel fitted using training data without the point x_i . If 'y' is vector valued function and the difference between order of magnitude of components is huge the cross validation error can be modified as

$$\epsilon_{CV} = \frac{1}{n} \sum_{i=1}^n \left(\frac{\mathbf{y}^{(i)}}{|\mathbf{y}^{(i)}|} - \frac{\hat{\mathbf{y}}^{(i)}}{|\hat{\mathbf{y}}^{(i)}|} \right)^2 \quad (4.14)$$

4.5 Invalid Sampling Points

Invalid Sampling Points (ISP's) are points that can be eliminated without affecting the metamodel's accuracy. As a result, the more invalid sample points in the training dataset, the more invalid output responses from the computational models, wasting computational time. As a result, the greater the number of invalid sampling points, the worse the metamodel's performance. It is a post design method, we cannot eliminate the ISP's in the real time. The Invalid Sampling Points(ISP's) can be calculated using the weights(λ_i) of the RBF Neural Network[19] as,

$$\bar{\lambda} = \frac{1}{N} \sum_{i=1}^N |\lambda_i| \quad (4.15)$$

the training point $[x_i, y_i]$ with weight λ_i is said to be an Invalid Sampling Point if,

$$|\lambda_i| \leq \bar{\lambda} * 5\% \quad (4.16)$$

4.6 Test Functions

The approach and methodology described above are applied to well-known benchmark functions. RBF Neural Network is used to approximate the functions, and three distinct sampling methods, *Grid Sampling*, *Latin Hypercube Sampling* and *Gradient Based Adaptive Sampling* are used. Invalid Sampling Points(ISP), Root Mean Square Error(RSME) and Leave-One-Out Cross Validation Error are chosen as the performance measures and are compared for three methods.

4.6.1 Single Valued/Objective Test Functions

4.6.1.1 Peaks Function

$$y = 3(1 - x_1^2).e^{-x_1^2 - (x_2+1)^2} - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right).e^{-x_1^2 - x_2^2} - \frac{1}{3}e^{-(x_1+1)^2 - x_2^2}, [x_1, x_2] \in [-3, 3; -3, 3] \quad (4.17)$$

The actual Peaks Function and its Contour are shown in Fig. 4.6, 4.7. Grid Sampling, Latin Hypercube Sampling and Gradient Based Adaptive Sampling are used to build the metamodel for the above function, the sampling points are shown in Fig. 4.8. Initial Sample for the Adaptive Sampling is generated using grid sampling with 3^2 points, the basis function used for the metamodel is *Inverse Multi-Quadric*, $M1=91(91+9=100)$, $M2 = 3$ and $\delta_r = 2\%$. The sampling methods performance is measured using Invalid Sampling Points(ISP's) and Root Mean Square Error(RMSE) and the results are shown in Table. 4.2.

Table 4.2: Comparsion of Sampling Techniques for Peaks Function

Measure	Grid Sampling	LHS	Adaptive Sampling
Functional Evaluations	100	100	83
ISP	33	17	12
RSME	0.0913	0.1392	0.0337

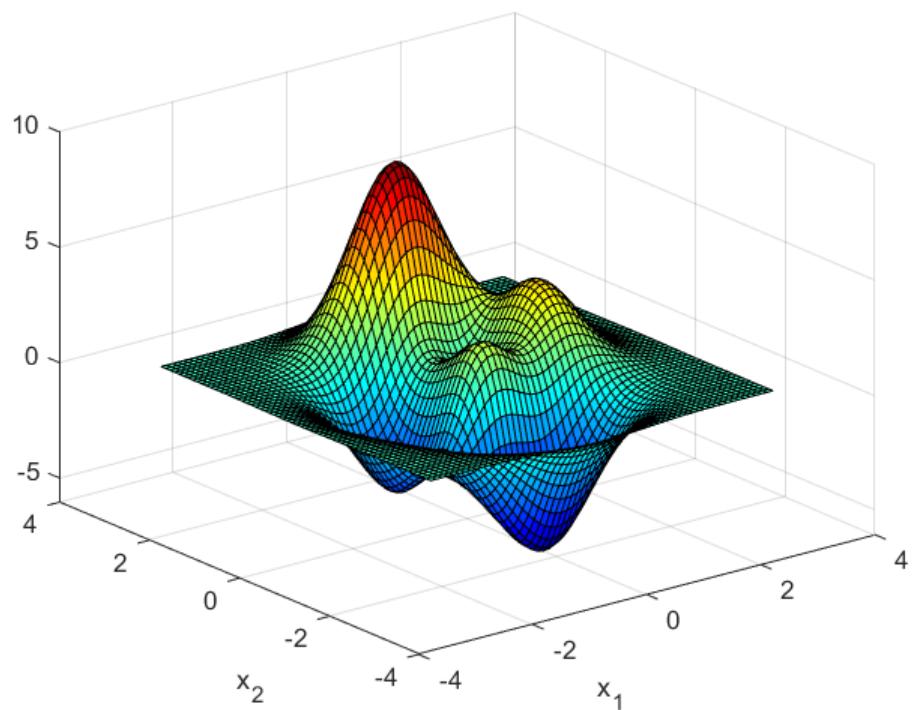


Figure 4.6: Peaks Function

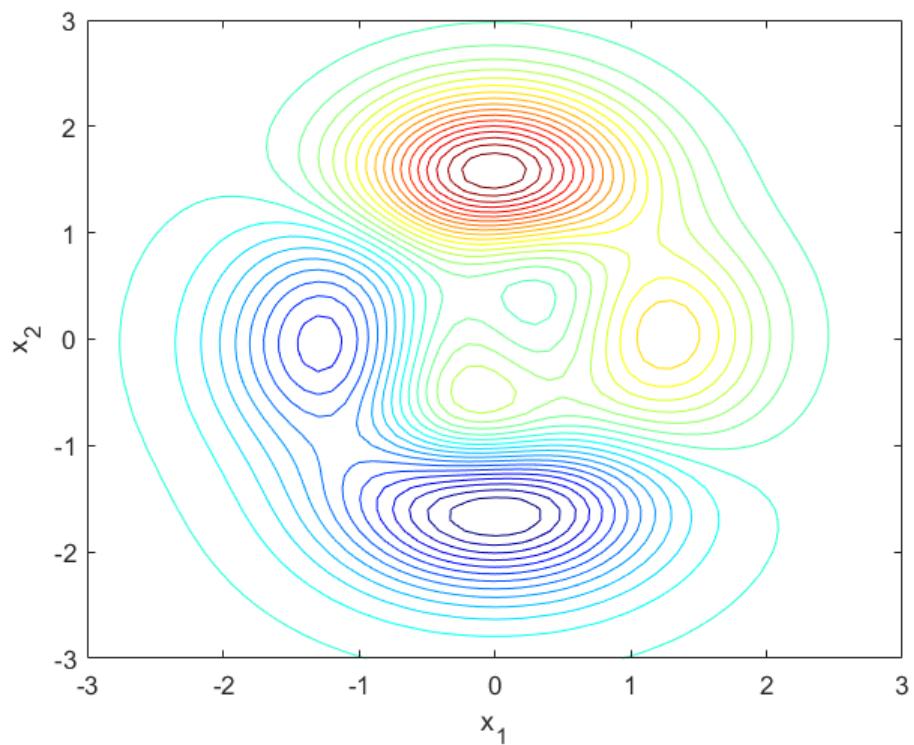


Figure 4.7: Contour of Peaks Function

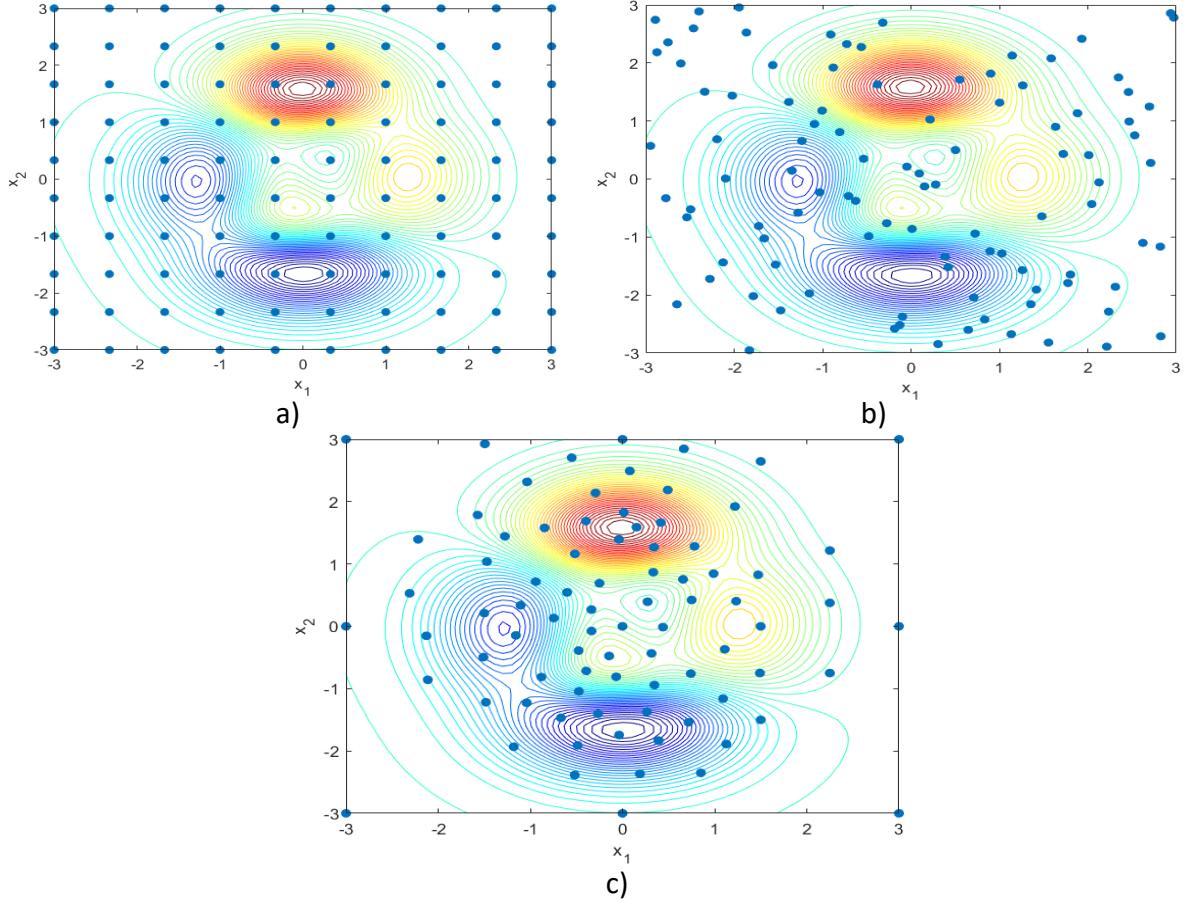


Figure 4.8: Sampling Points obtained from a) Grid Sampling, b) Latin Hypercube Sampling and c) Adaptive Sampling

4.6.1.2 Easom Function

$$y = -\cos x_1 \cos x_2 e^{-(x_1 - \pi)^2 - (x_2 - \pi)^2}, [x_1, x_2] \in [-10, 10; -10, 10] \quad (4.18)$$

The actual Easom Function and its Contour are shown in Fig. 4.6, 4.7. Grid Sampling, Latin Hypercube Sampling and Gradient Based Adaptive Sampling are used to build the metamodel for the above function, the sampling points are shown in Fig. 4.11. Initial Sample for the Adaptive Sampling is generated using grid sampling with 3^2 points, the basis function used for the metamodel is *Inverse Multi-Quadric*, $M1=55(55+9=64)$, $M2 = 3$ and $\delta_r = 2\%$. The sampling methods performance is measured using Invalid Sampling Points(ISP's) and Root Mean Square Error(RMSE) and the results are shown in Table. 4.3.

Table 4.3: Comparsion of Sampling Techniques for Peaks Function

Measure	Grid Sampling	LHS	Adaptive Sampling
Functional Evaluations	64	64	48
ISP	37	38	13
RSME	0.0538	0.0384	0.0028

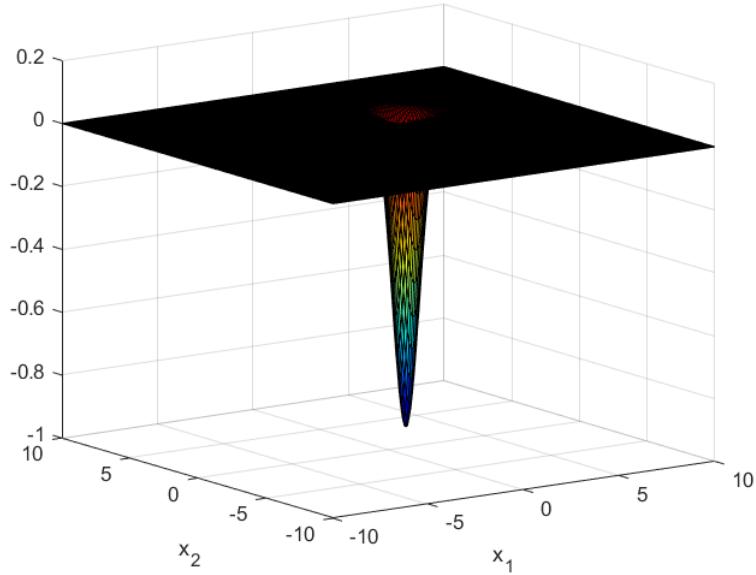


Figure 4.9: Easom Function

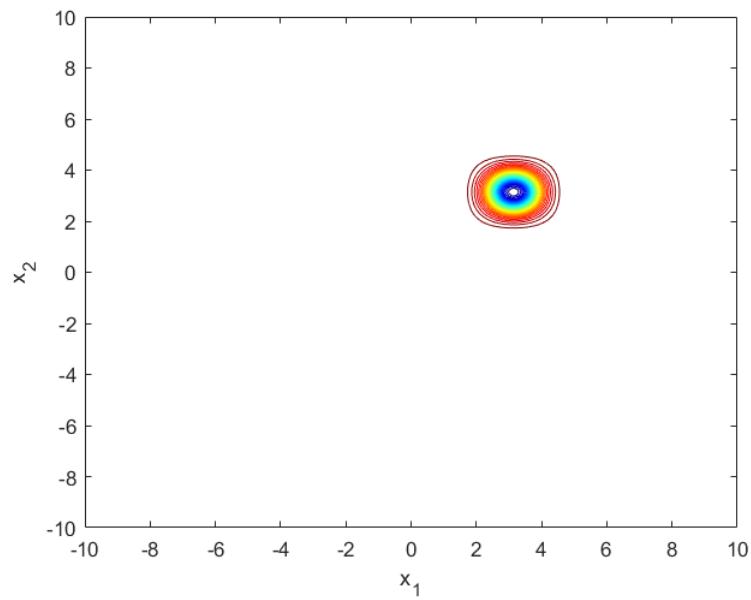


Figure 4.10: Contour of Easom Function

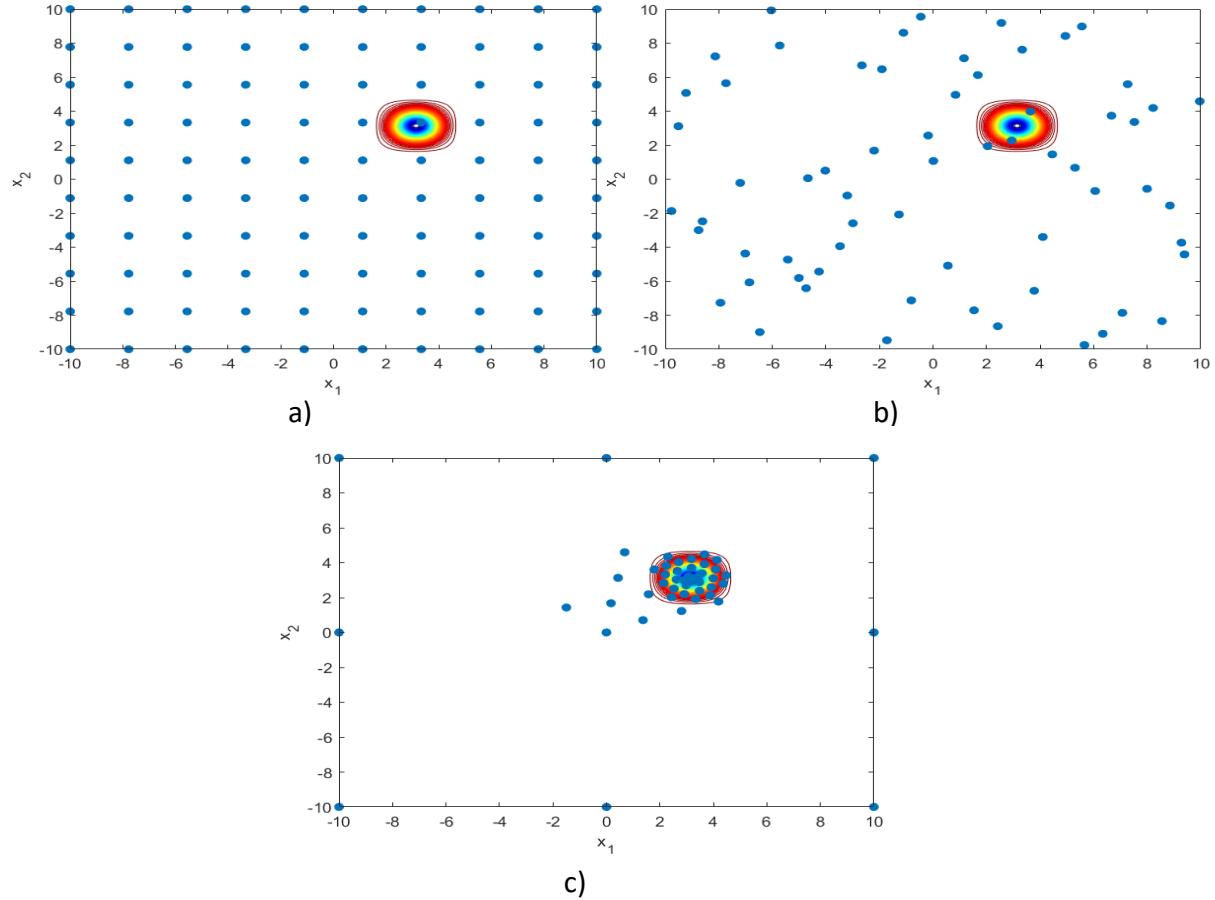


Figure 4.11: Sampling Points obtained from a) Grid Sampling, b) Latin Hypercube Sampling and c) Adaptive Sampling

4.6.2 Multi Valued/Objective Test Functions

4.6.2.1 Fonseca Fleming Function

$$f_1(\mathbf{x}) = 1 - e^{\left(-\left[x_1 - \frac{1}{\sqrt{2}}\right]^2 - \left[x_2 - \frac{1}{\sqrt{2}}\right]^2\right)} \quad (4.19)$$

$$f_2(\mathbf{x}) = 1 - e^{\left(-\left[x_1 + \frac{1}{\sqrt{2}}\right]^2 - \left[x_2 + \frac{1}{\sqrt{2}}\right]^2\right)} \quad (4.20)$$

$$x_1, x_2 \in [-4, 4] \quad (4.21)$$

The actual Fonesca Fleming Function components and their Contours are shown in Fig. 4.12, 4.13. Grid Sampling, Latin Hypercube Sampling and Gradient Based Adaptive Sampling are used to build the metamodel for the above function, the sampling points are shown in Fig. 4.14. Initial Sample for the Adaptive Sampling is generated using grid sampling with 5^2 points, the basis function used for the metamodel is *Inverse Multi-Quadratic*, M1=39(39+25=64), M2 = 3 and $\delta_r = 2\%$. The sampling methods performance is measured using Invalid Sampling Points(ISP's) and Root Mean Square Error(RMSE) and the results are shown in Table. 4.4.

Table 4.4: Comparsion of Sampling Techniques for Fonseca Fleming Function

Measure	Grid Sampling	LHS	Adaptive Sampling
Functional Evaluations	64	64	64
ISP	0	0	0
RSME	0.0225	0.0871	0.0088

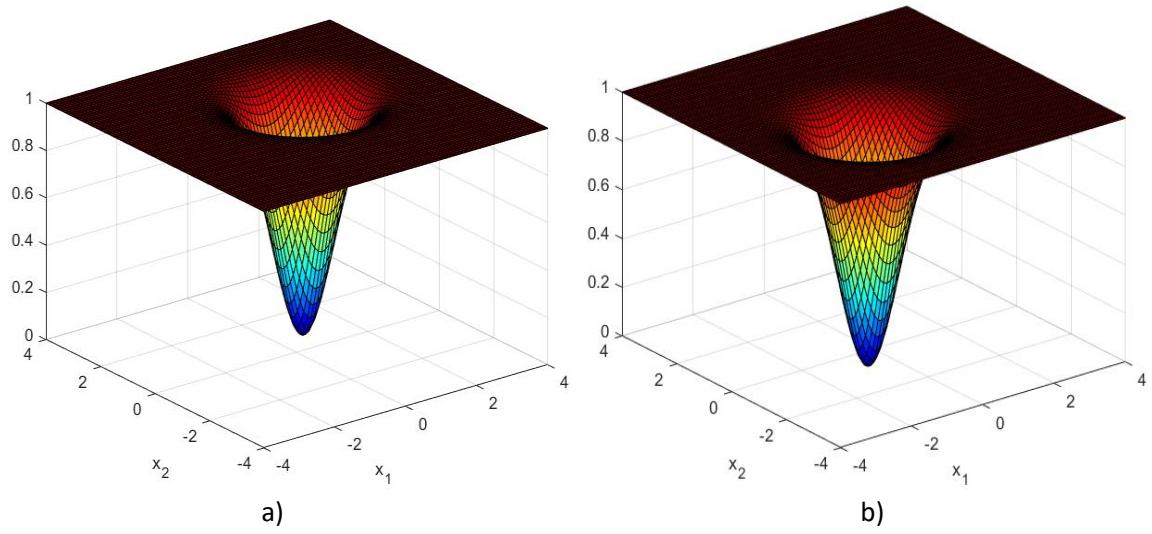


Figure 4.12: a) Fonesca Fleming Function f_1 and b) Fonesca Fleming Function f_2

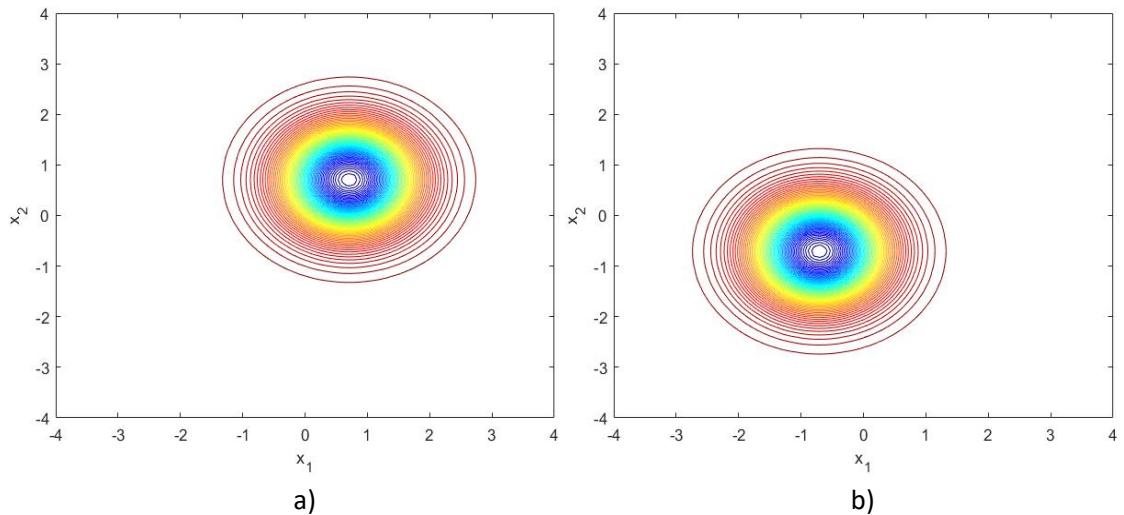


Figure 4.13: a) Contour of Fonesca Fleming Function f_1 and b) Contour of Fonesca Fleming Function f_2

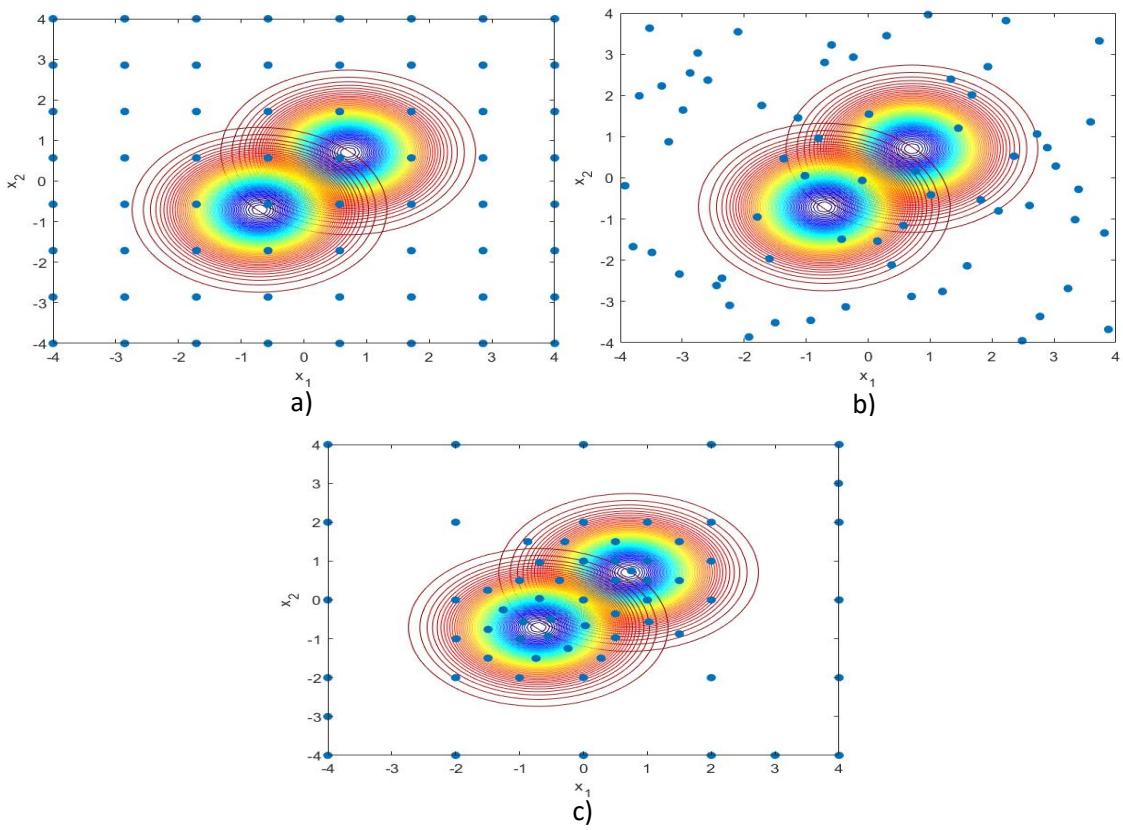


Figure 4.14: Sampling Points obtained from a) Grid Sampling, b) Latin Hypercube Sampling and c) Adaptive Sampling

4.6.2.2 Viennet Function

$$f_1(\mathbf{x}) = 0.5 * (x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2) \quad (4.22)$$

$$f_2(\mathbf{x}) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15 \quad (4.23)$$

$$f_3(\mathbf{x}) = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1e^{-(x_1^2 + x_2^2)} \quad (4.24)$$

$$x_1, x_2 \in [-2, 2] \quad (4.25)$$

The actual Viennet Function components and their Contours are shown in Fig. 4.15, 4.16. Grid Sampling, Latin Hypercube Sampling and Gradient Based Adaptive Sampling are used to build the metamodel for the above function, the sampling points are shown in Fig. 4.17. Initial Sample for the Adaptive Sampling is generated using grid sampling with 5^2 points, the basis function used for the metamodel is *Inverse Multi-Quadric*, M1=75 (75+25=64), M2 = 3 and $\delta_r = 2\%$. The sampling methods performance is measured using Invalid Sampling Points(ISP's) and Root Mean Square Error(RMSE) and the results are shown in Table. 4.5.

Table 4.5: Comparsion of Sampling Techniques for Viennet Function

Measure	Grid Sampling	LHS	Adaptive Sampling
Functional Evaluations	100	100	100
ISP	0	0	0
RSME	0.0950	0.1739	0.0528

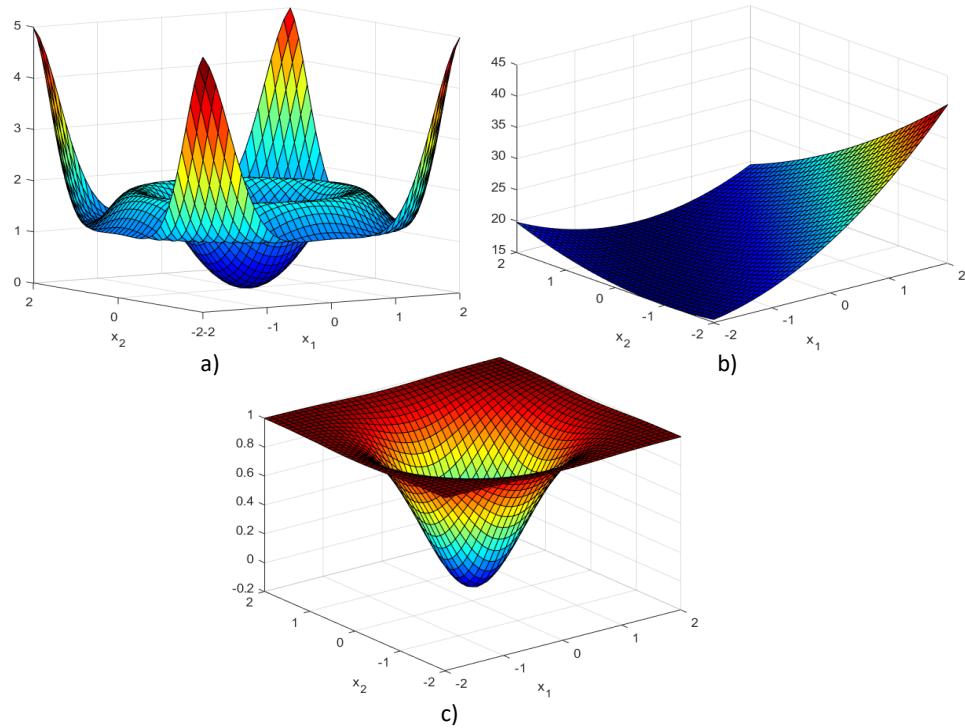


Figure 4.15: a) Viennet Function f_1 , b) Viennet Function f_2 and c) Viennet Function f_3

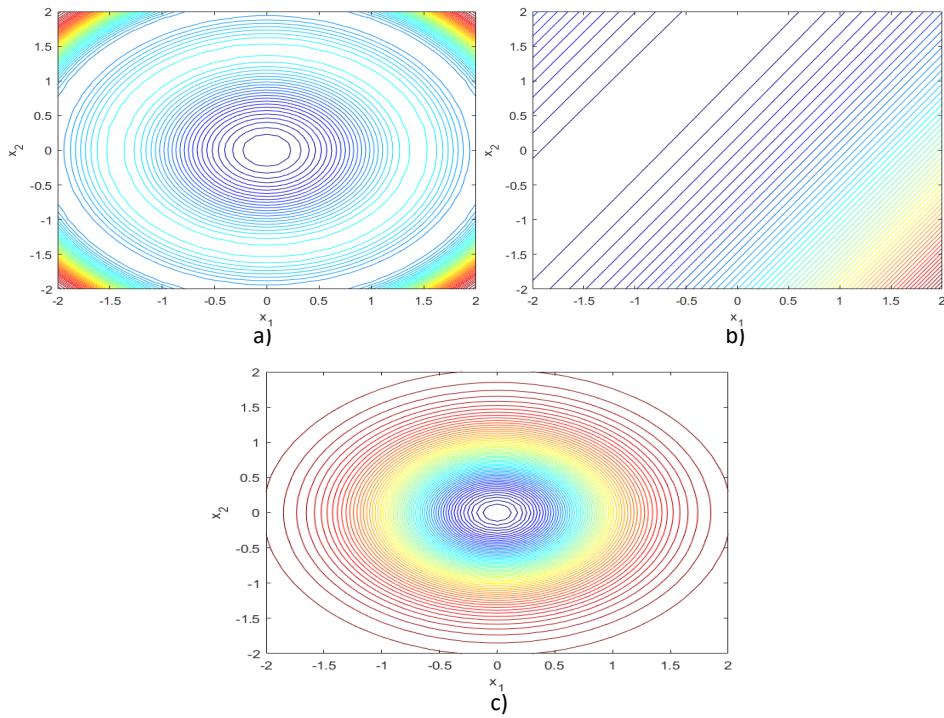


Figure 4.16: a) Contour of Viennet Function f_1 , b) Contour of Viennet Function f_2 and c) Contour of Viennet Function f_3

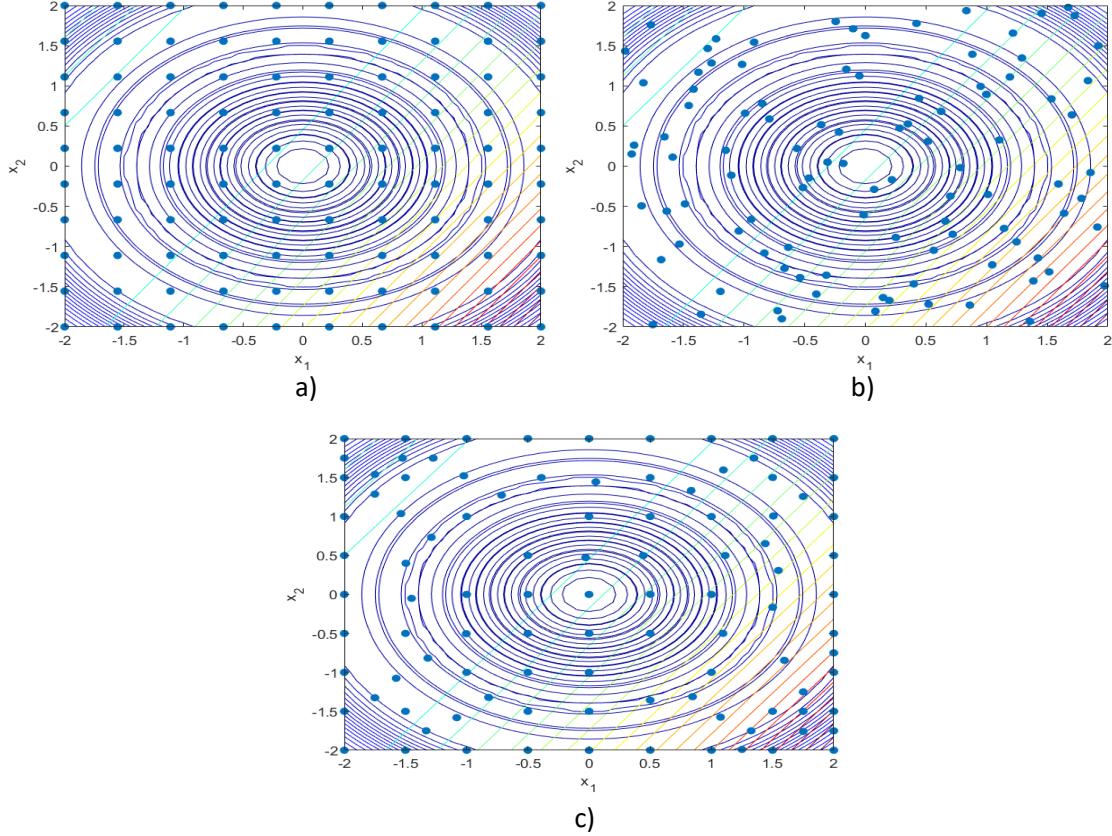


Figure 4.17: Sampling Points obtained from a) Grid Sampling, b) Latin Hypercube Sampling and c) Adaptive Sampling

It can be seen that for all the single valued and multi-valued test functions, the Sampling Points obtained from Gradient Based Sequential Adaptive Sampling provided best function approximation with minimum possible functional evaluations, Invalid Sampling Points and Root Mean Square Error. This is because adaptive sampling chooses the sampling points at which large gradients(or maximum non-linearity) occurs, to gather the best information about the actual target function and builds the best metamodel using those points, this can be evident from the Fig. 4.8, 4.11, 4.14 and 4.17. Next to Adaptive Sampling, Grid Sampling gave better accuracy than Latin Hypercube Sampling. For the current work, Adaptive Sampling and Grid Sampling are chosen for further analyses.

4.7 Metamodel for the Finite Element Analysis

As discussed in the Chapter. 3, it is not possible to use Finite Element Software directly in Optimization processes, a metamodel is built for the finite element analysis using the same techniques as discussed above.

4.7.1 Methodology

For a given input parameters, Velocity(V), Pitch Angle(θ) and Pitch Angle Rate($\dot{\theta}$), the Finite Element Analysis gives Depth of Penetration vs Time, Maximum Acceleration in X and Y directions and Maximum Angular Acceleration in XY plane of Reusable Stage as outputs. The metamodel also expected to give these as outputs but depth of penetration vs time is a curve rather than a single value like other outputs, so a cubic curve is fitted to the actual depth of penetration vs time from the Finite Element analysis and the coefficients of the cubic curve are considered as outputs, so here the input is three-dimensional($q=3$) and output is seven-dimensional(4 cubic curve coefficients+3 accelerations)($p=7$). Grid Sampling and Adaptive Sampling with maximum number of evaluations, 125, are chosen for building the metamodel and the results are discussed in the later chapter.

Chapter 5

Re-Entry Trajectory Module

Trajectory design lies at the heart of the design of launch vehicle either it is ascent or descent. Trajectory optimization helps to find the final mass which is very crucial for cost effectiveness of the launch vehicle. As discussed earlier, the re-entry trajectory for the current vertically landing reusable stage is more or less similar to the launch profile shown in Fig. 1.3. This chapter elaborates the formulation, methodology and optimization of vertically landing re-entry trajectory.

5.1 Equations of Motion

A globally accepted two-dimensional point mass launch vehicle dynamics with rotating earth following oblate spheroid gravity model, given by Tewari[24] is chosen for the current work. It involves three main co-ordinate frames, Earth Fixed Co-ordinate frame(origin at the earth center), Body Frame and Wind Axis Frame. The state of the vehicle at any time can be obtained from radial distance(r), latitude(ϕ), velocity(v), Flight Path Angle(γ) and Attitude Angle or pitch angle(θ). The state equations are written as follows:

$$\dot{r} = v \sin \gamma \quad (5.1)$$

$$\dot{\phi} = \frac{v \cos \gamma}{r} \quad (5.2)$$

$$\dot{v} = \frac{T \cos(\alpha) - D}{m} - g_r \sin \gamma + g_\phi \cos \phi + r w_e^2 \cos \phi (\sin(\gamma - \phi)) \quad (5.3)$$

$$\dot{\gamma} = \frac{T \sin(\alpha)}{mv} + \frac{v \cos \gamma}{r} - \frac{g_r \cos \gamma}{v} - \frac{g_\phi \sin \gamma}{v} + \frac{r w_e^2 \cos \phi}{v} (\cos(\gamma - \phi)) \quad (5.4)$$

$$\dot{m} = -\frac{T}{I_{sp} g_0} \quad (5.5)$$

where α is the angle of attack, T represents the thrust, $w_e = 7.29211 \times 10^{-5}$ rad/s is the

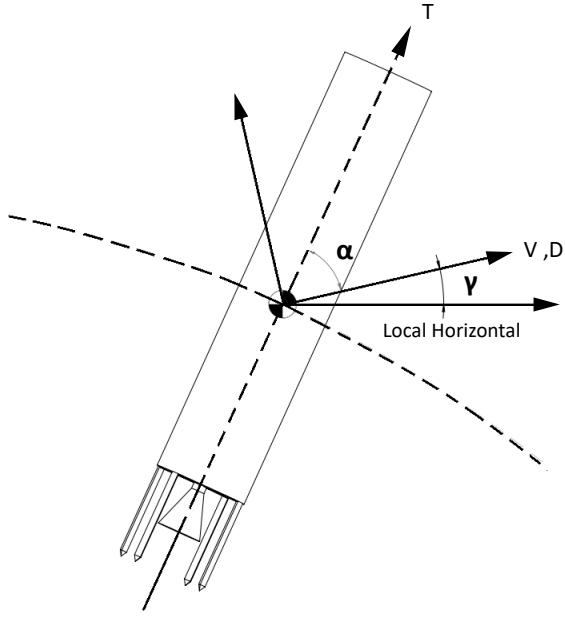


Figure 5.1: Geometry of the Reusable Stage

angular velocity of the earth, m is the mass of the vehicle, g_r and g_ϕ are the oblate gravity components in radial and latitude directions and are given by,

$$g_r = \frac{\mu_E}{r^2} \left[1 - \frac{3}{2} J_2 \left(\frac{R_E}{r} \right)^2 (3 \sin^2 \phi - 1) \right] \quad (5.6)$$

$$g_\phi = -\frac{3\mu_E}{r^2} J_2 \left(\frac{R_E}{r} \right)^2 \sin \phi \cos \phi \quad (5.7)$$

where, μ_E is the Earth's Gravitation Constant ($\mu_E = 3.98600448 \times 10^{14} m^3/s^2$), J_2 is the second zonal harmonic ($J_2 = 1.0826368 \times 10^{-3}$) and R_E is the equatorial radius of earth ($R_E = 6378137 m$). The attitude of the vehicle at any time can be obtained from angle of attack and flight path angle,

$$\theta = \alpha + \gamma \quad (5.8)$$

The Lift term is removed from the above equations of motion because, the angle of attack is equal to 180° during most of the time of flight and there is no significant amount of lift produced, and because of this reason C_D is also assumed constant and taken as 0.75. The thrust T can be written as $T = sT_{avg}$ where s is the throttling parameter which regulates

the amount of thrust required for the re-entry. The exponential density model is considered for the current work,

$$\rho(h) = \rho_0 e^{-\frac{h(t)}{H_0}} \quad (5.9)$$

where $\rho_0 = 1.225 \text{ kg/m}^3$, $H_0 = 7500 \text{ m}$ and $h(t)$ is the altitude as function of time. From all the above equations, The vehicle's state dynamics can be described as follows,

$$\dot{x} = f(x, s, \alpha) \quad (5.10)$$

$$x = [r, \phi, v, \gamma]^T \quad (5.11)$$

Here s and α are the control variables,

$$u = [s, \alpha]^T \quad (5.12)$$

Finding the optimal control variables will optimize the trajectory with respect to some objective function.

5.2 Methodology

As discussed earlier, the vertically landing re-entry trajectory can divided into several phases, one can use any number of phases and as the number increases the complexity of the problem increases. In the current work, it is assumed that the re-entry trajectory is starting after the flipover maneuver by roll control thrusters till the angle of attack becomes 180° , and a total of three phases are considered, *Boostback Burn Phase*, *Aerobraking Phase* and *Landing Burn Phase*.

1. **Boostback Burn Phase:** In Boostback Burn Phase, the reusable vehicle starts burning the engine to reduce the velocity of the stage. It is essential because, at this phase the reusable stage is at higher altitude where the density is very small and hence the drag force, the down range will be so large which is not a desirable condition of effective recovery.
2. **Aerobraking Phase::** In this phase, the engine of the reusable stage shuts off and allowed to fall under gravity maintaining the attitude(grid fins deploy in this phase). First the acceleration due to gravity act against the velocity component in gravity direction till it becomes zero and then the velocity increases in the gravity direction but

at the same time the altitude decreases and density increases therefore aerodynamics drag increases which reduces the velocity.

3. **Landing Burn Phase:** After the aerobraking phase, the final landing burn starts to reduce the leftover velocity to required value and aerodynamics surfaces steer the reusable stage to required attitude.

In the current work, the time lengths required for the above phases(t_{bb} , t_{ff} , t_l) are taken as design variables for the optimization problem, whereas the angle of attack in boostback burn phase and aerobraking phase are taken as 180° but the angle of attack in landing burn phase(α_l) is taken as design variable because, in the current work the reusable stage can also have inclined landing. The throttling parameters in boostback burn phase(s_{bb}) and landing burn phase(s_l) are also considered as design variables.

5.3 Re-Entry Trajectory Optimization

Direct method is used for the re-entry trajectory optimization, the optimal control problem is formulated as Non-Linear Programming Problem(NLP) with time lengths of different phases, angle of attack and throttling parameters as design variables subjected to various terminal constraints. For only optimizing the trajectory, maximizing the final mass of the reusable stage at the time of landing can be taken as objective function for the NLP problem. The mathematical formulation of Re-Entry trajectory optimization is shown below.

5.3.1 Problem Statement

$$\text{Minimize} \quad J = -m\{t_{la}\} \quad (5.13)$$

$$\text{With Respect to} \quad \{s_{bb}, s_{la}, t_{bb}, t_{ff}, t_{la}, \alpha_{la}\} \quad (5.14)$$

$$\text{Subject to} \quad (5.15)$$

$$\text{Terminal Constraints} \quad (5.16)$$

$$c_1 : 5m/s \leq V_{t_{la}} \leq 15m/s \quad (5.17)$$

$$c_2 : 70^\circ \leq \theta_{t_{la}} \leq 110^\circ \quad (5.18)$$

$$c_3 : -20deg/sec \leq \dot{\theta}_{t_{la}} \leq 20deg/sec \quad (5.19)$$

$$c_4 : r_{t_{la}} = R_E \quad (5.20)$$

Where $m\{t_{la}\}$ is the mass of stage, $V_{t_{la}}$ is the velocity, $\theta_{t_{la}}$ and $\dot{\theta}_{t_{la}}$ are pitch and pitch rate respectively at touchdown. Note that the above problem statement is only when the re-entry trajec-

tory is considered for optimization, the objective function may change when it is clubbed with other disciplines. The initial and final values of state variables are shown in Table. 5.1. And the ranges for design variables are shown in Table. 5.2.

Table 5.1: Boundary Conditions

Variables	Initial Value	Final Value
Radial distance(r)	$R_E + 60\text{Km}$	R_E
latitude(ϕ)	13.4191°	free
velocity(v)	1500m/s	5-15m/s
Flight Path Angle(γ)	25°	free
mass	16350	free

Table 5.2: Range of Design Variables

Design Variable	Range	Unit
s_{bb}	[0.5, 1]	-
s_{la}	[0.5, 1]	-
t_{bb}	[0, 15]	sec
t_{ff}	[100, 360]	sec
t_{la}	[0, 15]	sec
α	[160, 180]	deg

5.4 Multi-Disciplinary Design Optimization

As discussed earlier, the design of vertical landing stage recovery is multifaceted and includes many disciplines/subsystems. The result obtained by optimizing one discipline may violate the constraints in other discipline. The traditional optimization techniques like sequential optimization method in which the disciplines/subsystems are optimized sequentially may result in a sub-optimal solution rather than a global optimal solution. Multi-Disciplinary Design Optimization(MDO) is proposed to overcome this scenario. MDO includes all the discipline at once and has one global objective function in which all disciplines interact and compromise with each other and facilitate a search for global minimum rather than sub-optimal solution which are local to the disciplines. MDO has several architectures such as monolithic, and multi-level and each has its own applicability to the design problems.

5.4.1 Multi-Disciplinary Feasible(MDF) Method

In the current work, the trajectory discipline gives the output velocity, attitude and attitude rate and these are directly given to landing dynamics discipline as input which computes the objective function and there are no feedback(inputs) from landing dynamics discipline to trajectory discipline. When the design problem is of this kind, *Multi-Disciplinary Feasible(MDF)* method, a monolithic MDO architecture is the ideal choice[21], [20]. MDF method combines the trajectory and landing dynamics discipline to form a single optimization problem with only one global objective function. The MDF architecture is shown in Fig.

5.4.2 Problem Statement

The main aim of the current work is to find the optimal set of trajectory design variables and velocity, attitude and attitude rates combinations such that the maximum reaction force(or deceleration) in each direction are minimized which results in less impact and no structural damage. The difference between orders of magnitudes of the reaction forces and moment are large, so for uniform search, the reaction forces are divided by mass and reaction moment is divided by moment of inertia. The mathematical form of the problem statement can be written as,

5.5 Problem Statement

$$\text{Minimize} \quad \sqrt{\frac{{R_{x,max}}^2}{m} + \frac{{R_{y,max}}^2}{m} + \frac{{R_{\theta,max}}^2}{I}} \quad (5.21)$$

$$\text{With Respect to} \quad s_{bb}, s_{la}, t_{ff}, t_{bb}, t_{la}, \alpha_{la} \quad (5.22)$$

$$\text{Subject to :} \quad (5.23)$$

$$\text{Terminal Constraints} \quad (5.24)$$

$$c_1 : r = R_E \quad (5.25)$$

$$c_2 : 5m/s \leq v(t_{la}) \leq 15m/s \quad (5.26)$$

$$c_3 : 70^\circ \leq \theta \leq 110^\circ \quad (5.27)$$

$$c_4 : -20deg/s \leq \dot{\theta} \leq 20deg/s \quad (5.28)$$

$$c_5 : d_{p,max} \leq 0.7m \quad (5.29)$$

$$c_6 : m(t_{la}) \geq 8350Kg \quad (5.30)$$

5.5.1 Optimization Solver

Differential Optimization is used for solving the above optimization problem and is described below,

5.5.1.1 Differential Evolution

Storn and Price proposed the Differential Evolution(DE) algorithm for solving optimization problems. It is a population-based metaheuristic technique that optimizes a problem iteratively by improving a candidate solution in the population based on an evolutionary process. In Differential Evolution, each solution in the population is known as *Target Vector*. Each target vector undergoes *Mutation* to form a *Donor Vector* followed by *Cross-Over* to form a *Trail Vector* in each iteration(as shown in Fig. 5.2) and a greedy selection is performed among trail vector and target vector. To put

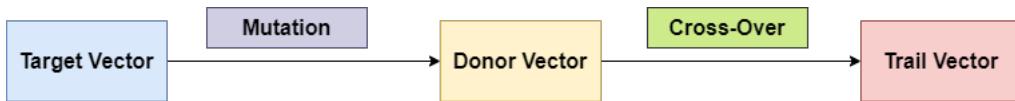


Figure 5.2: Process of Differential Evolution

it mathematically, let us assume the case of minimization of objective function(or fitness function) f , in a D-dimensional domain space. A random population or a set of random possible solutions of size N_p is created using uniformly distributed function or any other random number generating function. Then each possible solution in the population or target vector(X) undergoes Mutation and Cross-over in each iteration as follows,

5.5.1.1.1 Mutation

- There are many Mutation schemes, *Difference Vector* based Mutation is used in the current work to create *Donor Vector* for the target vector(X_i).

$$V = X_{r_1} + F(X_{r_2} - X_{r_3})$$

Where F is the Mutation factor and X_{r_1} , X_{r_2} and X_{r_3} are the randomly selected three distinct target vectors other than X_i .

- For doing this mutation size of population must be greater than or equal to 4.

5.5.1.1.2 Cross-OVER

- There are many Cross-Over schemes, *Binomial Cross-OVER* is used in the current work, the components of the Trail Vector(U_i) are obtained by,

$$u^j = \begin{cases} v^j & \text{if } \text{rand}[0,1] \leq P_c \text{ OR } j = \delta \\ x^j & \text{if } \text{rand}[0,1] > P_c \text{ AND } j \neq \delta \end{cases}$$

Where P_c is the cross over probability and δ is the randomly generated integer $\in [1, D]$. Once all the Trail vectors are formed, the population is updated by greedy selection between Trail vectors and Target vectors. The detailed Differential Evolution is shown in Algorithm. 5.1.

Algorithm 5.1: Differential Evolution

```

1: Input: Fitness Function, lb, ub, T, Np, F,  $P_c$ 
2: Initialize a Random Population( $P(N_p \times D)$ )
3: Evaluate the fitness( $f$ ) of P
4: for  $t = 1$  to  $T$  do
5:   for  $i = 1$  to  $N_p$  do
6:     Generate Donor Vector( $V_i$ ) using Mutation.
7:     Perform Cross-OVER to generate Trail Vector( $U_i$ ).
8:   end for
9:   for  $i = 1$  to  $N_p$  do
10:    Bound  $U_i$ .
11:    Evaluate the fitness( $f_{U_i}$ ) of  $U_i$ .
12:    Perform Greedy Selection using  $f_{U_i}$  and  $f_i$  to update the population(P)
13:  end for
14: end for
```

In the current work, Differential Evolution is implemented using the *Scipy* library's default Differential Evolution routine.

Chapter 6

Results and Discussions

This chapter compiles all the results obtained which are not mentioned in the previous chapters.

6.1 Metamodel for the Finite Element Analysis

Grid Sampling and Gradient Based Adaptive Sampling are chosen as the sampling techniques for the metamodel. Since the dataset is small and functional evaluations are very expensive to obtain, *leave-one-out cross validation* and Invalid Sampling points are taken as the performance metrics. The basis function for the RBFNN and constant c are chosen based on the grid sampling training data and the same are used for Adaptive Sampling. The value of c is chosen to be 0.1, because the values of c lesser than these value causes gradients to explode when r=0. The cross validation errors for different basis functions for c=0.1 are shown in Table. 6.1.

Table 6.1: Leave-one-out Cross Validation Error for all basis funtions for c=0.1

Basis Function	CV error
Linear	0.1063
Cubic	0.0762
Multi-Quadric	0.1061
Inverse Multi-Quadric	0.5857
Gaussian	0.3799

The *cubic* basis function has the least leave-one-out cross validation error, therefore it is chosen as the basis function for Adaptive Sampling and further analyses. The initial sampling set for the adaptive sampling is made using grid sampling of 3^3 points, then the remaining points are obtained sequentially using Gradient Based Adaptive Sampling method. Fig. 6.1 shows the leave one out cross validation error vs number of functional evaluations. Table. 6.2 shows the comparison of sampling techniques used for metamodeling of finite element analysis.

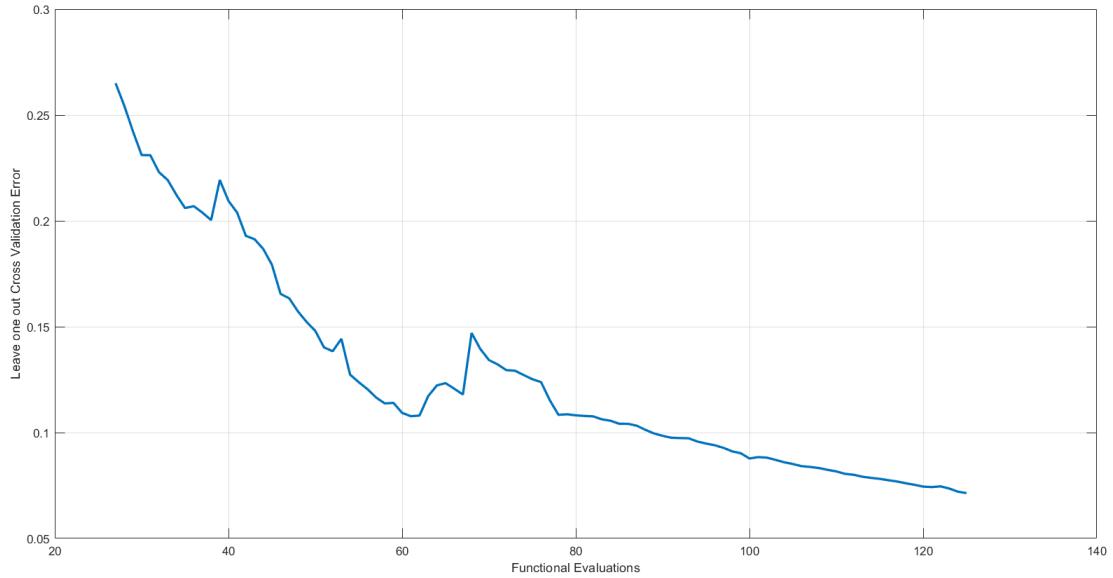


Figure 6.1: Leave one out cross validation error vs functional evaluation for adaptive sampling

Table 6.2: Comparison of sampling techniques

Sampling Method	Cross Validation Error	Functional Evaluations
Grid Sampling	0.0762	125
Adaptive Sampling	0.0762	118
	0.0714	125

Adaptive Sampling method has taken just 118 functional evaluations to equal the error obtained using grid sampling and after 125 functional evaluations, the cross validation error is much lesser than that of grid sampling. This is because the adaptive sampling chooses the points where large gradients/non-linearities occur to obtain the maximum information of the target function.

6.2 Optimization

To emphasize the importance of multi-disciplinary design optimization, first the optimization is done using traditional sequential optimization method in which each discipline is optimized separately with their own problem statements and objective functions, the results of one discipline are used in the other disciplines. The results obtained using the traditional method are compared with multi-disciplinary design optimization method.

6.2.1 Sequential Optimization Method

6.2.1.1 Trajectory Optimization

The trajectory optimization problem(equations (3.13)-(3.20)) is optimized used Differential Evolution first to obtain the velocity, attitude and attitude rate. The Landing dynamics has no independent design variable so it is used just to compute depth of penetration and acceleration in x, y and θ direction. The trajectory is optimized using Differential Evolution with population size of 25 and number of iterations equal to 150. The optimized design variables are shown in Table. 6.3.

Fig.6.2a, 6.2b and 6.2c shows the radial distance, velocity and mass time histories and 6.3 shows the objective values vs number of iterations using differential evolution. Here the total mass required for the trajectory to complete is 699.60705Kg, because our objective function is to maximize the mass. Now Assuming that 699.60705Kg is only used for re-entry and using the terminal values, the accelerations and depth of penetration can be obtained as shown Table. 6.4. The depth of penetration vs time curve obtained from metamodel is shown in Fig. 6.2d.

6.2.1.2 Multi-Disciplinary Design Optimization

The design problem now solved using multi-disciplinary design optimization(equations(5.26)-(5.35)), both the disciplines are now combined to form a global optimization problem and is solved using Differential Evolution solver with population size of 25 and maximum number of iteration equal to 150. The optimized design variables and terminal variables are shown in Table. 6.3.

Fig.6.4a, 6.4b and 6.4c shows the radial distance, velocity and mass time histories and 6.5 shows the objective values vs number of iterations using differential evolution. Here the total mass required for the trajectory to complete is 2549.6738Kg, because our objective function is to maximize the

Table 6.3: Optimized Design Variables and terminal conditions

Variable	Using Traditional Optimization Method	Using MDO
s_{bb}	0.91	0.68
s_{la}	0.57	0.8
t_{bb}	0.026 sec	12.7 sec
t_{ff}	283.15 sec	253.58 sec
t_{la}	8.03 sec	2.98 sec
α_{la}	174.97°	175.96° deg
$v(t_{la})$	14.01 m/s	5.05 m/s
$\theta(t_{la})$	93.56°	95.49° deg
$\dot{\theta}(t_{la})$	2.48 deg/sec	-10.01deg/sec
$m(t_{la})$	15650.3929 kg	13800.33 kg

Table 6.4: Accelerations and depths of penetration obtained from metamodel

Variable	Using Traditional Optimization Method	Using MDO
a_x	17.12 m/s ²	19.35 m/s ²
a_y	289.39 m/s ²	81.28 m/s ²
a_θ	0.22 rad/s ²	2.66 rad/s ²
$d_{p,max}$	0.57 m	0.28 m

mass. Now Assuming that 2549.6738Kg is only used for re-entry and using the terminal values, the accelerations and depth of penetration can be obtained as shown Table. 6.4. The depth of penetration vs time curve obtained from metamodel is shown in Fig. 6.4d.

6.3 Comparison of results

It can be observed that the acceleration and depth of penetration obtained using traditional Sequential Optimization are far more greater than the values obtained from Multi-Disciplinary Design Optimization. The major reason is due to conflicting objectives, in traditional method each discipline is optimized sequentially using the discipline's own problem statement(s) and objective function(s) to obtain the global optimal design, whereas multi-disciplinary design optimization has only one problem statement and objective function in which all disciplines interact and work together to obtain the global optimal solution. This study is one of the few, which shows that traditional optimization method may result in sub-optimal design whereas multi-disciplinary design optimization results in absolute global optimal design. This study shows the importance of Multi-Disciplinary Design Optimization methods when designing the complex systems such as reusable launch vehicle.

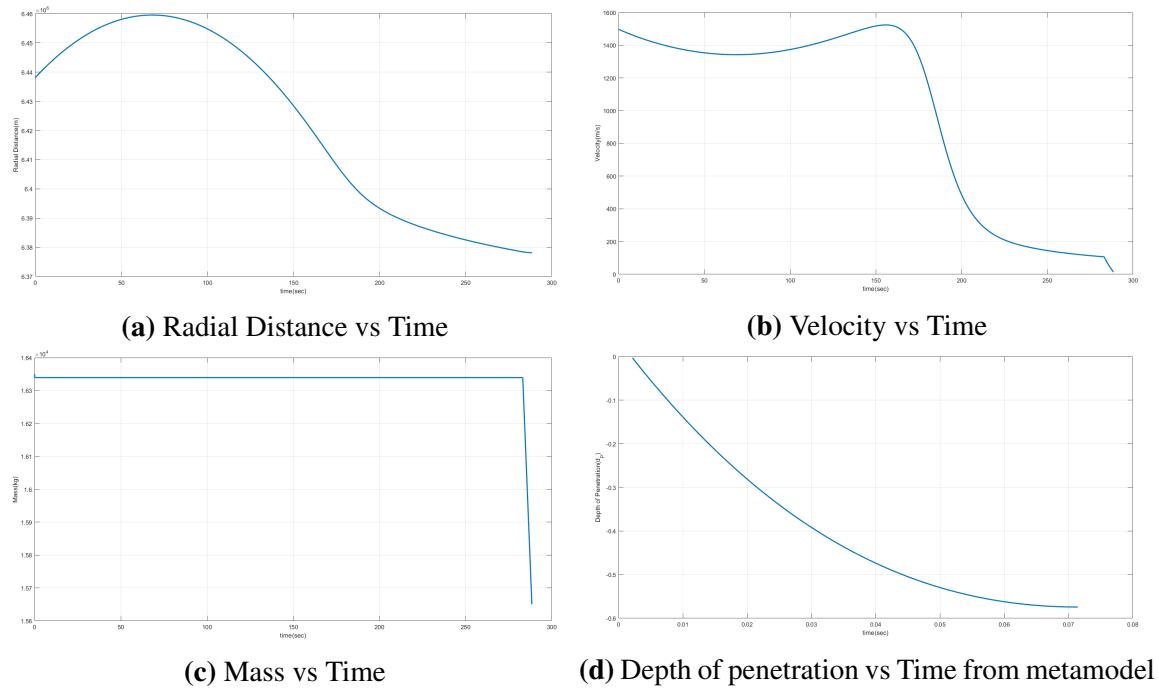


Figure 6.2: Results obtained using traditional Sequential Optimization Method

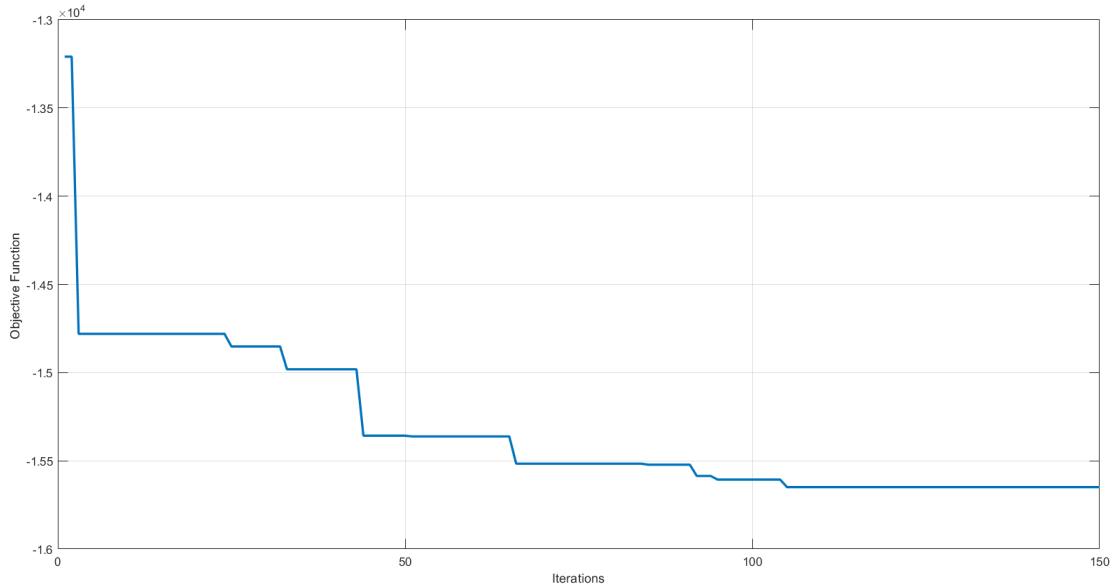


Figure 6.3: Objective Function vs Number of Iterations using Differential Evolution

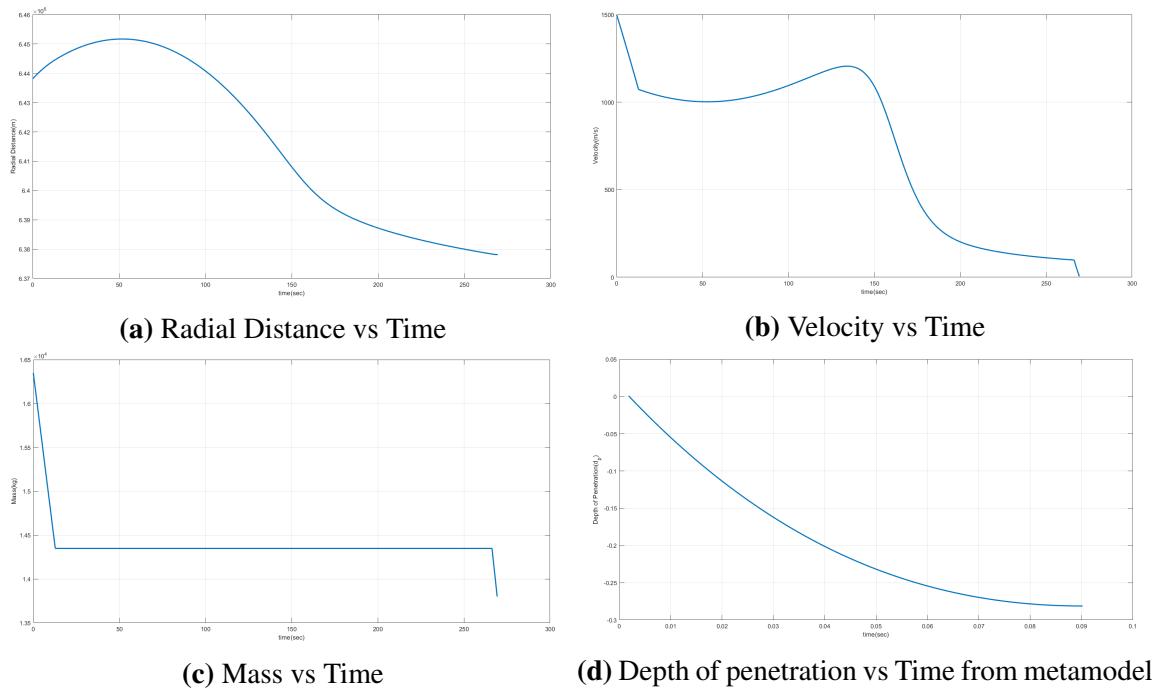


Figure 6.4: Results Obtained from Multi-Disciplinary Design Optimization

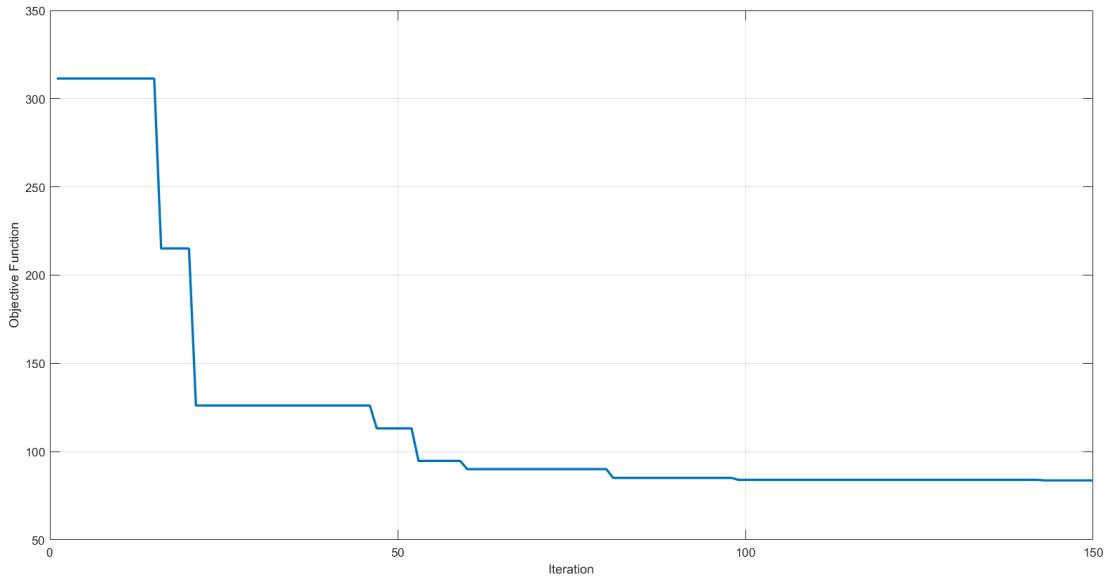


Figure 6.5: Objective Function vs Number of Iterations using Differential Evolution

Chapter 7

Conclusions

The conclusions drawn from the current work are described below,

1. The Coupled Eulerian Lagrangian(CEL) method is proven to be applicable for cone penetration problems. In the current work, Three dimensional analysis of spear penetration in sand is done using the CEL method.
2. Gradient Based Adaptive Sampling outperformed the other sampling methods in the case of both test functions and finite element analysis and the RBFNN metamodel built using the same is employed for further analyses and optimization.
3. The Two Degree of Freedom trajectory is formulated for the current work and is integrated with the finite element analysis metamodel as per the Multi-Disciplinary Feasible(MDF) architecture.
4. Differential Evolution(DE) is used as the optimization solver for both Traditional Sequential optimization method and Multi-Disciplinary Design Optimization Method.
5. The results obtained using the MDO method are far more superior than the results obtained using traditional method.
6. The optimal landing(or minimum impact) is obtained for the pitch angle other than 90° , i.e, with non-zero angle between the axis of the reusable stage and the local vertical and non-zero body rate, which ensures the feasibility of the current recovery method.

So far, in the current work the metamodel is only used to calculate accelerations and depths of penetration and there is no independent design variable for the metamodel. A reliability based optimization using only the metamodel is in progress to find out the optimal combination of velocity, attitude and attitude rate which will result in a reliable stage recovery.

7.1 Future Scope

So far the techniques, methodologies used in the current work are a tip of an iceberg. There is immense scope of extension of the current work to conceptually design more accurate stage recovery.

1. In the current work, the geometry of spears(or landing legs) are taken as fixed, the design problem can be extended such that geometry of the spear can also be taken as design variables.
2. The landing legs(or spears) can be considered as non-rigid and the material can also be included in the design variables.
3. Only three phases are considered for the re-entry trajectory in the current work. It can be extended to many phases.
4. In the current work, the dimension of the output from the metamodel is only seven and it can be extended to include more variables other than accelerations and depth of penetration in the design process.
5. In the current work, only two disciplines are considered, other disciplines like propulsion, structures can also be added to obtain global optimal design.

Bibliography

- [1] “SpaceX - Falcon 9,” <https://www.spacex.com/vehicles/falcon-9/>, accessed: 10-05-2022.
- [2] “How Not to Land an Orbital Rocket Booster,” <https://www.youtube.com/watch?v=bvim4rsNHkQ>, accessed: 10-05-2022.
- [3] K. S. G. Anglim, Z. Zhang, and Q. Gao, “Minimum-fuel optimal trajectory for reusable first-stage rocket landing using particle swarm optimization,” *International Journal of Mechanical and Mechatronics Engineering*, vol. 11, no. 5, pp. 981 – 990, 2017. [Online]. Available: <https://publications.waset.org/vol/125>
- [4] M. Smith, *ABAQUS/Standard User’s Manual, Version 6.9*. United States: Dassault Systèmes Simulia Corp, 2009.
- [5] T. Bansal, “Inelastic analysis of tunnels under generalised bi-axial state of in-situ stress,” Ph.D. dissertation, 05 2017.
- [6] J. Ko, S. Jeong, and J. Kim, “Application of a coupled eulerian-lagrangian technique on constructability problems of site on very soft soil,” *Applied Sciences*, vol. 7, no. 10, 2017. [Online]. Available: <https://www.mdpi.com/2076-3417/7/10/1080>
- [7] A.-B. Ryberg, R. D. Bäckryd, and L. Nilsson, “Metamodel-based multidisciplinary design optimization for automotive applications,” 2012.
- [8] H. Liu, Y.-S. Ong, and J. Cai, “A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design,” *Structural and Multidisciplinary Optimization*, vol. 57, no. 1, p. 393–416, jan 2018. [Online]. Available: <https://doi.org/10.1007/s00158-017-1739-8>
- [9] M. Nazem, J. Carter, D. Airey, and S. H. Chow, “Dynamic analysis of a smooth penetrometer free-falling into uniform clay,” *Géotechnique*, vol. 62, pp. 893–905, 10 2012.
- [10] P. Priyadarshi, L. Joseph, A. Chauhan, J. R. Pillai, M. S. Ahmed, and M. Alam, “Promising indigenous strategies for launch vehicle stage recovery,” VSSC/ACDD/TR/05/2018.

- [11] P. Priyadarshi, K. S. K. Reddy, P. Babu, H. M. R, and S. R. Kamath, “Experimental studies and finite element simulation to verify the spear in sand approach for launch vehicle recovery without engine throttling,” in *National Conference on Multidisciplinary Design, Analysis, and Optimization*, NCMDAO-2020, October, 2020.
- [12] X. Liu, “Fuel-optimal rocket landing with aerodynamic controls,” *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 1, pp. 65–77, 2019. [Online]. Available: <https://doi.org/10.2514/1.G003537>
- [13] J. Zhao, X. He, H. Li, and L. Lu, “An adaptive optimization algorithm based on clustering analysis for return multi-flight-phase of vtv reusable launch vehicle,” *Acta Astronautica*, vol. 183, pp. 112–125, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0094576521001107>
- [14] E. Susila and R. D. Hryciw, “Large displacement fem modelling of the cone penetration test (cpt) in normally consolidated sand,” *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 27, no. 7, pp. 585–602, 2003. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nag.287>
- [15] S. Falla, K. Gavin, and S. Jalilvand, “Numerical modelling of cone penetration test in clay using coupled eulerian lagrangian method,” *In Proceedings of Civil Engineering Research in Ireland 2016: 29th- 30th August, Galway, Ireland*.
- [16] A. Ghosh Dastider, S. Mana, S. Chatterjee, and P. Basu, “Use of coupled eulerian-lagrangian method in modelling of free fall penetration in clay,” 01 2017.
- [17] D. Wang, B. Bienen, M. Nazem, Y. Tian, J. Zheng, T. Pucker, and M. F. Randolph, “Large deformation finite element analyses in geotechnical engineering,” *Computers and Geotechnics*, vol. 65, pp. 104–114, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0266352X1400233X>
- [18] W. Yao, X. Chen, and W. Luo, “A gradient-based sequential radial basis function neural network modeling method,” *Neural Computing and Applications*, vol. 18, pp. 477–484, 06 2009. [Online]. Available: <https://doi.org/10.1007/s00521-009-0249-z>
- [19] X. Wei, Y.-Z. Wu, and L.-P. Chen, “A new sequential optimal sampling method for radial basis functions,” *Applied Mathematics and Computation*, vol. 218, no. 19, pp. 9635–9646, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S009630031200207X>
- [20] J. R. R. A. Martins and A. B. Lambe, “Multidisciplinary design optimization: A survey of architectures,” *AIAA Journal*, vol. 51, no. 9, pp. 2049–2075, 2013. [Online]. Available: <https://doi.org/10.2514/1.J051895>

- [21] M. Balesdent, N. Bérend, P. Dépincé, and A. Chriette, “A survey of multidisciplinary design optimization methods in launch vehicle design,” *Structural and Multidisciplinary Optimization*, vol. 45, pp. 619–642, 2012.
- [22] D. Benson, “Computational methods in Lagrangian and Eulerian hydrocodes,” *Computer Methods in Applied Mechanics and Engineering*, vol. 99, no. 2-3, pp. 235–394, Sep. 1992.
- [23] R. L. Hardy, “Multiquadric equations of topography and other irregular surfaces,” *Journal of Geophysical Research*, vol. 76, pp. 1905–1915, 1971.
- [24] A. Tewari, *Atmospheric and Space Flight Dynamics: Modeling and Simulation with MATLAB® and Simulink®*, ser. Modeling and Simulation in Science, Engineering and Technology. Birkhäuser Boston, 2007. [Online]. Available: <https://books.google.co.in/books?id=0MQSCvCunvoC>