Accelerating Structural Design and Optimization using Machine Learning

Karanpreet Singh

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Rakesh K. Kapania, Chair
Mayuresh J. Patil
Michael K. Philen
Gary D. Seidel
Daniel C. Hammerand

December 06, 2019
Blacksburg, Virginia

Keywords: Structural Design and Optimization, Finite Element Methods, Parallel Processing, Machine Learning, Deep Learning, Active Learning

Copyright 2020, Karanpreet Singh
Accelerating Structural Design and Optimization using Machine Learning
Karanpreet Singh

(ABSTRACT)

Machine learning techniques promise to greatly accelerate structural design and optimization. In this thesis, deep learning and active learning techniques are applied to different non-convex structural optimization problems. Finite Element Analysis (FEA) based standard optimization methods for aircraft panels with bio-inspired curvilinear stiffeners are computationally expensive. The main reason for employing many of these standard optimization methods is the ease of their integration with FEA. However, each optimization requires multiple computationally expensive FEA evaluations, making their use impractical at times. To accelerate optimization, the use of Deep Neural Networks (DNNs) is proposed to approximate the FEA buckling response. The results show that DNNs obtained an accuracy of 95% for evaluating the buckling load. The DNN accelerated the optimization by a factor of nearly 200. The presented work demonstrates the potential of DNN-based machine learning algorithms for accelerating the optimization of bio-inspired curvilinearly stiffened panels. But, the approach could have disadvantages for being only specific to similar structural design problems, and requiring large datasets for DNNs training. An adaptive machine learning technique called active learning is used in this thesis to accelerate the evolutionary optimization of complex structures. The active learner helps the Genetic Algorithms (GA) by predicting if the possible design is going to satisfy the required constraints or not. The approach does not need a trained surrogate model prior to the optimization. The active learner adaptively improve its own accuracy during the optimization for saving the required number of FEA evaluations. The results show that the approach has the potential to reduce the total required FEA evaluations by more than 50%. Lastly, the machine learning is used
to make recommendations for modeling choices while analyzing a structure using FEA. The decisions about the selection of appropriate modeling techniques are usually based on an analyst’s judgement based upon their knowledge and intuition from past experience. The machine learning-based approach provides recommendations within seconds, thus, saving significant computational resources for making accurate design choices.
This thesis presents an innovative application of artificial intelligence (AI) techniques for designing aircraft structures. An important objective for the aerospace industry is to design robust and fuel-efficient aerospace structures. The state of the art research in the literature shows that the structure of aircraft in future could mimic organic cellular structure. However, the design of these new panels with arbitrary structures is computationally expensive. For instance, applying standard optimization methods currently being applied to aerospace structures to design an aircraft, can take anywhere from a few days to months. The presented research demonstrates the potential of AI for accelerating the optimization of an aircraft structures. This will provide an efficient way for aircraft designers to design futuristic fuel-efficient aircraft which will have positive impact on the environment and the world.
Dedication

To my parents.
I would like to give my sincere acknowledgment to my advisor, Prof. Rakesh K. Kapania, who gave me a significant support during my Ph.D. study. He always supported my research interests. Without his help, I wouldn’t be able to finish my Ph.D. program. I would also like to thank the rest of my committee members for their valuable suggestions during my research.

I would like to thank NASA and US Navy SBIR/STTR program for funding my research. I would also like to thank the AOE department of Virginia Tech for awarding me teaching assistanship along with Pratt and Martin-Marietta fellowships awards throughout my Ph.D. study.

I would like to thank my parents who are always a source of positivity, motivation and support. I would also like to thank my best friend and my mentor, Shreyas Saxena, for always guiding in my life and career since my undergrad. I was grateful to make some of the best friends while living here in Blacksburg, especially Stefan Povolny who was my flatmate, class-mate and office-mate from day 1 of Ph.D., and Rikin Gupta who has been my school-mate since my undergrad. I also had wonderful time with Abinash Padhi, Berkan Alanbay, Wei Zhao, Harmee Singh and Shubdildeep Singh. Finally, I must thank my dear Tanya for her love and support.

At the last, I am incapable of expressing my innermost into words for the grace of Almighty, who makes all things possible and provides me with the courage and resilience to overcome all the challenges that life put worth.
# Contents

List of Figures .......................... xii

List of Tables ............................ xxiii

1 Introduction ......................... 1

2 Hybrid Optimization of Curvilinearly Stiffened Shells using Parallel Processing 7

   2.1 Abstract .......................... 7

   2.2 Introduction ....................... 8

   2.3 Methodology ....................... 12

      2.3.1 Procedure .................... 12

      2.3.2 Parameterization of stiffener placement and cross-section ......... 19

      2.3.3 Problem description .......... 22

   2.4 Application and Results .......... 23

      2.4.1 The Shear Dominated Load Case (L1) ............... 25

      2.4.2 The Shear Dominated Load Case (L2) ............... 28

      2.4.3 The Compression Load Case (L3) ............... 30

      2.4.4 The Compression Load Case (L4) ............... 33
3 Buckling Load Maximization of Curvilinearly Stiffened Tow-Steered Laminates using Parallel Processing

3.1 Abstract ................................................................. 37
3.2 Nomenclature .......................................................... 38
3.3 Introduction ............................................................. 39
3.4 Methodology ............................................................. 42
   3.4.1 PSO Based Single Objective Optimization Procedure .......... 43
   3.4.2 NSGA-II Multi-Objective Optimization ............................. 46
   3.4.3 Parallel Processing ............................................... 46
   3.4.4 Glue Contact Capability ......................................... 47
   3.4.5 Parameterization .................................................. 48
3.5 Application and Results ............................................... 53
   3.5.1 Problem Description ............................................ 53
   3.5.2 PSO Based Optimization Results ................................ 55
   3.5.3 NSGA-II Based Optimization Results ............................ 68
3.6 Conclusions ......................................................... 70

4 An Optimization Framework For Curvilinearly Stiffened Composite Pressure Vessels And Pipes

viii
4.1 Abstract ........................................................................................................ 72
4.2 Introduction .................................................................................................... 73
4.3 Methodology .................................................................................................. 76
  4.3.1 Procedure ................................................................................................. 76
  4.3.2 Global Optimization ................................................................................ 77
  4.3.3 Verification Study for *Glue Contact* Capability .............................. 79
4.4 Parameterization Of Stiffener Placement .................................................. 80
  4.4.1 Parameterization Process ....................................................................... 80
  4.4.2 Transformation of Stiffeners from Reference Surface to Physical Surface 82
4.5 Problem Description ....................................................................................... 83
4.6 Application and Results ................................................................................. 83
  4.6.1 Constraints ............................................................................................... 84
  4.6.2 Orthogonal Stiffeners ............................................................................. 86
  4.6.3 Curvilinear Stiffeners ............................................................................. 89
4.7 Conclusions ..................................................................................................... 92

5 Accelerated Optimization of Curvilinearly Stiffened Panels using Deep Learning 95
  5.1 Abstract ........................................................................................................ 95
  5.2 Introduction ................................................................................................... 96
  5.3 Methodology ................................................................................................ 101
List of Figures

1.1 AIRBUS Future Concept Plane .................................................. 2

2.1 Flowchart of the optimization procedure .................................... 15

2.2 Shape Optimization flowchart (PSO) for finding the optimal shape design with maximum buckling load .................................................. 17

2.3 a) Parameterization of stiffener shape and position (shown in red). b) Four arbitrarily-placed stiffeners ................................................. 21

2.4 Design variables for defining cross-section of Z-stringer ................. 21

2.5 Schematic figure of curved panel panel (skin); (a) boundary conditions. (b) three different loads ......................................................... 23

2.6 Optimal panel while designing for the shear load case (L1) using equally spaced straight stiffeners. ................................................. 26

2.7 Global iteration history for the shear load case (L1) using arbitrarily-placed stiffeners .............................................................. 27

2.8 Optimal panel after 2nd global iteration while designing for the shear load case (L1) using arbitrarily-placed stiffeners. ....................... 27

2.9 Optimal panel while designing for the shear load case (L2) using equally spaced straight stiffeners. ................................................. 29

2.10 Global iteration history for the shear load case (L2) using arbitrarily-placed stiffeners ............................................................. 29
2.11 Optimal panel after 8th global iteration while designing for the shear load case (L2) using arbitrarily-placed stiffeners. ........................................... 30
2.12 Optimal panel while designing for the compression load case (L3) using equally spaced straight stiffeners. ................................................................. 31
2.13 Global iteration history for the compression load case (L3) using arbitrarily-placed stiffeners .......................................................... 32
2.14 Optimal panel after the 4th global iteration while designing for the compression load case (L3) using arbitrarily-placed stiffeners. ................................. 32
2.15 Optimal panel while designing for the compression load case (L4) using equally spaced straight stiffeners. ................................................................. 34
2.16 Global iteration history for the compression load case (L4) using arbitrarily-placed stiffeners .......................................................... 34
2.17 Optimal panel after the 4th global iteration while designing for the compression load case (L4) using arbitrarily-placed stiffeners. ................................. 35
3.1 Flowchart of the procedure ................................................................. 44
3.2 Flowchart of the double-level parallel processing ................................. 47
3.3 Benefit of Glue Contact: The stiffener (shown in red) and panel mesh do not need to coincide at the interface .......................................................... 48
3.4 (a) Parameterization of curved fiber paths (b) \( T_0 = 0^\circ \) and \( T_1 = 60^\circ \) with 20 divisions along the width (B). .................................................. 50
3.5 (a) Parameterization of stiffener shape and position \([1]\) (b) Four curvilinear stiffeners design\([1]\) .......................................................... 51
3.6 Plate (1m × 1m) under compressive and shear loads .................. 57

3.7 Composite laminate using equidistant straight stiffeners with straight fibers (Case Study 1) .................. 58

3.8 Composite laminate using straight stiffeners with straight fibers (Case Study 1) .................. 60

3.9 Composite laminate using curved stiffeners with curved fibers (Case Study 1) .................. 60

3.10 Composite laminate using equidistant straight stiffeners with straight fibers (Case Study 2) .................. 63

3.11 Composite laminate using straight stiffeners with straight fibers (Case Study 2) .................. 63

3.12 Composite laminate using curved stiffeners with curved fibers (Case Study 2) .................. 64

3.13 Composite laminate using equidistant straight stiffeners with straight fibers (Case Study 3) ........ 65

3.14 Composite laminate using straight stiffeners with straight fibers (Case Study 3) .................. 66

3.15 Composite laminate using curved stiffeners with curved fibers (Case Study 3) .................. 67

3.16 Pareto-Fronts of NSGA-II Optimization. (Left) Curved Stiffener + Curved Fibers. (Right) Equidistant Straight Stiffeners + straight Fibers ........ 69

3.17 (Left) Pareto-fronts of NSGA-II optimization under pure shear load case. (Right) Improvement in buckling load over use of equidistant straight stiffeners + straight fibers .................. 70

4.1 Flowchart of global optimization .................. 76
4.2 Verification study of glue contact ........................................... 79
4.3 Parameterization of the stiffeners placement on 2D base reference plane ... 80
4.4 Transformation of stiffeners from reference surface to physical surface ........ 81
4.5 Problem description ............................................................. 81
4.6 Orthogonal stiffeners ............................................................ 86
4.7 Optimal configuration using orthogonal stiffeners under torsional load case.
(nh = 1, nl = 12, buckling factor = 1.000, maximum failure index = 0.913,
stress factor (stiffeners) = 0.995, mass = 36.48 kg). The optimal cross-section
dimensions are given in Table 4.5. ................................................ 88
4.8 Optimal configuration using orthogonal stiffeners under bending load case.
(nh = 19, nl = 12, buckling factor = 0.999, maximum failure index = 0.107,
stress factor (stiffeners) = 0.813, mass = 36.10 kg). The optimal cross-section
dimensions are given in Table 4.5. ................................................ 89
4.9 Optimal configuration using curvilinear stiffeners under torsional load case.
(buckling factor = 1.004, maximum failure index = 0.869, stress factor (stiff-
eners) = 0.822, mass = 38.26 kg) the optimal cross-section dimensions are
given in Table 4.7. ................................................................. 92
4.10 Optimal configuration using curvilinear stiffeners under bending load case.
(buckling factor = 0.997, maximum failure index = 0.123, stress factor (stiff-
eners) = 0.916, mass = 32.20 kg) the optimal cross-section dimensions are
given in Table 4.7. ................................................................. 93

5.1 Parameterization of stiffener placement and curvature on a panel ............ 103
5.2 Design variables for defining a rectangular cross-section stiffener 104

5.3 Flowchart of Global Optimization Procedure 106

5.4 Shape Optimization using PSO using a trained DNN in lieu of FEA 108

5.5 Validation of FEA results with literature 110

5.6 A deep neural network (DNN) with input, output and hidden layers shown
   in blue, orange and green respectively 111

5.7 Problem description for training of DNNs 114

5.8 Variation of accuracy with data set of 50,000 data points 117

5.9 Variation of accuracy with data set of 100,000 data points 118

5.10 Variation of accuracy with data set of 190,000 data points 118

5.11 Variation of loss function of training and validation set with number of epochs 119

5.12 Variation of the buckling parameter with respect to aspect ratio of panel and
   height of the stiffener using FEA and DNN 120

5.13 Variation of the buckling parameter with respect to thickness and height of
   the stiffener using FEA and DNN 121

5.14 Variation of accuracy with different sizes of dataset and degree of the polynomial 122

5.15 Comparison of CPU run time for conducting optimization sequentially, with
   parallel processing and with use of a DNN 123

5.16 Comparison of optimization history of case study I: (Left) Using FEA in PSO
   iterations, (Right) Using DNN in PSO iterations 124
5.17 Optimal design in case study I by using FEA in PSO iterations. Left: Geometry, Middle: $1^{st}$ Buckling mode, Right: von Mises stress ............... 125

5.18 Optimal design in case study I by using DNN in PSO iterations. Left: Geometry, Middle: $1^{st}$ Buckling mode, Right: von Mises stress ............... 125

5.19 Comparison of optimization history of case study II: (Left) Using FEA in PSO iterations, (Right) Using DNN in PSO iterations ................. 126

5.20 Optimal design in case study II by using FEA in PSO iterations. Left: Geometry, Middle: $1^{st}$ Buckling mode, Right: von Mises stress ............... 126

5.21 Optimal design in case study II by using DNN in PSO iterations. Left: Geometry, Middle: $1^{st}$ Buckling mode, Right: von Mises stress ............... 127

6.1 Flowchart of the constrained optimization procedure. ......................... 137

6.2 (a) A toy data set of 600 instances available from scikit-learn [2]; (b) 50 labeled instances randomly picked from the dataset for initial training of active learner 140

6.3 (a) Confidence plot after initial training of active learner made of decision trees (accuracy: 68%); (b) 183 queries issued by the learner; (c) confidence plot after including queries in the training dataset (accuracy: 93%) .......... 141

6.4 (a) Confidence plot after initial training of active learner made of ANNs (accuracy: 85%); (b) 144 queries issued by the learner; (c) confidence plot after including queries in the training dataset (accuracy: 91%) ............... 142

6.5 (a) Confidence plot after initial training of active learner made of ANNs (accuracy: 78%); (b) 419 queries issued by the learner; (c) confidence plot after including queries in the training dataset (accuracy: 89%) ............... 143
6.6 Flowchart of the constrained optimization procedure using active learning. 146

6.7 Ten-bar truss problem. ................................................................. 148

6.8 Analysis of different parameters of active learner on feasibility of optimal design in the ten-bar truss problem. ............................................. 149

6.9 Comparison of the optimal solution with Haftka [3] solution using different optimization parameters in the ten-bar truss optimization problem. . . . . . . 150

6.10 Comparison of % savings in the total evaluations using active learning in the ten-bar truss problem. ............................................................. 151

6.11 Comparison of % savings using ALGA over constrained GA and penalty method (Case Study I) ................................................................. 151

6.12 Comparison of % savings in total evaluations during the optimization of ten-bar structure ................................................................. 152

6.13 Analysis of different parameters of active learner on feasibility of optimal solution in Hesse function. ......................................................... 154

6.14 Comparison of the optimal solution with analytical solution using different population sizes in the Hesse Function problem (Case Study II). . . . . . . 155

6.15 Comparison of % savings in the total evaluations using active learning in the Hesse Function problem. ............................................................. 156

6.16 Comparison of % savings using ALGA over constrained GA and penalty method (Case Study II). ................................................................. 156

6.17 Comparison of % savings in the total evaluations during optimization of Hesse Function problem. ............................................................. 157
6.18 (Left) Parameterization of stiffener placement on panel. (Middle) Applied loading condition on panel. (Right) Panel’s thickness and stiffener’s cross-section design variables. ................................. 159

6.19 Comparison of optimal weights in optimization in curvilinearly stiffened panel. 159

6.20 Comparison of % savings in the total evaluations in optimization in curvilinearly stiffened panel. ........................................... 160

6.21 Comparison of % savings in the total evaluations in optimization in curvilinearly stiffened panel. ........................................... 160

6.22 Optimal design using ALGA. Left: Geometry, Middle: 1st Buckling mode, Right: von Mises stress. (Mass = 3.21 kg) ......................... 161

6.23 Optimal design using constrained GA. Left: Geometry, Middle: 1st Buckling mode, Right: von Mises stress. (Mass = 3.23 kg) ............. 162

6.24 Optimal design using penalty method based GA. Left: Geometry, Middle: 1st Buckling mode, Right: von Mises stress. (Mass = 3.28 kg) .... 162

7.1 Combined hierarchical qualification/classical validation approach. ........ 170

7.2 Example T-joint structure under applied traction with key geometric sizing parameters indicated. .................................................. 172

7.3 Mesh convergence study for a full 3-D model of a T-joint. ................... 173

7.4 Contour plot of strain energy density in T-Joint: (Left) Full 3-D, (Right) Full 2-D. ................................................................. 174

7.5 Contour plot of displacement in T-Joint: (Left) Full 3-D, (Right) Full 2-D. 174
7.6 Contour plot of maximum principal stress in T-Joint: (Left) Full 3-D, (Right) Full 2-D.

7.7 Contour plot of maximum principal strain in T-Joint: (Left) Full 3-D, (Right) Full 2-D.

7.8 Contour plot of von Mises stress in T-Joint: (Left) Full 3-D, (Right) Full 2-D.

7.9 Different models of cantilever beam with different number of through-thickness elements.

7.10 Peak bending stress convergence with varying number of thru-the-thickness elements (Elm denotes element integration point values, while Nodal refers to interpolated nodal values).

7.11 T-Joint modeling: (a) multi-fidelity model with 2-D and 3-D element topologies and (b) results region of interest with grey color showing edge regions near applied traction and kinematic boundary conditions that are not considered in the results comparisons.

7.12 Parameterization of the multi-fidelity T-Joint model with 2-D and 3-D regions indicated in orange and blue, respectively.


7.14 Stepped plate example structure with geometric parameters and applied traction.

7.15 Mesh convergence study for full 3-D stepped plate model.
7.16 Contour plot of strain energy density in a stepped plate: (Top) Full 3-D, (Bottom) Full 2-D. ................................. 183

7.17 Contour plot of displacement in a stepped plate: (Top) Full 3-D, (Bottom) Full 2-D. ................................. 184

7.18 Contour plot of maximum principal stress in a stepped plate: (Top) Full 3-D, (Bottom) Full 2-D. ................................. 184

7.19 Contour plot of maximum principal strain in a stepped plate: (Top) Full 3-D, (Bottom) Full 2-D. ................................. 185

7.20 Contour plot of von Mises stress in a stepped plate: (Top) Full 3-D, (Bottom) Full 2-D. ................................. 185

7.21 Stepped plate modeling: (a) multi-fidelity model with 2-D and 3-D element topology and (b) results region of interest with grey areas near kinematic and traction boundary conditions not considered. ................................. 186

7.22 Stepped plate parameterization of multi-fidelity model with 2-D and 3-D regions indicated in orange and blue, respectively. ................................. 187

7.23 Contour plots with respect to $\alpha$ and $\beta$ for %error in a) Total Strain Energy b) Max. Disp. c) Max. Abs. Principal Stress d) Max. Principal Strain e) Max. von Mises Stress in stepped plate problem. ................................. 188

7.24 Multi-Input/Multi-Output (MIMO) neural network with two hidden layers. ................................. 190

7.25 Machine learning modeling recommendations using fixed error level. ................................. 193

7.26 Machine learning modeling recommendations using variable error level. ................................. 194
7.27 ANN performance improvement with training epochs for T-Joint using a prescribed 5\% error limit. .......................................................... 197

7.28 ANN performance improvement with training epochs for variable error level case study of T-Joint. .................................................. 199

7.29 ANN performance improvement with training epochs for fixed error level case study of stepped plate. ........................................ 201

7.30 The ANN performance improvement with training epochs for variable error level case study of stepped plate. ................................. 203
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Range of the design variables (Arbitrarily-Placed Case) (See Fig. 2.3)</td>
<td>22</td>
</tr>
<tr>
<td>2.2</td>
<td>Load Cases [4]</td>
<td>23</td>
</tr>
<tr>
<td>2.3</td>
<td>Range for design variables of $X_{size}$ (Fig. 2.4)</td>
<td>24</td>
</tr>
<tr>
<td>2.4</td>
<td>Dimensions and Material Properties of the Model</td>
<td>25</td>
</tr>
<tr>
<td>2.5</td>
<td>Optimal final values for $X_{size}$ (Fig. 2.4) for the shear load case (L1)</td>
<td>26</td>
</tr>
<tr>
<td>2.6</td>
<td>Optimal final values for $X_{size}$ (Fig. 2.4) for the shear load case (L2)</td>
<td>30</td>
</tr>
<tr>
<td>2.7</td>
<td>Optimal final values for $X_{size}$ (Fig. 2.4) for the compression load case(L3)</td>
<td>33</td>
</tr>
<tr>
<td>2.8</td>
<td>Optimal final values for $X_{size}$ (Fig. 2.4) for the compression load case(L4)</td>
<td>35</td>
</tr>
<tr>
<td>3.1</td>
<td>Range of the design variables for placement of four stiffener (See Fig. 3.5)</td>
<td>52</td>
</tr>
<tr>
<td>3.2</td>
<td>Range for size design variables</td>
<td>53</td>
</tr>
<tr>
<td>3.3</td>
<td>Load cases</td>
<td>55</td>
</tr>
<tr>
<td>3.4</td>
<td>Material properties used for modeling composite laminate skin</td>
<td>56</td>
</tr>
<tr>
<td>3.5</td>
<td>Comparison of the optimized results under shear dominated load case (Case Study 1)</td>
<td>59</td>
</tr>
<tr>
<td>3.6</td>
<td>Comparison of the optimized results under compression dominated load case (Case Study 2)</td>
<td>62</td>
</tr>
<tr>
<td>3.7</td>
<td>Comparison of the optimized results under pure shear load case (Case Study 3)</td>
<td>68</td>
</tr>
</tbody>
</table>
4.1 Material properties for composite laminate (carbon fiber) used for modeling skin ............................................. 85
4.2 Isotropic material properties used for modeling stiffeners .................................................. 85
4.3 Range of Xshape for orthogonal stiffeners ................................................................. 88
4.4 Range of Xsize design variables ................................................................. 88
4.5 Optimal final values of design variables for orthogonal stiffeners ........................................ 89
4.6 Range of Xshape for curvilinear stiffeners ................................................................. 91
4.7 Optimal final values of design variables for curvilinear stiffeners ........................................ 91
4.8 Comparison of different configuration under bending load case ........................................ 92

5.1 Domain for creating LHS based data-set (See Fig. 5.7) .............................................. 116
5.2 Percentile based accuracy of DNN over the dataset of 190,000 data points .. 120
5.3 Load cases and panel dimensions for case studies .............................................. 123

6.1 Data for ten bar truss [3] ............................................................... 147
6.2 Load case and panel dimensions .......................................................... 158

7.1 Dimensions of a T-joint. .......................................................... 171
7.2 Comparison of fully 2-D and fully 3-D results for a T-joint. ........................................ 173
7.3 Comparison of multi-fidelity model and high-fidelity (full 3-D) results for a
    T-joint (α = β = 0.4). .......................................................... 179
7.4 Comparison of computational cost of the models shown in Table 7.3. .... 180

xxiv
7.5 Comparison of multi-fidelity model and high-fidelity (full 3-D) results for a T-joint ($\alpha = \beta = 0.1$). .................................................. 181
7.6 Dimensions of a stepped plate. ................................................. 182
7.7 Comparison of 2-D and 3-D results for a stepped plate structure. .... 182
7.8 Comparison of multi-fidelity model and high-fidelity (full 3-D) results for a stepped plate ($\alpha = \beta = 0.3$) ................................................. 187
7.9 Comparison of computational cost of the models shown in Table 7.8 ... 187
7.10 Comparison of multi-fidelity model and high-fidelity (full 3-D) results for a stepped plate ($\alpha = \beta = 0.1$). ................................................. 189
7.11 Few test data points in the T-joint problem. ................................. 196
7.12 Modeling recommendation predictions using the trained ANN for the test data points (Table 7.11) in the T-joint problem. ......................... 197
7.13 Error prediction using ANN for data points from the test set for a T-joint. 199
7.14 Modeling recommendations for different error limits for a T-Joint. .... 199
7.15 Few test data points in stepped plate problems. ............................. 201
7.16 Modeling recommendations using trained ANN for test data points (Table 7.15) in stepped plate problems. ................................................. 202
7.17 Error prediction using ANN for select data points from the test set for the stepped plate example. ...................................................... 203
7.18 Modeling recommendations based on different error limits for a stepped plate problem. ................................................................. 204
Chapter 1

Introduction

An important objective for the aerospace industry is to design robust and fuel efficient aerospace structures. The advances in the manufacturing capabilities have made it possible to design and manufacture bio-inspired futuristic aerospace structures. These bionic structures mimic the bone structure of birds. These structures have the potential to save structural weight and fuel burn in an aircraft. Airbus\(^1\), as shown in Fig. 1.1, presented a concept for 3D printed future aircraft cabin in which the innovative design mimics the organic cellular structure and bone growth in living organisms. The idea could help in saving significant weight in comparison to the current manufactured designs. However, there is still a motivation in the industry towards implementing the conventional aerospace structures because of the reasons such as manufacturing constraints, enormous design space, and especially the lack on computational environment for designing unconventional aerospace structures. Finite Element Analysis (FEA) is generally used in various industries to design large complex structures. However, the FEA based standard optimization methods for aircraft panels with bio-inspired curvilinear stiffeners are computationally expensive. The research work in this thesis is conducted with the motivation to develop an efficient computational design environment while taking a step forward in the direction to design futuristic aerospace structures. Different optimization approaches are proposed with the motivation to accelerate the conventional FEA-based structural design and optimization. Chapters 2-

---

7 in this thesis are based on journal/conference papers presented, and extensive relevant
literature survey is presented in those chapters.

The next three chapters of this thesis discuss about the use of different optimization schemes
on complex non-convex optimization problems based on futuristic aerospace structures.
These studies show that enormous computational resources and time are required to optimize
these complex structures when using FEA. Parallel processing was first used to decrease the CPU time. Subsequently, as detailed in the subsequent chapters, the use of machine learning is proposed to accelerate the structural design and optimization by replacing the FEA evaluations using the machine learning based surrogate models.

Firstly, Chapter 2, which appeared as “Hybrid Optimization of Curvilinearly Stiffened Shells Using Parallel Processing” in AIAA Journal of Aircraft in 2019, presents different optimization studies with the motivation to find the benefit of using arbitrarily-placed curvilinear stiffeners on the shell structures. The optimization studies are implemented with an objective to reduce structural mass of the structure. A two-step hybrid optimization including the shape optimization and the size optimization are considered for optimal stiffener configurations for the panels that can be used for fuselage and space launch vehicles designs. The shape optimization is conducted using a gradient-free optimization technique Particle Swarm Optimization (PSO) and the size optimization is implemented using a gradient-based optimization technique (GBO). The optimization studies show that the curvilinear stiffeners have the potential to save the weight of the structures in comparison to the conventional equidistant straight stiffeners. The optimization of a panel takes a few days of CPU run-time if it is run sequentially. The parallel processing is used to reduce the time consumption.

In Chapter 3, which appeared as “Buckling Load Maximization of Curvilinearly Stiffened Tow-Steered Laminates” in AIAA Journal of Aircraft in 2019, different optimization case studies are presented to design curvilinearly stiffened tow-steered composite laminates. The objective of the optimization is to maximize the buckling load of the structure. The optimization is based on the use of the single gradient-free optimization technique (PSO). These studies showed the benefits of using curvilinear stiffeners and curvilinear fibers for improving the buckling load of the structure. However, again the major disadvantage of using FEA-based PSO is observed that it requires many FEA evaluations, thus, resulting in sig-
nificant computational resources and time. In the presented case studies, parallel processing is employed to reduce the CPU run time.

Chapter 4 was presented as “An Optimization Framework for Curvilinearly Stiffened Composite Pressure Vessels and Pipes” in ASME 2017 Pressure Vessels and Piping Conference, and is under review for journal publication in Advances in Computational Design (ACD). The chapter presents a bi-level programming (BLP) based optimization technique for designing curvilinearly stiffened composite pressure vessels and pipes. In this technique, the GBO is used inside the PSO such that the size optimization is conducted for each particle of the PSO. In literature, this method has been shown to have an advantage of converging faster than using PSO. However, the optimization takes about a day while making use of parallel processing because of expensive FEA evaluations. The use of curvilinear stiffeners showed a potential to save structural mass in comparison to the use of conventional straight stiffeners.

After conducting the above mentioned optimization studies on different complex problems, it is observed that the parallel processing alone can not reduce the CPU run time. Therefore, the use of machine learning is studied in the remaining chapters to accelerate the structural design and optimization.

In Chapter 5, which is under review for journal publication in AIAA Journal of Aircraft, the use of deep learning is proposed to accelerate the optimization of curvilinearly stiffened panels. The framework developed in the Chapter 2 is used, but the FEA evaluations in the PSO are replaced with a trained deep neural network (DNN). This approach showed a significant saving of computational time and resources. The DNN-based approach is proposed such that a single DNN can be used to predict the buckling load of large number of curvilinearly stiffened panels under different load cases. This approach reduced the optimization CPU run time from days to minutes. The optimal designs are very close to the ones found
using FEA-based optimization case studies. The approach doesn’t need to have a new neural network whenever a new curvilinearly stiffened panels has to be designed. The benefits of the approach over what has already been done in the literature are also presented.

Chapter 5 shows that the DNNs can be successfully used as surrogate models to replace expensive FEA evaluations. However, large datasets are needed for getting high accuracy of the surrogate model. Also, pre-trained surrogate models only work for a specific optimization problem. In Chapter 6, which is under review for journal publication in AIAA Journal, an adaptive machine learning technique, called active learning, is used to accelerate the evolutionary optimization of complex structures. This approach is more general to any structural optimization problem. An active learner is a machine learning-based model which can interactively query the outputs of certain data points, whenever the model would be uncertain about those outputs. The approach does not need a trained surrogate model prior to the optimization. The active learner adaptively learns about the structure during the optimization to improve the computational performance. The results show that the approach has the potential to reduce the total required constraint evaluations by more than 50%.

In the research work presented in from Chapters 2-6, the main motivation is to design a structure efficiently. An another important task for a design agent while designing a structure is to make right modeling choices while analyzing a structure. FEA has been well-established for modern structural design and analysis. However, significant modeling choices must be made to achieve valid answers sufficient for engineering decisions, but without employing high computational effort. In Chapter 7, which was presented as “Machine Learning Approaches for Finite Element Modeling Recommendations” in ASME V&V Verification and Validation Conference in 2019, the subjective nature of such decision making is removed via comprehensive multi-level qualification scheme where high-fidelity models are used to judge sufficiency
of low and mid-fidelity models. Machine learning software are trained based on these modeling assessment results to make recommendations for creating valid models for arbitrary geometric and load parameters. This approach provides recommendations within seconds, thus, saving significant computational resources for making accurate modeling choices.

In conclusion, the research work conducted in this thesis shows that machine learning techniques have a great potential to accelerate structural design and optimization. The pre-trained machine learning software can be used to replace expensive FEA evaluations. If it is difficult to get required datasets, active learning-based evolutionary optimization can be used to save expensive constraints evaluations. At last, the Chapter 8 summarizes the research work along with the future work.
Chapter 2

Hybrid Optimization of Curvilinearly Stiffened Shells using Parallel Processing

2.1 Abstract

With the advances being made in additive manufacturing, it is becoming increasingly possible to fabricate a broad class of complex-shaped designs for practical applications. This manufacturing capability has allowed structural designers to implement the use of bio-inspired curvilinear stiffeners for achieving better designs of stiffened plate and shell structures. Curvilinear stiffeners have proven to be useful over straight stiffeners in some applications for achieving better structural efficiency. A framework has been developed that employs a hybrid optimization technique of using Particle Swarm Optimization (PSO) for stiffener shape optimization and gradient based optimization as implemented in MSC.NASTRAN SOL 200 for optimization of stiffeners’ cross-section and shell thickness. Parallel processing has been utilized to save extensive wall-clock time. The framework has been employed for optimally designing cylindrical shells/panels stiffened by arbitrarily-placed stiffeners, the motivation being lighter-weight fuselage and space launch vehicle designs. Structural optimization results have been presented for the optimal design of stiffened shells with the
objective of weight minimization for shells subjected to both buckling and stress constraints. Optimization studies show that both the stiffener placement and the stiffener geometric curvature influence the shell buckling load, which can help to decrease the structural weight by optimizing the cross-sectional dimensions for stiffeners and panel thickness. In this chapter, cylindrical shells, stiffened by four stiffeners, placed arbitrarily, are studied under compression and shear load cases. It is seen that arbitrarily-placed curvilinear stiffeners lead to a potential 13% weight saving as compared to the use of equally spaced straight stiffeners when designed for the compression load case.

2.2 Introduction

An important objective for the aerospace industry remains to develop robust, multi-functional and fuel-efficient aerospace structures. Stiffened cylindrical shells are one of the important components in the aerospace industry. Over the past decades, enormous research has been done in this area with the motivation to continuously improve the structural design and analysis of fuselages and space launch vehicles. Latest developments in manufacturing technologies, especially additive manufacturing, have made this objective increasingly possible by manufacturing complex-shaped designs for practical applications. The availability of high performance computing and computing software such as MSC.NASTRAN, MSC.PATRAN have now made it possible to both analyze and optimize a large class of complex-shaped designs. Commercially available geometric modeling software, such as Rhinoceros 3D, have made it possible to create, using Non-Uniform Rational B-Splines (NURBS), geometric modeling of highly complex shapes.

The conventional tube-like fuselage used in modern aircraft is a cylindrical shell stiffened by straight stringers, and longerons placed along the longitudinal direction and rings placed
along circumferential direction. The use of straight stiffeners leads to a limited design space. To enhance the design space for a possible better stiffened shell design, one might consider the use of curved, non-uniform thickness stiffening members as well as changing the panel thicknesses in the domain between two stiffeners.

During the last decade, significant research has been conducted on the use of curvilinear stiffeners for designing stiffened panels. Inspired by structures found in nature, Kapania, Li and Kapoor [5] have shown that the use of curvilinearly stiffened panels might lead to lighter weight designs than panels with straight stiffeners for certain design loads. Subsequently, Mulani, Slemp and Kapania [4] developed an efficient optimization framework, EBF3PanelOpt, for curvilinear blade-stiffened panels. In their framework, Particle Swarm Optimization (PSO) is utilized as the optimizer to handle the discontinuous shape design space, resulting from changes in stiffeners’ location, with the motivation to minimize the weight of the panel stiffened with two stiffeners while being subjected to various constraints including both the yield strength and buckling. Jrad, Mulani and Kapania [6] included damage tolerance in EBF3PanelOpt to optimize curvilinearly stiffened panels subjected to in-plane normal and shear loads and considering the possible presence of small cracks. To see the benefit of curvilinear stiffeners in composite laminates, Zhao and Kapania [7] studied the influences of the stiffener shape and laminate configuration on the buckling response of a curvilinearly stiffened composite plate subjected to in-plane normal and in-plane shear loads. Their parameterization study results showed that it is possible to improve the structural buckling responses by tailoring the stiffener shape and placement with less weight penalty as compared to that using straight stiffeners. Wang and Abdalla [8] have performed several parametric studies on post-buckling response of flat panels stiffened by curvilinear stiffeners and have also performed optimization on these panels. It has been shown that curved stiffeners have the capability to improve the structural stability by redistributing the
Tooren and Krakers [9] built a Multi-Disciplinary Optimization (MDO) framework involving mechanical, acoustic and thermal analysis of stiffened and unstiffened simplified fuselage sections. The concepts of multi-bubble and Y braced box fuselage configuration were presented by Mukhopadhyay [10]. It has been shown that multi-bubble geometric configurations are efficient in distributing the stress due to internal pressure load. However, Y braced box fuselage could be a practical alternative while considering the manufacturing constraints. Bouazizi, Lazghab and Soula [11] proposed a new stringer design concept for fuselage where the panels from the conventional fuselage are replaced with the panels stiffened with optimal hexagonal grid with frames. They presented case studies where in an increase in the eigen-frequencies and a decrease in the radial displacement responses were demonstrated.

Over the last decade, several optimization methods have been proposed and frameworks have been developed for optimization of stiffened shells. Lagaros, Fragiadakis and Papadrakakis [12] made use of evolutionary algorithms to optimize design of stiffened shell structures with straight stiffening beams. Hao et al. [13] proposed an efficient optimization framework of cylindrical stiffened shells with reinforced cutouts by curvilinear stiffeners. They developed an adaptive method to find the field near the cut-out and define curvilinear stiffeners in that region and straight stiffeners away from it. They also used Numerical Implementation of Asymptotic Homogenization (NIAH) Method to reduce the computational effort during the optimization. Later, they presented [14] a fast procedure for non-uniform optimum design of stiffened shells under buckling constraint. The authors presented an efficient equivalent analysis model of stiffened shells based on the NIAH Method. However, with curvilinear stiffeners, it is difficult to implement the homogenization method. Wang et al. [15] developed a multi-level optimization framework for hierarchical stiffened shells accelerated by adaptive equivalent strategy to improve the post-buckling optimization efficiency.
2.2. Introduction

The current research work has been implemented with the motivation to study the benefit of arbitrarily-placed curvilinear stiffeners on the shell structures. With the use of more number of arbitrarily-placed stiffeners, design space becomes very large and thus it takes significant wall-clock time to find the optimal design. Therefore, parallel processing is utilized to save extensive wall-clock time. In this chapter, an integrated optimization framework, utilizing the scripting language Python, NURBS based Rhinoceros [16], MSC. PATRAN and MSC.NASTRAN, has been developed in the form of a computational design environment to design curvilinearly stiffened shells. The design has been implemented for finding an optimal design of cylindrical shells stiffened using arbitrarily-placed stiffeners. Stress and buckling analysis have been carried out for satisfying associated buckling and stress constraints with an objective to minimize the weight. Rhinoceros 3D is used for generating geometry including the surface of the shell and the placement of the stiffeners using control points. MSC.PATRAN has been used to generate mesh and NASTRAN input files. Finally, the input file is passed on to MSC.NASTRAN to perform the finite element analysis. An efficient scripting language, Python, has been used to automate the entire process for carrying out the optimization. In Section 2.3, the methodology for performing the optimization, using the integrated approach employing different software, and the parameterization for stiffener placement on the shell surface have been presented. In Section 2.4, the methodology, explained in Section 2.3, has been applied to several cylindrical shell examples with different stiffener configurations. Arbitrarily-placed stiffeners under shear and compression loads have been considered. Finally, in Section IV, conclusions are presented.
2.3 Methodology

The present work is focused on the integration of the versatile capabilities of various commercially available software using a scripting-based language, Python, with the motivation to conduct an optimization of cylindrical curved panels, stiffened by curvilinear stiffeners, along with a parametric study conducted over a range of parameters associated with stiffeners’ placement and stiffeners’ geometric curvature subject to both buckling and stress constraints. A hybrid optimization including the shape optimization and the size optimization has been considered for optimal stiffener configurations for the panels that can be used for fuselage and space launch vehicles designs.

2.3.1 Procedure

In this chapter, NURBS (Non-Uniform Rational B-Splines) have been used to represent the geometry. The software Rhinoceros [16], a commercial available geometric modeler, has been used to generate NURBS based CAD model for the curvilinearly stiffened panels. The geometry is then exported in a parasolid format to the MSC.PATRAN. A session file, using PATRAN Command Language (PCL), can be developed using this software to generate the finite element model for the parameterized geometry of curvilinearly stiffened shell. The mesh is generated in this software along with a definition of the constraints and the applied loads. Finally, the input file is passed on to MSC.NASTRAN to carry out the finite element analysis. This complete process is controlled using a Python script. The optimization responses including objective function and constraints are transferred to the optimizer for structural optimization. The flowchart of the complete procedure is shown in Fig. 2.1 and the details about this procedure are discussed in this section. The overall objective of the optimization is to minimize the weight of the structure. In each iteration, two sub-
optimizations are carried out. First, the size design variables of the structure are kept fixed and only a shape optimization (Particle Swarm Optimization (PSO)) is carried out with an objective to maximize the buckling load without any constraints. During this optimization, only the stiffener shape design variables are modified. Subsequently, this shape-optimized design is fixed during the size optimization (Gradient Based Optimization (GBO)) while optimizing the size design variables, under both buckling and stress constraints. In other words, during the second optimization, shape of the stiffeners is kept fixed and the stiffener and panel size design variables are modified. The objective of the second sub-optimization is to minimize the weight while considering buckling and stress constraints. Size design variables are once again fed to the PSO for shape optimization. This back and forth iteration between the shape and size variables continues until the best possible weight is achieved when these iterations converge. The benefit of using this two-step hybrid optimization is that it divides the total design variables into two groups and each optimization has its own design variables. This approach helps in getting robust optimal results \[17\]. The optimization starts with an initial set of user-defined values \(X_{\text{size}}\) for the cross-section of the stiffeners and panel thickness and a very large value for the best value of weight, \(W_{\text{best}}\). These initial values along with the range for the design variables for defining stiffeners’ locations are passed on to Particle Swarm Optimizer (PSO) to find an optimal shape design that has the maximum possible buckling load. Later on, this design is submitted to MSC.NASTRAN SOL 200, a gradient based optimizer (GBO), for minimizing the weight by designing stiffeners’ cross-section dimensions and the panel thickness under required constraints. The GBO uses finite difference method to compute the sensitivity of weight, buckling load and stress with respect to size design variables. Finally, the weight \(W_{\text{iter}}\) after performing GBO is compared with the weight \(W_{\text{best}}\) as shown in the Fig. 2.1. If the convergence is achieved, the optimization is stopped. Otherwise, the optimized stiffeners’ cross-section and panel thickness are again passed on to the PSO to repeat the optimization process until \(W_{\text{best}}\) is converged.
The advantage of using PSO is that it allows a global search approach and allows for different topologies to be explored. Instead of presented two-step optimization process, one could use an all-in-one approach either by using PSO or GBO. Using PSO in an all-in-one approach would be computationally inefficient as a larger number of particles would be required if one increases the number of design variables. This would significantly increase the CPU-wall clock time. On the other hand, one could use GBO for all-in-one approach by using Finite Difference (FD) to compute all the numerical gradients. However, the optimization of stiffeners’ shape design variables is non-convex and the probability of GBO getting stuck in a local minima is very high. This is not the case for size design variables and thus, one can use GBO for them with some assurance that the optimizer will not be stuck in a local optima. Therefore, in the presented approach, it was decided to separate the design variables for PSO and GBO based on this criterion, resulting into a hybrid optimization technique.

**Geometric Modeling**

Rhinoceros [16] is used to model the geometry in this study. It can create, edit and analyze NURBS curves, surfaces, and solids. *Rhino.Python* is a feature available in the software, wherein a Python script can be written to automate various tasks such as modeling a geometry multiple times with a different set of parameters and perform calculations. This capability helps in changing the dimensions of the geometry, stiffeners’ placement and stiffeners’ geometric curvature during the optimization process. In Rhinoceros, there are more than 50 possible formats which can be used to export CAD model for subsequent use in other software. The geometry can also be exported directly for 3D printing or additive manufacturing.
2.3. Methodology

Shape Optimization

In this framework, Particle Swarm Optimization (PSO) [18] is utilized to maximize the buckling load of the studied model. The objective, as shown in Eq. 2.1, of the optimization is to minimize, $F(X)$, for a certain range of only stiffener shape design variables, $X$, as shown in Eq. 2.1 under no other constraint.

\[
\text{minimize } F(X) = \frac{1}{\text{Buckling factor}} = \frac{\text{Applied Load}}{\text{Buckling Load}}
\]

subject to : \( a_h \leq X_h \leq b_h, \quad h = 1, 2, \ldots, n \)

(2.1)
In this method, a set of random particles (known as designs in our case) are defined over the full domain and the objective function is evaluated for every particle. The design variables of these particles are improved based upon their own and social differences \[4\], meaning based on a particle’s own best value and the global best value in the swarm. Improvement in the particles for obtaining the best objective function is achieved by updating the direction of individual particle during optimization. During an iteration, evaluation of objective functions for these particles can be evaluated in parallel and thus saving computational time. After each iteration, new values for the design variables are calculated using Eqs. 2.2 and 2.3 for all particles. Design variables, called the positions of the particles, are denoted as \(x^i_k\), as shown in Eqs. 2.2 and 2.3. The velocity vector, \(v^i_k\), in these equations, is the velocity of the particle and \(w\) is the weighting parameter for velocity vector. Finally, \(c_1\) and \(c_2\) are the acceleration parameters. A suitable selection of the weighting parameter, \(w\), can provide a balance between local and global search, and result in less iterations on an average for finding an optimal solution \[19\]. Bansal et al. \[20\] tested different inertia weight strategies in PSO. They concluded that the use of constant inertia weight (0.7) produces results near the optimum as compared to the other methods. Thus, the value of \(w\) has been set to 0.7 in this framework. The best particle position and best swarm position are denoted by \(p^i\) and \(p^g_k\), respectively. The time step \(\Delta t\) is mostly taken as unity in these equations. Parameters \(r_1\) and \(r_2\) are random numbers uniformly varying from 0 to 1. The random numbers are used in PSO to initialize a population of particles with uniformly distributed random positions and velocities in the design space \[19\].

\[
v^i_{k+1} = w v^i_k + c_1 r_1 \frac{(p^i - x^i_k)}{\Delta t} + c_2 r_2 \frac{(p^g_k - x^i_k)}{\Delta t} \quad (2.2)
\]

\[
x^i_{k+1} = x^i_k + v^i_{k+1} \Delta t \quad (2.3)
\]
2.3. Methodology

Figure 2.2 shows the local PSO iteration flowchart for carrying out the optimization. As stated above, initial $X_{size}$ values and range for the stiffeners’ placement are provided to PSO. It generates its own initial particles, depending upon the prescribed range, for the first iteration. Firstly, Rhinoceros is called to generate CAD models for all the $N$ particles in a generation. A Python script is run through the various required features of Rhino.Python. The script includes all the decisions that are needed to determine whether the geometry is reasonable, meaning if all the stiffeners are on the surface of the panel or not (details have been mentioned in the next section). Next, if the stiffeners and the panel geometry have been successfully generated, the models are exported in parasolid format for use in MSC.PATRAN. After this, parallel processing is utilized to generate mesh and run buckling
analyses. For each particle of PSO, MSC.PATRAN is used for generating a mesh, defining external loads and boundary conditions through a session file. An input file is generated through MSC.PATRAN for obtaining the buckling load using finite element analysis in MSC.NASTRAN. After all the analyses are completed, the resulting objective functions are stored. A penalty is imposed on the objective function of the particles that failed due to any meshing failure issues. Finally, the new values for the design variables are calculated using Eqs. 2.2 and 2.3 for all the particles and the process is repeated. The optimization is stopped once the swarm best particle value is converged. This way, an optimal shape design is achieved, which is later on passed on to MSC.NASTRAN to run SOL 200 (GBO) for weight minimization with respect to size design variables as shown in Fig. 2.1.

Running the buckling analyses of the swarm particles sequentially at each PSO iteration could take significant wall time. The fact that these particles are independent allows for the use of parallel computation at each iteration and saved, therefore, extensive wall-clock time. Such improvement was possible due to the availability of 100 MSC.PATRAN and MSC.NASTRAN licenses at Virginia Tech. At each iteration, the main process (also referred to as "parent process"), responsible for running PSO prepares a new set of particles, spawns the required number of child processes and splits the list of tasks among those processes. These assigned jobs include mesh generation using MSC.PATRAN and finite element buckling analysis using MSC.NASTRAN for each particle. Once all these child processes are done with their analyses, the main process collects and processes the returned results and continues with the next shape optimization iteration if the convergence criterion is unsatisfied.
2.3. Methodology

Size Optimization

In the present work, a Gradient-Based Optimizer (GBO), MSC.NASTRAN SOL 200, is used for the size optimization of the structure. During this optimization, panel thickness \( t_s \) and geometric dimensions of the stiffener are optimized with the objective to minimize the weight of the structure. The GBO uses finite difference method for finding the sensitivities of the size design variables. The objective, as shown in Eq. 2.4, of the optimization is to minimize, \( G(Y) \), mass, for a certain range of stiffener and panel size design variables, \( Y \), as shown in Eq. 2.4, under buckling and stress constraint.

\[
\begin{align*}
\text{minimize} & : G(Y) = \text{Mass} \\
\text{subject to} & : c_h \leq Y_h \leq d_h, \quad h = 1, 2, ..., m \\
& : \text{Buckling Factor} \geq 1 \\
& : \text{von Mises stress} (\sigma) \leq \text{yield stress} (\sigma_y)
\end{align*}
\]  

(2.4)

2.3.2 Parameterization of stiffener placement and cross-section

In this framework, a stiffener’s initial placement is defined in a base 2D plane and later on, it is projected on to the curved panel. To parameterize the stiffeners’ placement, two parameters have been defined \([5, 7]\); \( \epsilon \), the perimeter parameter, and \( \alpha \), the curvature parameter, as shown in Fig. 2.3. The parameter, \( \epsilon \), is defined on the outer boundary of the 2D base plane and varies from 0 to 1. The placements of the start and the end points (points A and B) for each stiffener are parameterized by the perimeter parameters \( \epsilon_A \) and \( \epsilon_B \). The third control point, \( C \), which governs the stiffener shape, is defined using the second parameter, \( \alpha \). To
use a few shape design variables to represent broader design space for stiffener shape, this parameter is assumed to move in the perpendicular direction from the middle point (Point D) of the straight line joining A and B. In future, more flexibility on the parameter $\alpha$ could be added. The parameter $\alpha$ is non-dimensionalized with panel edge length, $L$. Therefore, each stiffener shape is parameterized by three design variables. The parameterization has been implemented in a Python script that is run using Rhino:Python.

NURBS have the so-called convex hull property, which states that the resulting curve always lies entirely within the convex hull of the control points, meaning the curve always lies within the control polygon as shown in Fig. 2.3. During optimization, PSO might produce a NURBS curve where the third control point (Point C in Fig. 2.3) lies outside of the defined 2D base plane. Therefore, its projection on the surface of the curved panel can fail. A geometry check has been added in the Python script of Rhino:Python the checks whether the third control point is on the required surface or not, and further makes a decision whether the iteration should continue or not. If the third control point is not on the required surface, then a penalty is imposed on the objective function so as to reject that particular design. This method of parameterization can be implemented on any number of required stiffeners.

In the present work, four arbitrarily-placed stiffeners are considered. A total of 12 shape design variables ($X_{\text{shape}}$) are used: three for each stiffener’s placement, four ($X_{\text{size}}$) for describing their cross-section and one for the panel thickness. In this approach, all the stiffeners are placed independently of each other. The range for stiffeners’ placement has been shown in Table 3.1. Here, $X_{ij}$ is the $j^{th}$ control point of the $i^{th}$ stiffener. Both the first and the second control point for any of the stiffeners remain on the boundary of the surface, while the third control point defines the curvature.

A Z-sectioned cross-sectioned is considered in the present work. Figure 2.4 shows different
parameters defined for parameterizing the cross-section. These parameters include $H_2$, $H_1$, $W$, $t$ and $t_f$. The parameter $t$ is the thickness of the flange, $t_f$ is the thickness of the upper and lower flanges, $W$ is the width of the flanges, and $H_1$ and $H_2$ are the inner and outer heights of the web. The parameter $t_f$ can be expressed as $(H_2 - H_1)/2$. Therefore, only four design variables ($H_2$, $H_1$, $W$, $t$) are used while designing the cross-section of the stiffener. The stiffener is modeled as beam elements so that all dimensions for the cross section can be optimized using MSC NASTRAN SOL 200.

Figure 2.4: Design variables for defining cross-section of Z-stringer
Table 2.1: Range of the design variables (Arbitrarily-Placed Case) (See Fig. 2.3)

<table>
<thead>
<tr>
<th>Design Variable</th>
<th>Lower Range</th>
<th>Upper Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stiffener 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{11}$ (A)</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>$X_{12}$ (B)</td>
<td>0.25</td>
<td>1.0</td>
</tr>
<tr>
<td>$X_{13}$ ($\alpha_1$)</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Stiffener 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{21}$ (C)</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>$X_{22}$ (D)</td>
<td>0.5</td>
<td>1.25</td>
</tr>
<tr>
<td>$X_{23}$ ($\alpha_2$)</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Stiffener 3</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{31}$ (E)</td>
<td>0.5</td>
<td>0.75</td>
</tr>
<tr>
<td>$X_{32}$ (F)</td>
<td>0.75</td>
<td>1.5</td>
</tr>
<tr>
<td>$X_{33}$ ($\alpha_3$)</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Stiffener 4</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{41}$ (G)</td>
<td>0.75</td>
<td>1.0</td>
</tr>
<tr>
<td>$X_{42}$ (H)</td>
<td>0</td>
<td>0.75</td>
</tr>
<tr>
<td>$X_{43}$ ($\alpha_4$)</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

2.3.3 Problem description

In this chapter, a cylindrical panel structure has been modeled with $Z$ stiffeners placed on its surface with the motivation to represent fuselage stiffened skin or a space launch vehicle’s stiffened skin with stiffeners. A fuselage experiences axial compression in the skin due to the bending loads acting on it. This compression can cause the fuselage skin to buckle. Also, there are shear loads acting on a fuselage panel which too must be considered. In Fig. 2.5 (a), the boundary conditions considered in this chapter are shown. A cylindrical coordinate system is used while implementing the methodology. Four load cases, shown in Table 2.2, have been considered. These values have been taken from our previous work \[4\]. The fuselage has to be designed under all these loading conditions along with the objective to minimize the total weight subjected to yield strength and buckling constraints. Both the dimensions of the model and the properties of the material used are shown in Table 2.4. The values of
Figure 2.5: Schematic figure of curved panel panel (skin); (a) boundary conditions. (b) three different loads

the dimensions of Z stringers (Fig. 2.4) in Table 2.4 are just initial values for starting the optimization process.

<table>
<thead>
<tr>
<th>Case</th>
<th>$N_{zz}$ (kN/m)</th>
<th>$N_{\theta\theta}$ (kN/m)</th>
<th>$N_{\theta z}$ (kN/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shear Dominated (L1)</td>
<td>93.52</td>
<td>1.58</td>
<td>116.46</td>
</tr>
<tr>
<td>Shear Dominated (L2)</td>
<td>93.52</td>
<td>1.58</td>
<td>-116.46</td>
</tr>
<tr>
<td>Compression Dominated (L3)</td>
<td>136.78</td>
<td>11.21</td>
<td>93.70</td>
</tr>
<tr>
<td>Compression Dominated (L4)</td>
<td>136.78</td>
<td>11.21</td>
<td>-93.70</td>
</tr>
</tbody>
</table>

2.4 Application and Results

In this section, the methodology explained in Section 2.3, has been applied to several curved panel examples under different load cases. Arbitrarily-placed stiffeners are considered in the
present studies. A total of 240 particles in one swarm have been considered for PSO local iterations.

A stress factor $S_{FS}$, defined as a ratio of the maximum von Mises stress in the skin to the yield stress ($\sigma_y$), is found for all the examples in this section. The ratio can be helpful to see how far away the maximum von Mises stress is from the yield stress for any configuration. The stress factor should be $\leq 1$ for all load cases. Also, a stress factor $S_{FS_t}$ is defined as a ratio of maximum stress in the stiffeners to the yield stress. The $S_{FS_t}$ should also be $\leq 1$ for all the load cases.

The optimization is performed on a windows machine with Intel® Xeon® Processor E5-2687W v4 (3.00 GHz and 64.0 GB RAM). The Python version of 2.7.13 is used for implementing the optimization. The optimization is conducted using 20 parallel processes in the PSO. For 12 design variables of arbitrarily-placed stiffeners, 240 particles in a generation have been used in the PSO while finding the optimal design with the maximum buckling load. One complete local PSO iteration (Fig. 2.2) takes about 7-10 minutes to complete all the required analyses of one generation when using parallel processing. PSO takes about 40-50 generations of these particles to find the optimal shape design. Without any parallel computation, the optimization for one load case takes around 100 hours of the wall-clock time. However, with the use of parallel processing, that time has been brought down to around 6 hours. This time consumption can be further reduced with the use of more parallel processors.

Table 2.3: Range for design variables of $X_{size}$ (Fig. 2.4)

<table>
<thead>
<tr>
<th></th>
<th>H1 (m)</th>
<th>H2 (m)</th>
<th>W (m)</th>
<th>t (m)</th>
<th>Panel thickness (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower bound</td>
<td>0.008</td>
<td>0.008</td>
<td>0.002</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Upper bound</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.008</td>
</tr>
</tbody>
</table>
Table 2.4: Dimensions and Material Properties of the Model

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material</td>
<td>Al 2139</td>
</tr>
<tr>
<td>Young’s Modulus (E) (GPa)</td>
<td>73.085</td>
</tr>
<tr>
<td>Poisson’s ratio</td>
<td>0.33</td>
</tr>
<tr>
<td>Allowable stress (MPa)</td>
<td>427.47</td>
</tr>
<tr>
<td>Radius of curvature(m)</td>
<td>2.5</td>
</tr>
<tr>
<td>Angle (degrees)</td>
<td>30</td>
</tr>
<tr>
<td>Length of curved ends (m)</td>
<td>1.309</td>
</tr>
<tr>
<td>Length of straight ends (L) (m)</td>
<td>1.294</td>
</tr>
<tr>
<td>$t_s$ (Skin thickness) (m)</td>
<td>0.004</td>
</tr>
<tr>
<td>H2 (Stiffener) (m)</td>
<td>0.04</td>
</tr>
<tr>
<td>H1 (Stiffener) (m)</td>
<td>0.032</td>
</tr>
<tr>
<td>t (Stiffener) (m)</td>
<td>0.004</td>
</tr>
<tr>
<td>W (Stiffener) (m)</td>
<td>0.035</td>
</tr>
</tbody>
</table>

2.4.1 The Shear Dominated Load Case (L1)

In this section, the results from arbitrarily-placed stiffeners under the shear load case (L1) have been presented. Initially, for a comparison, only straight stiffeners are used. The weight has been minimized by including the buckling constraint (Buckling factor $\geq 1$) and the stress constraints ($SF_S \leq 1$ and $SF_{ST} \leq 1$). The results for optimal panel for the shear load case, using equally spaced straight stiffeners, have been presented in Fig. 2.6. The optimized mass for the shear load case with straight stiffeners is 16.384 kg with the buckling factor $= 0.9973$, $SF_S = 0.1599$ and $SF_{ST} = 0.2786$. The buckling constraint is active at the optimal design with equally spaced straight stiffeners. However, the stress constraints are not active. Later, the methodology presented in Section 2.3 is implemented for designing arbitrarily-placed stiffeners. The global iteration history is shown in Fig. 2.7. The figure shows the weight ($W_{Iteration}$) and best weight recorded ($W_{Best}$) at each global iteration. The optimal weight is found after 2$^{nd}$ global iteration. However, it can be seen that the optimizer also found different geometries ($5^{th}$ and $9^{th}$ iterations) having weights close to the optimal.
The oscillations in the weight, $W_{\text{Iteration}}$, at every iteration is due to the randomness in the PSO. The PSO can find different geometries all close to the optimal, every time it is used. Nevertheless, the $W_{\text{Best}}$ converges very well and the optimal weight, $W_{\text{opt}} = 14.61$ kg, is found after the $2^{nd}$ iteration with the buckling factor $= 0.9989$, $SF_S = 0.2002$ and $SF_{ST} = 0.2997$. Figure 2.8 shows the results from the use of arbitrarily-placed stiffeners after $2^{nd}$ global iteration. Again, the buckling constraint is active at the optimal design. It has been seen that there is a 10.8% weight saving by this optimized case as compared to the use of equally placed straight stiffeners. The optimized cross-section values for all approaches under the shear load case are given in Table 2.5.

Figure 2.6: Optimal panel while designing for the shear load case (L1) using equally spaced straight stiffeners.

<table>
<thead>
<tr>
<th>Design case</th>
<th>H1 (m)</th>
<th>H2 (m)</th>
<th>W (m)</th>
<th>t (m)</th>
<th>Panel thickness (m)</th>
<th>Mass (kg)</th>
<th>% saving w.r.t. Straight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Straight (Fig. 2.6)</td>
<td>3.085E-02</td>
<td>3.785E-02</td>
<td>2.000E-03</td>
<td>1.000E-03</td>
<td>3.423E-03</td>
<td>16.384</td>
<td>-</td>
</tr>
<tr>
<td>Arbitrary (Fig. 2.8)</td>
<td>3.489E-02</td>
<td>4.189E-02</td>
<td>3.934E-03</td>
<td>1.000E-03</td>
<td>2.990E-03</td>
<td>14.610</td>
<td>10.8</td>
</tr>
</tbody>
</table>
2.4. Application and Results

Shear History

\[ W_2 = 14.610 \, \text{kg} \]
\[ W_5 = 14.617 \, \text{kg} \]
\[ W_9 = 14.631 \, \text{kg} \]
\[ W_{\text{Straight}} = 16.384 \, \text{kg} \]

Figure 2.7: Global iteration history for the shear load case (L1) using arbitrarily-placed stiffeners

Buckling factor = 0.9989, Stress factor (skin) = 0.2002, Stress factor (stiff) = 0.2997,

Figure 2.8: Optimal panel after 2\textsuperscript{nd} global iteration while designing for the shear load case (L1) using arbitrarily-placed stiffeners.
2.4.2 The Shear Dominated Load Case (L2)

In this section, the results from arbitrarily-placed stiffeners under the shear load case (L2) have been presented. The only change L2 load case has from L1 load case is the direction of the shear loads (See Fig. 2.5 and Table 2.2). Again, for a comparison, straight stiffeners for the panel are used and the weight is minimized under the buckling and stress constraints. The results for the optimal panel for the shear load case, using equally spaced straight stiffeners, have been presented in Fig. 2.9. The optimized mass with the use of equally spaced straight stiffeners is 16.378 kg with the buckling factor = 0.9975, $SF_S = 0.1599$ and $SF_{ST} = 0.2786$. Once again, the buckling constraint is active at the optimal design with equally spaced straight stiffeners. The global iteration history for the optimization of arbitrarily-placed stiffeners is shown in Fig. 2.10. The optimal weight, 14.661 kg, is found at 8th global iteration with the buckling factor = 0.9983, $SF_S = 0.1992$ and $SF_{ST} = 0.3325$. However, it can be seen that the optimizer also found different geometry (5th iteration) having weight close to the optimal geometry. Figure 2.11 shows the results from the use of arbitrarily-placed stiffeners after 8th global iteration. It has been seen that there is a 10.4% weight saving by this optimized case as compared to the use of equally placed straight stiffeners. The optimized cross-section values for all approaches under the shear load case are given in Table 2.6. It can be seen from Figs. 2.8 and 2.11 that the direction of shear loads has an important role in the orientation of optimal stiffener configurations. Although all the loads have the same magnitude in both the load cases L1 and L2 and only the direction of the shear loads is different (Table 2.2), the optimal orientation of the stiffeners is found to be significantly different.
2.4. Application and Results

Figure 2.9: Optimal panel while designing for the shear load case (L2) using equally spaced straight stiffeners.

Figure 2.10: Global iteration history for the shear load case (L2) using arbitrarily-placed stiffeners
Figure 2.11: Optimal panel after 8th global iteration while designing for the shear load case (L2) using arbitrarily-placed stiffeners.

Table 2.6: Optimal final values for $X_{size}$ (Fig. 2.4) for the shear load case (L2)

<table>
<thead>
<tr>
<th>Design case</th>
<th>H1 (m)</th>
<th>H2 (m)</th>
<th>W (m)</th>
<th>t (m)</th>
<th>Panel thickness (m)</th>
<th>Mass (kg)</th>
<th>% saving w.r.t. Straight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Straight</td>
<td>3.086E-02</td>
<td>3.786E-02</td>
<td>2.000E-03</td>
<td>1.000E-03</td>
<td>3.422E-03</td>
<td>16.378</td>
<td>-</td>
</tr>
<tr>
<td>(Fig. 2.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arbitrary</td>
<td>2.983E-02</td>
<td>3.717E-02</td>
<td>3.102E-03</td>
<td>1.000E-03</td>
<td>3.0181E-03</td>
<td>14.661</td>
<td>10.4</td>
</tr>
<tr>
<td>(Fig. 2.11)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.4.3 The Compression Load Case (L3)

In this section, the compression load case (L3) has been investigated. In this load case, the magnitude of the compressive load is higher than the shear loads (See Fig. 2.5 and Table 2.2).

Again, for comparison purposes, straight stiffeners are used. The weight has been minimized subjected to both the buckling and the stress constraints. Figure 2.12 shows the results for the optimal design for the compression load case using equally spaced straight stiffeners.

The optimized mass, by using straight stiffeners, is 17.886 kg with the buckling factor = 0.9971, $SF_S = 0.1793$ and $SF_{ST} = 0.4285$. The global iteration history for the optimization of arbitrarily-placed stiffeners is shown in Fig. 2.13. The optimal weight, 15.520 kg, is found
at 4th global iteration with the buckling factor = 0.9957, $SF_S = 0.2220$ and $SF_{ST} = 0.4871$. Figure 2.14 shows results from the use of the arbitrarily-placed stiffeners approach. It has been seen that there is a 13.2 % weight saving for this optimized case as compared to the equally spaced straight stiffeners case. The optimized cross-sectional values for all approaches under the compression load case are given in Table 2.7. The direction of the shear loads in the load case (L3) is same as that of the load case (L1). It can be seen from Figs. 2.8 and 2.14 that the direction of shear loads has again played an important role in defining the optimal stiffener configuration. Even though the magnitude of the compressive load is higher than the shear loads (Table 2.2), the orientations of the optimal stiffener configurations are very similar for both the load cases L1 and L3. This shows that shear loads have more impact than compressive loads on the optimal stiffener configuration.

![Diagram](image)

**Figure 2.12**: Optimal panel while designing for the compression load case (L3) using equally spaced straight stiffeners.
Figure 2.13: Global iteration history for the compression load case (L3) using arbitrarily-placed stiffeners.

Figure 2.14: Optimal panel after the 4th global iteration while designing for the compression load case (L3) using arbitrarily-placed stiffeners.
2.4. Application and Results

Table 2.7: Optimal final values for $X_{size}$ (Fig. 2.4) for the compression load case (L3)

<table>
<thead>
<tr>
<th>Design case</th>
<th>H1 (m)</th>
<th>H2 (m)</th>
<th>W (m)</th>
<th>t (m)</th>
<th>Panel thickness (m)</th>
<th>Mass (kg)</th>
<th>% saving w.r.t. Straight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Straight (Fig. 2.12)</td>
<td>3.454E-02</td>
<td>4.154E-02</td>
<td>3.500E-03</td>
<td>1.000E-03</td>
<td>3.708E-03</td>
<td>17.886</td>
<td>-</td>
</tr>
<tr>
<td>Arbitrary (Fig. 2.14)</td>
<td>3.701E-02</td>
<td>4.412E-02</td>
<td>3.536E-03</td>
<td>1.000E-03</td>
<td>3.216E-03</td>
<td>15.552</td>
<td>13.2</td>
</tr>
</tbody>
</table>

2.4.4 The Compression Load Case (L4)

In this section, the compression load case (L4) has been implemented. In this load case, the magnitude of the compressive load is higher than the shear loads in comparison to load cases L1 and L2, and the direction of the shear loads is opposite to that of the load case L3 (See Fig. 2.5 and Table 2.2). Once again, for comparison purposes, equally spaced straight stiffeners for the panel are used. Figure 2.15 shows the results for the optimal design using equally spaced straight stiffeners. The optimized mass, by using straight stiffeners, is 17.878 kg with the buckling factor $= 0.9971$, $SF_S = 0.1793$ and $SF_{ST} = 0.4285$. Figure 2.16 shows the global iteration history for the optimization of arbitrarily-placed stiffeners. The optimal weight, 15.553 kg, is found at 2nd global iteration with the buckling factor $= 0.9976$, $SF_S = 0.2388$ and $SF_{ST} = 0.3864$. Figure 2.17 shows results of the optimal design using arbitrarily-placed stiffeners approach. There is a 13% weight saving for this optimized case as compared to the equally spaced straight stiffeners case. The optimized cross-sectional values for all approaches under the compression load case are given in Table 2.8. Once again, it is seen from Figure 2.14 and 2.17 that the direction of shear loads is important in defining the orientation of the optimal stiffener configuration. The optimal stiffener configuration changed significantly by changing the direction of shear loads in the load case L3.
Figure 2.15: Optimal panel while designing for the compression load case (L4) using equally spaced straight stiffeners.

Figure 2.16: Global iteration history for the compression load case (L4) using arbitrarily-placed stiffeners.
2.5 Conclusions

In this chapter, a framework for designing stiffened curved panels with arbitrarily-placed stiffeners is presented. An integrated optimization framework utilizing the scripting language Python, NURBS based Rhinoceros, MSC.PATRAN and MSC.NASTRAN has been developed. It can be seen that stiffeners’ placements and their curvatures have a significant influence on the buckling load of the stiffened panel and this can help to save structural weight along with optimizing the stiffeners’ cross-section dimensions and the panel thickness.
The framework, developed here, involves a hybrid optimization technique of using Particle Swarm Optimization (PSO) for maximizing buckling load by optimizing stiffeners' location on panel and gradient based optimization (MSC.NASTRAN SOL 200) to optimize stiffeners cross-section dimensions and panel thickness for minimizing weight subjected to buckling constraint and yield strength. Four load cases involving shear and compression loads are considered. It is seen that shear loads have an important role in defining optimal stiffener configurations. It is concluded that there is a significant potential for weight saving by using arbitrarily-placed curvilinear stiffeners as compared to the use of conventional equally spaced straight stiffeners.

Parallel processing has been used in the PSO while finding the design with the maximum buckling load. The fact that these particles in PSO are independent, allows one the use of parallel computation during each iteration and, thus, saving extensive wall-clock time. Genetic algorithm based optimization could be another alternative to the use of PSO for shape optimization. Also, the two-step hybrid optimization proved to be highly beneficial in partitioning the total design variables into two groups. This partition helped in optimizing these design variables in a two-step optimization strategy. This method can also be applied in designing other structures problems where shape and size design variables can be divided into two sub-optimization groups and a similar procedure can be followed. Indeed Zhao and Kapania [17] have used such a strategy to great benefit for performing multi-disciplinary design and optimization of a light composite drone.
Chapter 3

Buckling Load Maximization of Curvilinearly Stiffened Tow-Steered Laminates using Parallel Processing

3.1 Abstract

Under certain loading conditions, curvilinearly stiffened panels are seen to have better structural performance while being lighter in mass when compared to panels with straight stiffeners. A curvilinearly stiffened tow-steered composite laminate could be manufactured by fastening or bonding metallic curvilinear stiffeners to the composite panel. In this chapter, such panels are computationally evaluated for their benefit in improving the panel buckling load. Initially, a parallel processing based optimization framework to design tow-steered composite laminates with metallic curvilinear stiffeners is presented, where the objective is to maximize the buckling load with only structural mass constraint and no stress constraint. Results show that the curvilinear stiffeners and curvilinear fiber paths in the composite laminated skin, can lead to more than 75% increase in the panel buckling load as compared to the use of equidistant straight stiffeners and straight fibers. Subsequently, a multi-objective optimization is conducted with the objectives to maximize buckling load, while minimizing the mass of the panels. It is seen that the improvement in the buckling load, by using curvi-
linear stiffeners and curvilinear fibers over equidistant straight stiffeners and straight fibers, depends upon the prescribed structural mass constraint.

### 3.2 Nomenclature

- \( B \) = Width of panel (m)
- \( L \) = Length of panel (m)
- \( T_0 \) = Fiber angle at middle \((x = 0)\) (deg)
- \( T_1 \) = Fiber angles at panel edge \((x = \pm B/2)\) (deg)
- \( X_{\text{shape}} \) = Shape design variables
- \( X_{\text{size}} \) = Size design variables
- \( X^s \) = Stiffener shape design variables (subset of \( X_{\text{shape}} \))
- \( X_{ij}^s \) = \( j^{th} \) control point of the \( i^{th} \) stiffener
- \( X_{\text{c}} \) = Fiber path shape design variables (subset of \( X_{\text{shape}} \))
- \( X_{\text{c}}^c \) = Curved fiber path shape design variables
- \( X_{\text{s}}^c \) = Straight fiber path shape design variables
- \( h_s \) = Height of stiffeners (m)
- \( t_s \) = Thickness of stiffener (m)
- \( t_l \) = Thickness of each layer in composite laminate (m)
- \( N \) = Number of particles in a swarm in PSO
- \( m \) = Maximum allowed generations in PSO
- \( p \) = Maximum number of machines running in parallel
- \( n \) = Maximum number of jobs running in each of the \( p \) machines
- \( \epsilon \) = Perimeter parameter on outer boundary of panel
- \( \alpha \) = Curvature parameter of a stiffener
$F_x, F_y$ = Bi-axial compressive loads on panel edges in $x$ and $y$ directions (kN/m)

$S_f$ = Shear load on panel edges (kN/m)

$K$ = Penalty factor ($10^6$)

### 3.3 Introduction

Composite materials are increasingly being used in the aerospace industry because of their advantage of being lighter while having more strength per unit weight than metals. The conventional method to manufacture composite structures involves using straight fibers. However, with advancements in composite fabrication techniques [22] like Automated Fiber Placement (AFP), it is now possible to manufacture Variable-Stiffness Composite Laminated (VSCL) panels. The VSCL panels have the advantage of providing the freedom to manipulate the properties in directions that are favorable for carrying loads with the minimum possible structural weight. The optimal design of tow-steered composite laminate panels with minimum possible weight [23] can be found by finding the best arrangement of curvilinear fibers within the structure. It has been demonstrated in the literature that the curvilinear fiber paths have the potential to improve the structural performance of the composite structure as compared to the straight-fiber laminates. Nik et al. [24] conducted simultaneous optimization of stiffness and buckling load of a composite laminated plate with curvilinear fiber paths. They showed results where curvilinear fiber paths can increase both the buckling load and stiffness of a laminate. Seetodeh et al. [25] showed that significant improvements in buckling load are possible by varying the stiffness properties spatially. Stodieck et al. [26] optimized tow-steered composite wing laminates for aeroelastic tailoring. The authors have shown results where aircraft performance can be significantly improved by using curvilinear fibers in comparison to optimized straight-fiber laminates. Gürdal, Tatting
and Wu [27] demonstrated that the variable stiffness concept in VSCL panels gives flexibility to the designer for trade-offs between overall panel stiffness and the buckling load. Lopes, Camanho and Gürdal [28] demonstrated the advantages of variable-stiffness laminates over straight-fiber laminates in terms of compressive buckling and first-ply failure. Murugan and Friswell [29] studied composites with curvilinear fiber paths and showed that curved fiber paths can minimize the in-plane stiffness and increase the bending stiffness simultaneously when compared to a baseline plate with straight fibers.

The additive manufacturing (AM) has made it possible to manufacture curvilinearly stiffened panels as a single unitized structure. These panels have been proven to have better structural performance while being lighter in weight, in comparison to panels with straight stiffeners under certain loading conditions. The conventional way to use stiffeners in the fuselage and wings of an aircraft is to use straight stiffeners. The design space can be enhanced by using curved stiffening members. Kapania, Li and Kapoor [5] have shown that curvilinear stiffened panels can lead to lighter weight designs in comparison to the panels with straight stiffeners under certain design loads. Singh, Zhao and Kapania [1] presented an optimization framework for finding an optimal design of cylindrical shells stiffened using curvilinearly-placed stiffeners under different load cases. It has been shown that the use of curvilinear stiffeners can provide lighter weight designs as compared to equally-spaced straight stiffeners. Zhao and Kapania [7] showed that an improvement in the buckling load is possible by using curvilinear stiffeners. Wang, Abdalla and Zhang [30] showed that grid-stiffened composite structures using curvilinear stiffeners are not only useful for structural weight reduction, but also for improved structural stability and damage tolerance in comparison with the use of conventional straight stiffened panels. Hao et al. [31] stated that the use of curvilinear stiffeners leads to an improved load path as well as a large increase in the collapse load of the structure in comparison to a panel with straight stiffeners.
Stanford and Jutte [32] studied the combination of tow-steered composite laminate skins and curvilinear stiffeners for aeroelastic optimization problems on a high-aspect-ratio wing-box of the Common Research Model. They showed that the combination of curvilinear stiffeners and tow steering laminates can provide significant mass reduction. They also stated that the metallic curvilinear stiffeners can be fastened to the VSCL panels. The current research work is motivated by the idea of computationally evaluating the advantages of using panels with curvilinear fibers and curvilinear stiffeners for improving buckling loads. In the future, experimental studies can be conducted to validate their benefits.

In this chapter, first, a single objective based optimization framework has been developed in the form of a computational design environment to design composite laminates. The optimization has been implemented for finding an optimal design of a composite laminate using curvilinear metallic stiffeners and curvilinear fiber paths in the composite skin. With the use of more curvilinear stiffeners along with independent curved fiber paths in each layer of the laminate, the design space becomes very large and thus it takes extensive computer runtime to find the optimal design. Therefore, parallel processing is implemented to accelerate the optimization. The optimization is conducted with the objective of finding the maximum possible buckling load with the constraint of an upper limit on structural mass. In this chapter, buckling response of the panels in the presence of simplified loading conditions is studied, the motivation being that it could help a designer to understand how to exploit features like stiffener and fibers placement and their curvature.

For generating the geometry of the panels, the NURBS-based CAD modeling software Rhinoceros 3D [16] is used. MSC.PATRAN has been used to generate the finite element model for buckling analysis conducted in MSC.NASTRAN. Glue contact (IGLUE), a built-in capability of MSC.NASTRAN, is used to satisfy equilibrium and compatibility conditions at the stiffener-skin interfaces. The advantage of using the IGLUE capability is that a curvi-
linear stiffener can be placed on the surface of the skin without the restriction of having to place common nodes at the stiffener-skin interfaces. This is relevant because attempts to directly create a mesh from the imported geometry from Rhinoceros 3D in MSC.PATRAN result failed to identify coincidence of the curvilinear stiffener and VSCL panel skin. In Section 3.4, the methodology for implementing the optimization framework is presented. In Section 3.5, a composite laminated panel stiffened by four metallic curvilinear stiffeners is subjected to different load cases, and the results are presented. There are many load cases and boundary conditions beyond the scope of the presented research for optimizing a stiffened panel. The authors have selected certain boundary condition and load cases that an aircraft wing could experience. These include compression dominated, shear dominated and pure shear load cases for optimizing the panel. It is seen that the curvilinear stiffeners along with the use of optimal curvilinear fiber paths in composite laminated skin can lead to a significant increase in the panel buckling load while having the same mass as compared to the use of equidistant straight stiffeners and straight fibers under pure load case. This improvement in the buckling load motivated the authors to also conduct a multi-objective Non-dominated Sorting Genetic Algorithm-II (NSGA-II) based optimization with the objectives to maximize buckling load, while minimizing the mass of the panels under pure shear load case. It is shown that the improvement in the buckling load, by using curvilinear stiffeners and curvilinear fibers over equidistant straight stiffeners and straight fibers, could change based on the required structural mass constraint.

3.4 Methodology

In this chapter, the versatile capabilities of various commercially available software have been integrated [1] using the scripting-based language Python. The Non-Uniform Rational
3.4. Methodology

B-Splines (NURBS)-based commercially available CAD modeling software *Rhinoceros 3D* is used for generating the geometry. *MSC.PATRAN* is used for generating the mesh and defining the boundary conditions. Finally, *MSC.NASTRAN* is used for conducting buckling analyses. Figure 3.1 shows the complete optimization procedure used in this study. The details about the procedure are explained in this section.

### 3.4.1 PSO Based Single Objective Optimization Procedure

The optimization program starts with a user-defined range of design variables and a structural mass constraint with the objective of maximizing the buckling load of the panel. Particle Swarm Optimization (PSO) \[1\] is used as the optimizer in this chapter, but with a double level parallel processing. The objective of the optimization, as shown in Eq. 3.1, is the reciprocal of the buckling factor of the structure. The buckling factor, evaluated using *MSC.NASTRAN* buckling analysis, is defined as the ratio of the buckling load to the applied load. The optimization is subject to a structural mass constraint along with upper and lower bounds on the design variables as shown in Eq. 3.1 and 3.2. In the present optimization, no stress constraint is considered.

**Problem Statement**

The objective of the optimization is to minimize, $F(\mathbf{X})$, for a certain range of $\mathbf{X}$, as shown in Eq. 3.1 and a user defined structural mass constraint, $M_0$. 
Shape Optimization (PSO)

**Objective:** Maximize Buckling Load

**Particles** = \( N \)

**Max generations** = \( m \)

**CAD model generation for all particles (Rhinoceros 3D)**

**Parallel Processing**

**Mesh (MSC.PATRAN)**

**MSC.NASTRAN (SOL 105)**

**Analysis**

**For Each Particle**

**Particle 1 Analysis**

**Particle 2 Analysis**

**Particle \( N \) Analysis**

**Stop Criteria:**
Swarm best objective Converged?

**No**

**Update particles’ locations**

**Yes**

**Optimal Design**

Figure 3.1: Flowchart of the procedure
3.4. Methodology

\( \min_{\mathbf{X}}: F(\mathbf{X}) = \frac{1}{\text{Buckling factor}} = \frac{\text{Applied Load}}{\text{Buckling Load}} \)

subject to: \( a_h \leq X_h \leq b_h, \quad h = 1, 2, \ldots, n \)

\[ \frac{M(\mathbf{X})}{M_0} - 1 \leq 0 \quad \text{or} \quad g(\mathbf{X}) \leq 0 \]  

\( (3.1) \)

Penalty Method

In the present work, the penalty method has been used to convert the constrained optimization problem into an unconstrained optimization problem. Therefore, the objective of the optimization problem becomes:

\( \min_{\mathbf{X}}: (\mathbf{X}) = F(\mathbf{X}) + K(\max(0, g(\mathbf{X})))^2 \)

subject to: \( a_h \leq X_h \leq b_h, \quad h = 1, 2, \ldots, n \)  

\( (3.2) \)

where, \( F(\mathbf{X}) \) is the objective function, as shown in Eq. \(3.1\), and \( K \) is a very large number, e.g. \(10^6\), taken in this chapter. The PSO is used to minimize \( \Phi(\mathbf{X}) \) as shown in Eq. \(3.2\). In PSO, a set of random particles (designs in our case) are defined over the full range of design variables. In Fig. \(3.1\), \( N \) denotes the maximum number of particles and \( m \), the maximum allowed generations for \( N \) particles. After each generation, the positions of the design variables of the particles are updated based upon their own best value and the swarm’s best value. For every particle in the swarm, buckling factor and constraint values are evaluated in order to find the unconstrained objective function Eq. \(3.2\). The optimization is stopped if either the maximum number of generations has been generated or the change in the global best value
in the consecutive generations is less than 0.01% for the user-defined number of generations. More details regarding PSO are available in Ref. [18].

3.4.2 NSGA-II Multi-Objective Optimization

In this chapter, a multi-objective optimization is also conducted for maximizing the buckling load while minimizing the mass of the composite laminate with curvilinear stiffeners. The results are presented in Section 3.5. The Non-dominated Sorting Genetic Algorithm-II (NSGA-II), used in this chapter, is a multi-objective optimization algorithm that was developed by Deb et al. [33] to alleviate issues of original NSGA: high computational complexity of non-dominated sorting and lack of elitism. More details regarding this are available in Ref. [33].

\[
\text{minimize } \quad \left( \frac{1}{\text{Buckling factor}}, \text{Mass} \right) \\
\text{subject to } \quad a_h \leq X_h \leq b_h, \quad h = 1, 2, \ldots, n \quad (3.3)
\]

3.4.3 Parallel Processing

Performing the buckling analyses of the swarm of different particles serially takes significant computer run-time. Thus, parallel processing is utilized to save time for running the analyses. In Fig. 3.1, it can be seen that the function evaluations run independently during any PSO or NSGA-II iteration. Each function evaluation runs MSC.PATRAN and MSC.NASTRAN for mesh generation and running the buckling analysis. The fact that these function evaluations are independent allows for the use of parallel computation at each iteration and results in
3.4. Methodology

significantly less computer run-time. Figure 3.2 shows the flowchart of the developed double-level parallel processing. At each PSO or NSGA-II iteration, the main process (Parent Process), responsible for running optimization prepares a new set of \( N \) generations, spawns the required number of child processes and splits various tasks among different machines. Each machine works independently and runs the assigned jobs in parallel. Once all these child processes have been analysed, the main process collects and processes the returned results and continues with the next PSO or NSGA-II iteration if the convergence criterion is unsatisfied.

![Flowchart of the double-level parallel processing](image)

Figure 3.2: Flowchart of the double-level parallel processing

### 3.4.4 Glue Contact Capability

In the current work, a built-in capability of MSC.NASTRAN, Glue Contact, has been used to define contact between the surface of the plate and the stiffeners to satisfy the compatibility and equilibrium conditions. The approach has the advantage of placing stiffeners arbitrarily on the plate. This removes the requirement for nodes to be coincident at stiffener-shell interfaces. The technology is applied as linear contact where two bodies (Stiffener and Plate)
remain in contact in any condition. Verification study and more details are provided in [34]. This is relevant because attempts to directly mesh imported geometry from Rhinoceros 3D in MSC.PATRAN result in failure to identify coincidence of the curvilinear stiffener and VSCL panel skin. Figure 3.3 shows an example case where stiffener location could be independent of the panel mesh.

![Figure 3.3: Benefit of Glue Contact: The stiffener (shown in red) and panel mesh do not need to coincide at the interface](image)

3.4.5 Parameterization

In the present work, the design variables ($X$) are categorized as: shape ($X_{\text{shape}}$) and size ($X_{\text{size}}$) design variables. The ($X_{\text{shape}}$) are further sub-categorized into the stiffener shape ($X^s$) and fiber path design variables ($X^e$) in a composite laminate. The size design variables
include the thickness of the laminated plate and cross-sectional dimensions of the stiffener.

**Shape Design Variables**

**Curvilinear Fibers:** The fiber path could be defined in different ways, but, due to the manufacturing constraints [29], many studies on curved fiber in the literature have treated the fiber paths to be a linear, 1-D variation, of a reference fiber path as originally suggested by Gürdal and Olmedo [35]. In the current chapter, a similar parameterization is defined. Along the direction $x$, fiber orientation $T(x)$ can be defined as:

$$T(x) = 2(T_1 - T_0) \left| \frac{x}{B} \right| + T_0$$ \hspace{1cm} (3.4)

where, $B$ is the width of the plate, and $T_0$ is the fiber angle at middle ($x = 0$) and $T_1$ is the fiber angle at the edges ($x = \pm B/2$) of the plate as shown in Fig. 3.4. In the present work, two design variables: $T_0$ and $T_1$ are used for defining the curvilinear fiber path for each layer of a three-layered composite laminate. This results in 6 curved fiber path design variables ($X^c$). For a straight fiber path in a composite laminates, $T_0$ can be set equal to $T_1$, thus, resulting into 3 straight fiber path design variables ($X^s$) for a three-layered composite laminate. In the optimization problem, the range of the design variables is defined to vary from $-90^\circ$ to $90^\circ$.

The parameterization of curvilinear fiber paths is implemented into the finite element model in *MSC.PATRAN* by dividing the panel along the width. Elements in each division are assumed to have the same fiber angles [29]. For example, in Fig. 3.4(b), the plate is shown having 20 divisions with $T_0 = 0^\circ$ and $T_1 = 60^\circ$. Each division in the figure has constant fiber orientation angle.

**Curvilinear Stiffeners:** Singh, Zhao and Kapania [1] implemented a NURBS-based pa-
Figure 3.4: (a) Parameterization of curved fiber paths (b) $T_0 = 0^\circ$ and $T_1 = 60^\circ$ with 20 divisions along the width (B).

Parameterization for curvilinear stiffeners on a panel using two parameters: $\epsilon$, the perimeter parameter, and $\alpha$, the curvature parameter as shown in Fig. 3.5(a). The perimeter parameter is defined on the outer boundary of the plate. It varies from 0 to 1 over the outer boundary of the panel. If the value of the parameter becomes larger than 1, then value of 1 is subtracted from its value to define the location on the outer boundary. For example, if the value of the parameter is 1.25, then on the panel it would be defined by 0.25. The start and the end control points (A and B) for the stiffener are parameterized by the perimeter parameters $\epsilon_A$ and $\epsilon_B$. A third control point C defines the curvature of the stiffener, and is defined using the second parameter, $\alpha$. This parameter is assumed to move in the perpendicular direction from the middle point (Point D) of the straight line joining A and B. Thus, three design variables are used to parameterize stiffener shape. A python script is written in Rhino.Python [16] to conduct parameterization and generate the geometry.

The parameterization of curvilinear stiffener is more flexible in comparison to the parameterization of the curvilinear fibers as the stiffener has the flexibility to easily stretch from left
to right edges or from top to bottom edges, but the fiber parameterization is slightly biased toward horizontal direction over vertical direction, since the fiber angles can vary along the horizontal but not along the vertical. However, it can be said that this is justifiable in the current study as the applied loads, considered in this chapter, are mostly dominating in the horizontal direction than in the vertical direction.

In this chapter, four stiffeners are defined over the plate surface. Thus, a total of 12 stiffener shape design variables ($X_s$) are used. In this approach, all the stiffeners are placed independently of each other. The range for the stiffeners’ placement is shown in Table 3.1. Here, $X_{ij}^s$ is the $j^{th}$ control point of the $i^{th}$ stiffener. Both the first and the second control points for any of the stiffeners remain on the boundary of the surface, while the third control point defines the curvature.
Table 3.1: Range of the design variables for placement of four stiffener (See Fig. 3.5)

<table>
<thead>
<tr>
<th>Design Variable</th>
<th>Lower Range</th>
<th>Upper Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stiffener 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{1i}^s$ (A)</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>$X_{12}^s$ (B)</td>
<td>0.25</td>
<td>1.0</td>
</tr>
<tr>
<td>$X_{i3}^s$ ($\alpha_1$)</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Stiffener 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{21}^s$ (C)</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>$X_{22}^s$ (D)</td>
<td>0.5</td>
<td>1.25</td>
</tr>
<tr>
<td>$X_{i3}^s$ ($\alpha_2$)</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Stiffener 3</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{31}^s$ (E)</td>
<td>0.5</td>
<td>0.75</td>
</tr>
<tr>
<td>$X_{32}^s$ (F)</td>
<td>0.75</td>
<td>1.5</td>
</tr>
<tr>
<td>$X_{i3}^s$ ($\alpha_3$)</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
<tr>
<td><strong>Stiffener 4</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_{41}^s$ (G)</td>
<td>0.75</td>
<td>1.0</td>
</tr>
<tr>
<td>$X_{42}^s$ (H)</td>
<td>0</td>
<td>0.75</td>
</tr>
<tr>
<td>$X_{i3}^s$ ($\alpha_4$)</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Size Design Variables**

A rectangular cross-section is considered for the stiffeners. Three size design variables ($X_{size}$) are defined to optimize stiffener cross-section and composite laminate layer thickness:

a. $h_s$: Height of the stiffener

b. $t_s$: Thickness or width of the stiffener cross-section

c. $t_l$: Composite laminate layer thickness

In the current work, a three-layered composite laminate is designed with all the layers having the same thickness. Each layer is a ply in this composite. The thickness of each layer/ply is defined using a single design variable. Also, the same cross-section is assumed for all the stiffeners. The range of the size design variables is shown in Table 3.2.
3.5 Application and Results

In this section, the methodology explained in Section 3.4, is applied to study different stiffener and fiber paths configurations on a stiffened composite laminated plate.

3.5.1 Problem Description

A three-layered composite laminate is considered as shown in Fig. 3.6. All the layers are considered to be having the same thickness, but with independent fiber orientations. Bi-axial compressive loads along with shear loads are considered as shown in Fig. 3.6 under simply-supported boundary conditions. The motivation of the optimization is to evaluate the combination of curvilinear fibers and curvilinear stiffener against the configuration of straight stiffener and straight fibers. The current case studies are conducted using three-layered laminate. But, the presented framework is applicable for composites with any number of laminates. Three different load cases are considered as shown in Table 3.3. These include shear dominated, compression dominated and pure shear loads. Initially, PSO based optimization is conducted with a single objective to maximize buckling load while having a mass constraint of 25 kg under these load cases without any stress constraint. The numerical results from 3 separate optimization attempts for each of the load cases and configurations are reported to show repeatability of the resulting designs. Moreover, the evolutionary based optimization algorithms, like Particle Swarm Optimization (PSO) or Genetic Algorithm (GA), sometimes provide local optimum solution. Therefore, multiple runs must be conducted to

<table>
<thead>
<tr>
<th>$h_s$ (m)</th>
<th>$t_s$ (m)</th>
<th>$t_l$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower bound</td>
<td>0.01</td>
<td>0.001</td>
</tr>
<tr>
<td>Upper bound</td>
<td>0.08</td>
<td>0.01</td>
</tr>
</tbody>
</table>
ensure a confidence in the results. The motivation of this work is to find out the configurations where buckling load could be maximized under shear dominated and compression dominated load cases. The von Mises stress levels are reported on the optimal designs. These stresses are based on the unit load cases as mentioned in Table 3.3. The actual stress for these panels would depend on actual applied loading and can be scaled accordingly. The current study can help the users to design the stiffeners configurations based on their applied loading to be shear or compression dominated. Later, the size design variables (height and thicknesses of stiffener and plate) could be optimized either by manual selection or gradient based optimization. Singh, Zhao and Kapania [1] studied that for designing stiffened panels, buckling load is one of the constraints. In the first step for designing stiffened panels, a configuration could be optimized for maximizing the buckling load. Later, in the second step, the weight of this optimized configuration could be optimized by optimizing the size design variables. The scope of current study was to focus only on maximizing the buckling load on simplified loading to give a reader an insight about the selections of the stiffeners and curvilinear fibers based on certain loading conditions.

Later, a NSGA-II based multi-objective optimization results are presented for maximizing the buckling load while minimizing the structural mass under pure shear load case. As NSGA-II requires a large number of function evaluation for generating a smooth Pareto front, the authors selected only one load case of interest out of many load cases. During the optimization, the global edge length for meshing is set as 0.01 m in MSC.PATRAN. As shown in the Fig. 3.6, the length of the square panel is 1m. The meshing size was selected after conducting a convergence study. MSC.NASTRAN 4-noded “CQUAD4” quadrilateral shell elements are used for modeling both the skin and the stiffeners. In the current work, a composite laminate of material T300/N5208, as shown in Table 3.4, is used for modeling the skin. The stiffeners are assumed to be metallic. An isotropic material, aluminum alloy
3.5. Application and Results

Al 2139, is considered for modeling stiffeners.

<table>
<thead>
<tr>
<th>Load Case</th>
<th>( F_x ) (kN/m)</th>
<th>( F_y ) (kN/m)</th>
<th>( S_f ) (kN/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shear Dominated</td>
<td>0.5</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>Compression Dominated</td>
<td>1</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>Pure Shear</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

3.5.2 PSO Based Optimization Results

In this section, three different stiffener and fiber configurations are studied for three different load cases. The number of particles in a swarm are set to be 25 times the design variables, selected based on the authors’ previous experience with the PSO. This ensures that final results are fairly representative of the global optima.

a) Equidistant Straight Stiffeners with Straight Fiber Paths: Initially, for comparison, equidistant straight stiffeners with straight fibers are considered. In this case, the stiffener shape design variables are fixed. The PSO is used to optimize 6 design variables (3 size design variables, \( X_{\text{size}} \), and 3 straight fiber path shape design variables, \( X_{\text{shape}}^{s} \)). The objective function, shown in Eq. 3.2, is minimized. A set of 150 particles are used while optimizing using PSO.

b) Straight Stiffeners with Straight Fiber Paths: In this case, straight stiffeners with straight fibers are considered. The straight stiffeners are defined using the parameterization of curvilinear stiffeners, but with no curvature (\( \alpha = 0 \)). The PSO optimizes a total of 14 design variables (11 shape design variables, \( X_{\text{shape}} \), and 3 size design variables, \( X_{\text{size}} \)). A set of 350 particles are used while optimizing using PSO.

c) Curvilinear Stiffeners with Curved Fiber Path: For this configuration, both the
Table 3.4: Material properties used for modeling composite laminate skin

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic modulus $E_{11}$ (GPa)</td>
<td>181</td>
</tr>
<tr>
<td>Elastic modulus $E_{22}$ (GPa)</td>
<td>10.3</td>
</tr>
<tr>
<td>Shear modulus $G_{12}$ (GPa)</td>
<td>7.17</td>
</tr>
<tr>
<td>Major Poisson’s ratio $\nu_{12}$</td>
<td>0.28</td>
</tr>
<tr>
<td>Ultimate tensile strength in 1-dir (MPa)</td>
<td>1500</td>
</tr>
<tr>
<td>Ultimate comp strength in 1-dir (MPa)</td>
<td>1500</td>
</tr>
<tr>
<td>Ultimate tensile strength in 2-dir (MPa)</td>
<td>40</td>
</tr>
<tr>
<td>Ultimate comp strength in 2-dir (MPa)</td>
<td>146</td>
</tr>
<tr>
<td>Ultimate in-plane shear strength (MPa)</td>
<td>68</td>
</tr>
</tbody>
</table>

curvilinear stiffeners with curved fiber paths are designed. A total of 21 design variables (18 shape design variables, $X_{\text{shape}}$, and 3 size design variables, $X_{\text{size}}$) are considered. In this model, 525 particles are used while conducting the optimization to minimize the objective function in Eq. 3.2.

Case Study 1

In this section, results from Case Study 1 are presented. In this study, the shear dominated load case, as shown in Table 3.3 and Fig. 3.6, under simply supported boundary conditions is considered. Figure 3.7 shows the results of composite laminate with four equidistant straight stiffeners and straight fibers with 3 separate optimization attempts. The optimal buckling factors are 5195.1, 5420.2 and 5288.4 for runs 1, 2 and 3, respectively. The optimal design, with the highest buckling factor, has almost the same fiber orientation for Layer 1 and 3. The maximum von Mises stress for the optimal design is 0.289 MPa and 0.190 MPa in the laminates and stiffeners respectively. Later, the design is optimized using straight stiffeners and straight fibers in the laminate as shown in Fig. 3.8. During this optimization, the parameterization of curvilinear stiffener is obtained by setting the curvature of the stiffener to
The optimal buckling factors are 5837.1, 5795.6 and 5355.7. The run 3 optimization converged in a local minima and has lower buckling factor than other optimization attempts. The maximum von Mises stress for the optimal design, with the highest buckling factor, is 0.391 MPa and 0.491 MPa in the laminates and stiffeners respectively. The use of straight stiffener over equidistant straight stiffener improved the buckling load by 7.69% while using straight fibers. It has been found that Layer 1 and 3 again have similar fiber paths. Also, the optimal stiffener configuration shows that all the stiffener orient towards one direction along the diagonal of the panel.

The authors noticed during optimization that the material has higher elastic modulus in one direction while being weak in the perpendicular direction. Therefore, to compensate, the optimizer tries to define the fiber angle of one of the layers in the direction other than the
fiber path direction of other two layers. Moreover, the optimal fiber path also depends on the loading condition. If there were single loading in $x$-direction, $F_x$, while other loading being zero, the optimizer would have tried to design all the fiber path direction in that direction. However, in our case, the loading is a combination of shear and compression load, and thus, the optimizer may not put all the fiber path in one direction.

Finally, Fig. 3.9 shows the optimal design using curvilinear stiffeners and curvilinear fibers. The optimal buckling factors are 6496.3, 5970.4, 6354.7 for different optimization attempts. The maximum von Mises stress for the optimal design, with the highest buckling factor, is 0.404 MPa and 0.408 MPa in the laminates and stiffeners respectively. In this design also, the fibers in both Layers 1 and 3 are almost straight and very similar. In comparison to the previous configuration, the change in the middle layer is not as dramatic. The value of the
constraint $g_1(x)$ is almost zero for the optimized models. Table 3.5 shows the comparison of the buckling factor for all the optimized configurations of this case study. It is seen that the configuration with curved stiffeners and curved fibers has 11.29% higher buckling load than the configuration with straight stiffeners and straight fibers, and 19.85% higher buckling load than the configuration with equidistant straight stiffeners and straight fibers. It has been seen that the optimal configuration of curvilinear stiffeners has almost no curvature and is close to the optimal configuration of straight stiffeners. This shows that the placement of the stiffener is more important than the curvature of the curvilinear stiffener for enhancing the buckling capacity of the panel. The use of curvilinear fibers helped improving the buckling load of the model. Thus, it shows that the improvement in the buckling load is higher by using the curvilinear fibers over straight fibers in comparison to the use of curvilinear stiffeners over straight stiffeners. However, it is also seen that the von Mises stress increased when stiffener configuration changed from equidistant straight stiffeners to curvilinear stiffener.

Table 3.5: Comparison of the optimized results under shear dominated load case (Case Study 1)

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Buckling Factor</th>
<th>% Improvement w.r.t (a)</th>
<th>% Improvement w.r.t (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Equidistant Straight Stiffeners and Straight Fibers</td>
<td>5420.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Straight Stiffeners and Straight Fibers</td>
<td>5837.1</td>
<td>7.69</td>
<td></td>
</tr>
<tr>
<td>(c) Curved Stiffeners and Curved Fibers</td>
<td>6496.3</td>
<td>19.85</td>
<td>11.29</td>
</tr>
</tbody>
</table>
Figure 3.8: Composite laminate using straight stiffeners with straight fibers (Case Study 1)

Figure 3.9: Composite laminate using curved stiffeners with curved fibers (Case Study 1)
3.5. Application and Results

**Case Study 2**

In Case 2, a compression dominated load case is considered as shown in Table 3.3. The load case consists of both compressive and shear loads. However, the compression loads are greater in magnitude than the shear loads. Figure 3.10 shows the numerical results for composite laminate with four equidistant stiffeners and optimal straight fibers with different optimization attempts. The optimal buckling factors are 2572.1, 2520.4 and 2627.4. The maximum von Mises stress for the optimal design, with the highest buckling factor, is 0.133 MPa and 0.557 MPa in the laminates and stiffeners respectively. The fiber angles of layers 1 and 3 are almost the same. Figure 3.11 shows the optimal design using straight stiffeners and straight fibers. The optimal buckling factors are 2570.6, 2513.2 and 2459.0. It is seen that in the middle layer follows the direction of orientation of the stiffeners and the top and the bottom layers follows the opposite. For example, in run 1, the stiffeners are oriented towards the left and, in run 2, they are oriented towards the right. Their orientation is followed by the middle layer while the top and bottom layers orient in its opposite direction. The maximum von Mises stress for the optimal design, with the highest buckling factor, is 0.264 MPa and 0.153 MPa in the laminates and stiffeners respectively. In this case study, there is no significant improvement seen with the use of straight stiffeners over the equidistant straight stiffener while keeping the fibers to be straight.

Finally, the configuration with curvilinear stiffeners and curvilinear fibers is shown in Fig. 3.12. The optimal buckling factors are 2550.7, 2672.6 and 2477.1. The maximum von Mises stress for the optimal design, with the highest buckling factor, is 0.285 MPa and 0.256 MPa in the laminates and stiffeners respectively. It is found that the optimized design has optimal stiffener shapes similar to the equidistant straight stiffeners. In this case study, there is no significant improvement seen with the use of either curved fibers or curved stiffeners. All the optimal designs have very similar buckling factors. As expected, equidistant stiffeners
perform better under compression load cases, thus, in current case studies, the configuration of curvilinear fibers and curvilinear stiffeners could not improve the buckling load. Table 3.6 shows the comparison of the buckling factors for all the optimized configurations in the compression-dominated load case. This case study showed that in certain load cases and under some boundary conditions, curved stiffeners and curved fibers may not improve the buckling load.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Buckling Factor</th>
<th>% Improvement w.r.t (a)</th>
<th>% Improvement w.r.t (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Equidistant Straight Stiffeners and Straight Fibers</td>
<td>2627.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Straight Stiffeners and Straight Fibers</td>
<td>2570.6</td>
<td>-2.16</td>
<td></td>
</tr>
<tr>
<td>(c) Curved Stiffeners and Curved Fibers</td>
<td>2672.6</td>
<td>1.71</td>
<td>3.96</td>
</tr>
</tbody>
</table>

Table 3.6: Comparison of the optimized results under compression dominated load case (Case Study 2)
3.5. Application and Results

Figure 3.10: Composite laminate using equidistant straight stiffeners with straight fibers (Case Study 2)

Figure 3.11: Composite laminate using straight stiffeners with straight fibers (Case Study 2)
Case Study 3

In this case study, the pure shear load case is considered as shown in Table 3.3. Figure 3.13 shows the optimized composite laminate with four equidistant straight stiffeners and straight fibers with different optimization attempts. The optimal buckling factors are 6882.8, 6836.8 and 6885.4. The maximum von Mises stress for the optimal design, with the highest buckling factor, is 0.242 MPa and 0.872 MPa in the laminates and stiffeners respectively. Once again, it is seen that the fiber angles of layers 1 and 3 are almost the same and are nearly perpendicular to the middle layer of the laminate. Figure 3.14 shows the optimal design using straight stiffeners and straight fibers. The optimal buckling factors are 11114.4, 12106.4 and 11442.6. The maximum von Mises stress for the optimal design, with the highest buckling factor, is 0.300 MPa and 0.271 MPa in the laminates and stiffeners respectively. This design has a
75.82% higher buckling load than the case with equidistant straight stiffeners and straight fibers. This shows that in this case by changing the stiffener placement and orientation, the buckling load can be changed significantly.

![Composite laminate using equidistant straight stiffeners with straight fibers (Case Study 3)](image)

Lastly, the configuration with curvilinear stiffeners and curvilinear fibers is shown in Fig. 3.15. The optimal buckling factors are 12333.5, 11256.2 and 12361.7 for different optimization attempts. The maximum von Mises stress for the optimal design, with the highest buckling factor, is 0.513 MPa and 0.186 MPa in the laminates and stiffeners respectively. This design also showed a significant increase in the buckling load with respect to equidistant straight stiffener and straight fiber configuration. The value of the constraint $g_1(x)$ is again almost zero in the optimized models. The optimal designs show that if stiffener are oriented towards the diagonal of the panel, then they would be able to better support the panel and would
increase its buckling capacity. This result is similar to what Singh et al. [34] found out with shear load case for isotropic panel with curvilinear stiffener. Table 3.7 shows the comparison of the buckling factor for all the optimized configurations of this case study. It is seen that the configuration with curved stiffeners and curved fibers has a 79.53% higher buckling load than the configuration with equidistant straight stiffeners and straight fibers, and a 2.1% higher buckling load than the configuration with straight stiffeners and straight fibers. The von Mises stress increased in the laminates, while decreased in the stiffeners, by changing the stiffener configuration from equidistant straight stiffeners to curvilinear stiffeners as former do not perform well in shear loads. Again, it has been seen that the stiffener placement is more important than the curvature of the stiffener. This shows that the use of optimally placed stiffeners could be more beneficial over equidistant straight stiffeners than the use of
3.5. Application and Results

curved fibers over straight fibers for certain loading conditions.

In the case studies presented in this section, three different stiffener configurations are included as: 1) equidistant straight stiffeners with straight fibers, 2) straight stiffeners with straight fibers, and 3) curved stiffeners with curved fibers. It is seen that the curvature of the stiffeners is not important because the optimal designs of the panels with the curved stiffeners have negligible curvature or it can be said that the curved stiffeners became almost straight in the optimal designs. Therefore, the case study with curved stiffeners and straight stiffeners is not included in the manuscript.

Figure 3.15: Composite laminate using curved stiffeners with curved fibers (Case Study 3)
Table 3.7: Comparison of the optimized results under pure shear load case (Case Study 3)

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Buckling Factor</th>
<th>% Improvement w.r.t (a)</th>
<th>% Improvement w.r.t (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Equidistant Straight Stiffeners and Straight Fibers</td>
<td>6885.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Straight Stiffeners and Straight Fibers</td>
<td>12106.4</td>
<td>75.82</td>
<td></td>
</tr>
<tr>
<td>(c) Curved Stiffeners and Curved Fibers</td>
<td>12361.7</td>
<td>79.53</td>
<td>2.10</td>
</tr>
</tbody>
</table>

3.5.3 NSGA-II Based Optimization Results

In this section, results of NSGA-II based optimization are presented. The benefit of using curved stiffeners and curved fibers in pure shear load case over the use of equidistant straight stiffeners and straight fibers motivated the authors to evaluate its performance using a multi-objective algorithm. As it is seen that there is no significant difference between the optimal buckling factors with the configurations of curvilinear stiffener or straight stiffener for pure shear load case, therefore, only the configuration of curvilinear stiffeners and curvilinear fiber is compared with equidistant straight stiffener and straight fibers in this section. An open-source Python Library, Distributed Evolutionary Algorithm in Python (DEAP), is used for conducting the NSGA-II optimization. The library was first verified on known examples before conducting optimization studies presented in this chapter. The objective of the optimization is to maximize the buckling load while minimizing the structural mass of the composite panel. In this study, two configurations: 1) curved stiffener and curved fibers, and 2) equidistant straight stiffener and straight fiber are evaluated. During the optimization, the number of particles are set to be 20 times the design variables. This was selected based on authors’ previous experience. The optimization was ran for 150 generations with the motivation to generate a smooth Pareto-Front. Figure 3.16 shows the Pareto-Fronts for different configurations of stiffener and fibers in the composite laminate. Figure 3.17 shows
the comparison of Pareto optimal set of different configurations. The figure also shows the variation of possible improvement in the buckling load, by using curved stiffener and curved fiber configurations, with respect to the configuration of equidistant straight stiffeners and straight fibers. It can be seen from the results that the improvement in the buckling load is dependent upon the user specified structural mass constraint. With lower structural mass constraint, higher improvement is possible in the buckling load. As there is an upper bound on the design variables of height and thickness of the stiffener and thickness of plate layers, the optimizer starts to have less flexibility for increasing the buckling load. However, with lower weight constraints, there is more flexibility for the optimizer to design the stiffeners and hence a higher percentage of increase in the buckling load is seen. For example, as long as the stiffener height and thicknesses are below the upper bound limit, the optimizer tries to increase the height of the stiffeners for increasing the buckling load capacity of the structure. However, this starts becoming difficult once the design variables of the stiffeners become close to its upper bound. Furthermore, the Figs. 3.16 and 3.17 also show and support the conclusion from the PSO-based optimization study carried out in Section 3.5.2, that the use of curvilinear stiffeners could be beneficial over equidistant straight stiffeners.

Figure 3.16: Pareto-Fronts of NSGA-II Optimization. (Left) Curved Stiffener + Curved Fibers. (Right) Equidistant Straight Stiffeners + straight Fibers
Figure 3.17: (Left) Pareto-fronts of NSGA-II optimization under pure shear load case. (Right) Improvement in buckling load over use of equidistant straight stiffeners + straight fibers

3.6 Conclusions

In this chapter, a framework for designing a tow-steered composite laminate with curvilinear isotropic stiffeners using parallel processing is presented. The motivation of the current research is to evaluate the advantages of using the combination of the curvilinear stiffeners along with curvilinear fibers in a composite laminate using computational resources. An integrated optimization framework utilizing the versatile capabilities of Rhinoceros 3D, MSC.PATRAN, MSC.NASTRAN and the scripting-based language Python has been developed. A three-layered composite laminate stiffened by four curvilinear stiffeners is subjected to different load cases and the results for the buckling load factors are presented. It has been seen that the combination of curvilinear stiffeners and curvilinear fibers can be beneficial for increasing the buckling load in shear dominated load cases. However, it is entirely possible that under certain load cases and boundary conditions, there may not be any significant benefit to using curved stiffeners and curved fibers. There are many load cases and boundary conditions beyond the scope of the presented research for optimizing a stiffened panel.
3.6. Conclusions

The authors conducted the current research with the motivation to optimize the panel under certain selected load cases and one boundary condition to evaluate the benefit of using the combination of curvilinear stiffeners and curvilinear fibers. Firstly, the PSO based optimization with a single objective is presented. Double-level parallel processing was developed to save significant computer run-time. For different load cases, different stiffener and fiber configurations are studied. It is seen that curvilinear stiffeners along with the use of optimal curvilinear fiber paths in composite laminate skin can lead to a more than a 75% increase in the buckling load as compared to the use of equidistant straight stiffeners and straight fibers under certain loading conditions. Finally, a multi-objective NSGA-II based optimization study is presented for evaluating the performance of different configurations under a pure shear load. It showed that curved stiffener and fibers have great potential for improving the buckling load, but the benefit depends on the user defined structural mass constraint. It is also seen that the placement of the stiffeners is more important in comparison to its curvature under studied load cases.
Chapter 4

An Optimization Framework For Curvilinearly Stiffened Composite Pressure Vessels And Pipes

4.1 Abstract

With improvement in innovative manufacturing technologies, it is now possible to fabricate any complex shaped structural design for practical applications. This innovative manufacturing technology allows for the fabrication of curvilinearly stiffened pressure vessels and pipes. Compared to straight stiffeners, curvilinear stiffeners have been shown to have better structural performance and weight savings under certain loading conditions. In this chapter, an optimization framework for optimal structural design for curvilinearly stiffened composite pressure vessels and pipes is presented. Non-Uniform Rational B-Spline (NURBS) curves are utilized to define curvilinear stiffeners over the surface of the pipe. An integrated tool using Python, NURBS-based Rhinoceros 3D, MSC.PATRAN and MSC.NASTRAN is implemented for performing topology optimization of curvilinearly stiffened cylindrical shells. Rhinoceros 3D is used for creating the geometry, which later can be exported to MSC.PATRAN for finite element model generation. Finally, MSC.NASTRAN is used to perform structural analysis. A Bi-Level Programming (BLP) optimization technique, consisting of Particle Swarm
Optimization (PSO) and Gradient Based Optimization (GBO), is used for finding the optimized locations of stiffeners, optimal geometric dimensions for stiffener cross-sections and the optimal layer thickness for the composite skin. Optimization studies show that stiffener placement influences the buckling mode of the structure. Furthermore, the structural weight can be decreased by optimizing the stiffener’s cross-section and skin thickness. In this chapter, a cylindrical pipe stiffened by orthogonal and curvilinear stiffeners under torsional and bending load cases is studied. It is seen that curvilinear stiffeners can lead to a potential 10.8% weight saving in the structure as compared to the case of using straight stiffeners.

4.2 Introduction

Pressure vessels and pipes are integral parts of many industries like oil refineries, chemical and nuclear industries etc. The important design loads for designing these pressure vessels and pipes are internal pressure, external pressure (vacuum inside) and boundary loads such as bending moment, axial compression and torsion. Design temperature is also an important variable for pressure vessels and piping [40]. Modern developments in manufacturing technologies have made it possible to manufacture high complexity shapes and designs. High performance computing and commercial software such as MSC.NASTRAN, MSC.PATRAN have the capability to both analyze and optimize a large class of such designs. Arbitrary-shaped geometric modeling of the these designs has become possible using commercially available geometric modeling software, such as Rhinoceros 3D, using Non-Uniform Rational B-Splines (NURBS). Advancements in manufacturing, designing and analysis capabilities have made it possible to design curvilinear stiffened plates and shells for practical applications.

In past decades, extensive research was conducted on the design and analysis of pressure
vessels and pipes. Murray et al. [41] experimentally investigated the bending and axial strain capacity of metallic pipelines under combined axial compression, bending and internal pressure. Their experimental tests involved pressurizing a section of a pipe and then putting a compressive load on it, until it failed. The load is applied either along the pipe axis or eccentrically in order to apply additional bending. Limam et al. [42] investigated the problem of inelastic bending and the collapse of elasto-plastic tubes in the presence of internal pressure using experiments and computational analysis. They presented results where small wrinkles appeared on the compressed side of the tube whose amplitude grew stably as bending progressed, eventually causing failure. Parnas and Katırcı [40] developed an analytical procedure to design and predict the behavior of fiber-reinforced composite pressure vessels under various loading conditions. They considered an internal pressure, axial force and body force due to rotation in addition to temperature and moisture variation throughout the body. Paquette and Kyriakides [43] studied plastic buckling and collapse of long metallic cylinders under combined internal pressure and axial compression through experiments and analyses. They have shown that under continued compression, the wrinkles grew stably on the surface, eventually leading to limit load instability.

The conventional method to use stiffeners for pressure vessels is to use straight stiffeners in the longitudinal direction and rings in the circumferential direction. In order to enhance the design space, one might consider curved stiffening members. During the last decade, significant research has been conducted on the use of curvilinear stiffeners for designing stiffened plates and shells. Kapania et al. [44] have shown that curvilinear stiffened panels lead to lighter weight designs than panels with straight stiffeners under certain design loads. Zhao and Kapania [45] studied the buckling response of curvilinearly stiffened composite plates subjected to various in-plane loads. Their research shows that the buckling response can be improved using curvilinear stiffeners in addition to tailoring the laminate configurations.
Kidane et al. [46] conducted buckling load analysis by developing an analytical model for the determination of equivalent stiffness parameters of a grid-stiffened composite cylindrical shell.

In this chapter, a bilevel programming (BLP) based optimization technique utilizing the scripting based language Python, NURBS-based Rhinoceros 3D [16], MSC.PATRAN and MSC.NASTRAN has been developed in the form of a computational design environment to design stiffened pressure vessels and pipes. The surface of the pipe is modeled with composite laminate material. The optimization technique involves a combination of Particle Swarm Optimization (PSO) and Gradient Based Optimization (GBO). This optimization has been implemented for finding the optimal design of a stiffened pipe using orthogonally-placed and curvilinearly-placed stiffeners. Linear static and buckling analysis have been conducted for satisfying composite skin failure, buckling and stress constraints, with the motivation of minimizing the weight of the structure. Rhinoceros 3D is used for modeling the geometry of the structure including the surface of the pipe and stiffener placement. MSC.PATRAN is used to generate the mesh and to define the loading and boundary conditions. MSC.NASTRAN is used for conducting finite element analysis including buckling and stress analysis. Considering the case where there is repeated mesh required for topology optimization of stiffened shell whenever there is change in stiffener placement, a built-in capability of MSC.NASTRAN known as Glue Contact has been used to define contact between the surface of the skin and the stiffeners to satisfy the equilibrium and the compatibility conditions at the stiffener-shell interfaces. The capability has an advantage of placing stiffeners arbitrarily on the skin of the pipe, without the requirement to place the nodes at the stiffener-shell interfaces. In Section 4.3, the methodology for performing the optimization is presented. In Section 4.4, the method has been applied to two stiffener configurations (orthogonally and curvilinearly-placed) subject to a bending and torsional load cases.
4.3 Methodology

The current work is focused on utilizing and evaluating the different capabilities of various commercially available software to solve the problem at hand. This enables the analysis of curvilinearly stiffened pipes with the objective of minimizing the structural weight by optimizing stiffener placement and their cross-section.

![Flowchart of global optimization](image)

**Figure 4.1: Flowchart of global optimization**

4.3.1 Procedure

In this chapter, the geometry has been represented by NURBS (Non-Uniform Rational B-Splines). A commercially available geometric modeling software, Rhinoceros 3D, is used to generate a NURBS-based CAD model for a stiffened pipe. The geometry is then exported in IGES format to MSC.PATRAN. MSC.PATRAN’s built-in native language, Pa-
tran Command Language (PCL), is used to generate session files. The session file can be used to perform finite element analysis multiple times for geometries with different dimensional parameters. Finally, the obtained mesh along with the applied loads are passed onto MSC.NASTRAN for linear static and buckling analysis. A Python script is written to control the complete optimization process. The flowchart of the global optimization is shown in Fig. 4.1. The optimization is conducted based on BLP optimization technique. At upper level, PSO is used to minimize the weight of the structure without any constraints. At lower level, GBO is used to minimize the weight under buckling, stiffener’s stress and composite skin failure index constraints. The optimization starts with an initial set of values, \( X_{\text{size}} \), for the cross-section of the stiffeners and the pipe’s surface thickness. The stiffener placement shape design variables are denoted by \( X_{\text{shape}} \) in this chapter. The range of \( X_{\text{shape}} \) along with the initial user-defined values of \( X_{\text{size}} \) are provided to the PSO. The details of the PSO are mentioned in the Section 4.3.2. For each particle of the PSO, size optimization using GBO has been implemented to optimize the stiffener thickness and shell thickness \( (X_{\text{size}}) \) for a given fixed stiffener shape design \( (X_{\text{shape}}) \). Therefore, for each particle of the PSO, an optimal size design is obtained. Zhao et al. [47] utilized a similar technique to optimize the internal structural layout of a composite aircraft wing. Compared to a two-step optimization approach [1], where PSO was used for shape optimization and then GBO was used separately for size optimization, the presented BLP based single step optimization approach converges faster [47].

### 4.3.2 Global Optimization

The global optimization framework is shown in Fig 4.1. In PSO, the objective of the optimization is to minimize the weight, \( f(x) \), without any constraints as shown in Eq. 4.1. The \( a \) and \( b \) denote bounds on \( x \) and \( x_h \) is a \( h^{th} \) design variable.
**Objective function**: Minimize \( f(x) \),

\[
\begin{align*}
a_h & \leq x_h \leq b_h, & h = 1, 2, \ldots, n \\
\end{align*}
\] (4.1)

In this optimization, a set of random particles uniformly distributed, known as designs in the present problem, are defined over the full domain of design variables. The design variables are shape design variables, \( X_{shape} \), in our case. The objective function is evaluated for every particle. The required number of particles, \( N \), and the maximum possible generations, \( m \), can be set by the user according to the number of design variables. The design variables of the particle are updated based on the best particle in the swarm with the best fitness function. After each generation, updated values of the design variables are evaluated for finding the required value of the objective function.

Fig. 4.1 shows a complete PSO iteration. For the first iteration, PSO generates its own \( N \) particles depending upon the user input values. For each particle, Rhinoceros 3D is called to generate the required CAD model through Rhino:Python, which is then exported to MSC.PATRAN for generating the mesh along with the boundary conditions and applied loads. MSC.NASTRAN then conducts the structural analysis and size optimization. The objective of the size optimization is to minimize the weight with user-defined constraints, such as buckling, stiffener’s stress and composite skin failure index. The GBO (MSC.NASTRAN SOL 200) would optimize composite skin laminate thickness and stiffener cross-section while satisfying the constraints. There can be some cases where the GBO could not converge to an optimal feasible design or if MSC.PATRAN could not successfully generate a mesh for a model. In these specific cases, a penalty is given to the objective function in PSO by providing a very large value to that particle in the swarm. Otherwise, the optimized weight is recorded for that particular particle.
4.3. Methodology

4.3.3 Verification Study for *Glue Contact* Capability

In the present work, a built-in capability of MSC.NASTRAN known as *Glue Contact* has been used to define contact between the surface of the skin and the stiffeners to satisfy the equilibrium and compatibility conditions. This capability has the advantage of placing stiffeners arbitrarily on the skin of the pipe. This eliminates the need for coincident nodes at stiffener-shell interfaces. The technology can be applied as linear contact where two bodies remain in contact in any condition. It is also possible to provide a very large value (1e20) of frictional shear stress, for the stiffener-shell interfaces when using *Glue Contact*, in MSC.NASTRAN in order to avoid any failure of contact between the bodies. Initially, in order to verify this capability, a simple verification study was conducted with an arbitrary model. A plate with curvilinear stiffeners on its surface has been considered. Fig. 4.2 shows the results of the model using *Glue Contact*, where the stiffeners and skin have dissimilar...
meshes, meaning the nodes of stiffeners and skin do not coincide at the interface. The results are compared to an equivalence model, where the nodes of the stiffener and shell elements coincide at the shell-stiffener interfaces. It can be seen that results between the two methods vary very slightly, just more than 1%.

4.4 Parameterization Of Stiffener Placement

4.4.1 Parameterization Process

In this chapter, a stiffener’s initial placement is defined in a 2D base reference plane, with corners A, B, C and D, as shown in Fig. 4.3. To parameterize a grid of curvilinear stiffeners, four parameters have been defined; \( n_j, n_k, \alpha_j \) and \( \alpha_k \). The parameters, \( n_j, n_k \) represent the number of curves or stiffeners and \( \alpha_j, \alpha_k \) represent the angle of these curves respectively. In
4.4. Parameterization Of Stiffener Placement

Step 1: a curve is defined starting from D at an angle $\alpha_j$ as specified by the user. Later, the $n_j - 1$ remaining curves are defined by offsetting the first initially defined curve by a distance, $(DC/n_j)$, in the positive x-direction along the edge DC. The parameter $\alpha_j$ is only allowed to vary from $0^\circ$ to $90^\circ$. In Step 2, all curves which are going out of the reference surface are identified and trimmed at the intersection with the edge BC. For example, in Fig. 4.3, the
curve GH is intersecting BC at J. In Step 3, all the trimmed curves outside the reference surface are identified and translated in the negative x-direction with a distance equal to DC. After translation of the trimmed curves, shown in green, a new point $J'$ is formed. If any of the new curves still intersect BC, the same procedure (Steps 1-3) is repeated until there are no curves outside of the reference surface. In Step 4, a complete grid is defined by repeating Steps 1-3, but in the opposite direction. A curve is defined starting from C at an angle $\alpha_k$, as specified by the user. Later, the $n_k - 1$ curves are defined by offsetting the first curve with the distance, $DC/n_k$, in the negative x-direction. The parameter, $\alpha_k$ is only allowed to vary from $0^\circ$ to $90^\circ$. All curves that are going out of the reference surface are identified and trimmed at the intersection with the edge AD. Later, all the trimmed curves (shown in white color in Fig. 4.3) are translated towards positive x-direction with the distance equal to edge DC. The parameterization is implemented by writing a Python script that is run in Rhinoceros 3D using its feature of *Rhino.Python*.

4.4.2 Transformation of Stiffeners from Reference Surface to Physical Surface

After the definition of the stiffeners in 2D base reference surface, the stiffeners are transformed onto a physical surface as shown in Fig. 4.4. The 2D base rectangular surface is folded to make a cylindrical surface. In the figure, the rectangular surface ABCD has been folded and the edges AB and CD have been joined. With the presented parameterization, the stiffeners automatically match at the joining edge. It can be seen that the edge AB became the circumference of the cylindrical surface and the edge BC became the length of the cylinder. The transformation has the advantage of transforming stiffeners to any complexly-shaped surface. The transformation has been implemented in Rhinoceros 3D using the command *FlowAlongSrf*. This command can map any curve from a source surface to a target surface.
4.5 Problem Description

In this chapter, a stiffened cylindrical pipe or pressure vessel is studied. Fig. 4.5 shows the boundary applied loading under simply supported boundary conditions. A cylindrical coordinate system is used for defining the geometry. Two load cases have been considered as 1) an in-plane bending moment \( (M) \) along with an internal pressure \( (P) \) 2) a torque \( (T) \) along with internal pressure \( (P) \) have been considered as shown in the figure. A three-layered composite laminate has been considered to define the material of the skin. The structure is designed under these two load cases with the objective of minimizing the weight.

4.6 Application and Results

In this section, the methodology, explained in Section 4.3, has been applied to study different stiffener configurations. The optimization is performed using parallel processing [48]. The fact that the function evaluations in PSO are independent allows for the use of parallel computation at each iteration and results in a significantly less computer run-time. The machines used in these studies have 2.2 GHz AMD Opteron Processors, at least 132 GB of RAM, and 48 CPUs.

In first load case, a bending moment \( (M) \) in Fig. 4.5) of 250 kN-m and an internal pressure \( (P) \) of 100 kPa is applied on the structure. In second load case, a torque \( (T) \) in Fig. 4.5) of 250 kN-m and an internal pressure \( (P) \) of 100 kPa is applied on the structure. There could be many load cases beyond the scope of this chapter. The radius and the length of the pipe are 0.5 \( m \) and 3.14 \( m \) respectively. The length of the pipe has been set equal to the circumference for this problem. Two different stiffener configurations have been studied as follows: 1) Orthogonal stiffeners (conventional stiffeners); 2) Curvilinear stiffeners. The
presented framework is designed such that the user could change the type of the applied loads at the ends of the pipe. The current studies demonstrate the use of the presented framework under two load cases. The framework is capable of designing pressure vessel and pipes under internal pressure, axial loads, shear loads and moment loads.

The skin of the cylinder is modeled with the composite laminate with woven fabric. The properties of the composite laminate material are mentioned in Table 4.1. An isotropic material, aluminum alloy Al 2139, has been considered for the stiffeners. The material properties for the isotropic material are shown in Table 4.2. The skin mesh is kept same in all the stiffener configurations studied in this chapter. MSC.NASTRAN “CTRIA3” triangular shell elements are used for modeling both the skin and the stiffeners. During mesh generation of skin of the cylinder in MSC.PATRAN, the average element edge length is set to 0.02 m for a converged result. For stiffeners, three elements through the height of the stiffener are used. The number of elements of different models could vary from 50,000 to 250,000 depending upon the number of stiffeners in the structure. The time taken by the GBO for size optimization of the structure depend upon the mesh size, the CPU configurations and how close the initial guesses for the design variables are from the local optima for any particular model. The time taken by GBO could vary from 10 min to 1.5 hours for optimizing the structure. In the presented studies, optimization wall-clock time varied from 20-30 hours while making use of parallel processing. Thus, this is a computationally expensive problem.

4.6.1 Constraints

The buckling factor, $\lambda$, is defined as the ratio of the buckling load of the structure to the applied load. Due to in-plane bending, there can be wrinkling [42] on the compressed side of the pipe. Also, the pipe could buckle under applied torque. Thus, a buckling constraint
4.6. Application and Results

Table 4.1: Material properties for composite laminate (carbon fiber) used for modeling skin

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic modulus $E_{11}$ (GPa)</td>
<td>70</td>
</tr>
<tr>
<td>Elastic modulus $E_{22}$ (GPa)</td>
<td>70</td>
</tr>
<tr>
<td>Shear modulus $G_{12}$ (GPa)</td>
<td>5</td>
</tr>
<tr>
<td>Major Poisson’s ratio $\nu_{12}$</td>
<td>0.1</td>
</tr>
<tr>
<td>Ultimate tensile strength in 1-dir (MPa)</td>
<td>600</td>
</tr>
<tr>
<td>Ultimate comp strength in 1-dir (MPa)</td>
<td>570</td>
</tr>
<tr>
<td>Ultimate tensile strength in 2-dir (MPa)</td>
<td>600</td>
</tr>
<tr>
<td>Ultimate comp strength in 2-dir (MPa)</td>
<td>570</td>
</tr>
<tr>
<td>Ultimate in-plane shear strength (MPa)</td>
<td>90</td>
</tr>
</tbody>
</table>

Table 4.2: Isotropic material properties used for modeling stiffeners

<table>
<thead>
<tr>
<th>Material (Aluminum Alloy)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Al 2139</td>
<td></td>
</tr>
<tr>
<td>Young’s Modulus (E) (GPa)</td>
<td>73.085</td>
</tr>
<tr>
<td>Poisson ratio</td>
<td>0.33</td>
</tr>
<tr>
<td>Allowable stress (MPa)</td>
<td>427.47</td>
</tr>
</tbody>
</table>

has been considered during optimization. In order to consider the failure of the composite laminate, the Tsai – Wu failure criterion has been used. According to the criterion, if the Failure Index (F.I.) < 1, then the composite laminate is safe. Also, a stress factor, defined as a ratio of the maximum von Mises stress and the yield stress ($\sigma_y$), is found for the stiffeners. Therefore, three constraints have been defined as:

- 1) Buckling Factor ($\lambda$) > 1
- 2) Failure Index (F.I.) < 1
- 3) Stress Factor (Stiffeners) < 1

The failure criterion constraint has been applied to each layer of the composite laminate. All the constraints are defined in MSC.PATRAN and are provided to MSC.NASTRAN for
In this section, the results of application of the methodology using orthogonal stiffeners under torsional and bending load cases are presented.

Design Variables:

Fig. 4.6 shows the different design variables that have been used to define the orthogonal stiffeners.
4.6. Application and Results

- $n_h$: Number of hoop stiffeners

- $n_l$: Number of longitudinal stiffeners

- $t_s$: Thickness of the stiffener

- $h_s$: Height of the stiffener

- $t_p$: Thickness of the each layer of composite laminate

The optimization starts with some user-input values of the size design variables, $X_{size}$, which includes $t_s$ and $t_p$. The shape design variables, $X_{shape}$, include $n_h$, $n_l$ and $h_s$. The PSO is provided with the domain of $X_{shape}$ as shown in the Table 4.3. The GBO is provided with the range of $X_{size}$ design variables as shown in Table 4.4. The optimal configuration is shown in Fig. 4.7 for torsional load case. The buckling factor of optimal configuration is 1.000 and the maximum failure index is 0.913. The stress factor is 0.995. This shows that buckling and stress factor constraints are both active in the optimal design. The mass of the optimal structure is 36.48 kg. The optimal configuration shows that adding longitudinal stiffeners could be more helpful in saving weight as compared to addition of hoop stiffeners.

The optimal configuration for bending load case is shown in Fig. 4.8. In this configuration, only the buckling constraint is active with the buckling factor of 0.999. The maximum failure index and stress factor are 0.107 and 0.813. The optimal mass of the structure in bending load case is 36.10 kg. It can be seen that local buckling occurs in the skin. Also, in comparison to the torsional load case, in bending load case adding both hoop and longitudinal stiffeners could help in reducing the weight. The optimal values of the design variables are tabulated in Table 4.5.
Table 4.3: Range of $X_{shape}$ for orthogonal stiffeners

<table>
<thead>
<tr>
<th></th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_h$</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>$n_l$</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>$h_s$ (m)</td>
<td>0.005</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 4.4: Range of $X_{size}$ design variables

<table>
<thead>
<tr>
<th></th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_s$ (m)</td>
<td>0.001</td>
<td>0.01</td>
</tr>
<tr>
<td>$t_p$ (m)</td>
<td>0.0001</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Figure 4.7: Optimal configuration using orthogonal stiffeners under torsional load case. ($n_h = 1$, $n_l = 12$, buckling factor = 1.000, maximum failure index = 0.913, stress factor (stiffeners) = 0.995, mass = 36.48 kg). The optimal cross-section dimensions are given in Table 4.5.
4.6. Application and Results

Figure 4.8: Optimal configuration using orthogonal stiffeners under bending load case. \( n_h = 19, \ n_l = 12, \) buckling factor = 0.999, maximum failure index = 0.107, stress factor (stiffeners) = 0.813, mass = 36.10 kg. The optimal cross-section dimensions are given in Table 4.5.

Table 4.5: Optimal final values of design variables for orthogonal stiffeners

<table>
<thead>
<tr>
<th></th>
<th>Torsional Load Case</th>
<th>Bending Moment Load Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_h )</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>( n_l )</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>( h_s ) (m)</td>
<td>3.914E-02</td>
<td>1.067E-02</td>
</tr>
<tr>
<td>( t_s ) (m)</td>
<td>1.000E-03</td>
<td>1.000E-03</td>
</tr>
<tr>
<td>( t_p ) (m)</td>
<td>6.788E-04</td>
<td>7.025E-04</td>
</tr>
<tr>
<td>Mass (kg)</td>
<td>36.48</td>
<td>36.10</td>
</tr>
</tbody>
</table>

4.6.3 Curvilinear Stiffeners

In this section, the results from application of the methodology using curvilinear stiffeners are presented.
Design Variables

Fig. 4.3 shows the different design variables that have been used to define the curvilinear stiffeners. In Fig. 4.3, AB=BC=CD=DA. The size design variables in this section are same like that of case of orthogonal stiffeners.

- \( n_j \): Number of stiffeners in the direction from left to right
- \( n_k \): Number of stiffeners in the direction from right to left
- \( \alpha_j \): Orientation angle of \( n_j \) stiffeners
- \( \alpha_k \): Orientation angle of \( n_k \) stiffeners

As stated above, the optimization starts with user-input values of the size design variables, \( X_{size} \). \( X_{size} \) includes \( h_s \) and \( t_p \). The shape design variables, \( X_{shape} \), includes \( n_j, n_k, \alpha_j, \alpha_k \) and \( h_s \). The PSO is provided with the domain of \( X_{shape} \) as shown in the Table 4.6. The GBO is provided with the range of \( X_{size} \) design variables as shown in Table 4.4. The parameters \( \alpha_j \) and \( \alpha_k \), shown in Fig. 4.3, can vary from 22.5\(^\circ\) to 90\(^\circ\). Fig. 4.9 shows the optimal configuration using curvilinear stiffeners under torsional load case. The buckling constraint is active with the buckling factor of 1.004. The maximum failure index and stress factor are 0.869 and 0.822. The optimal weight of this configuration is 38.26 kg. This optimal weight is very close to the optimal weight of the configuration using straight stiffeners. The optimal curvilinear stiffeners become almost straight like in the optimal configuration of straight stiffeners. It should be noted that it is not possible to have a single hoop stiffener in the middle of the pipe using the parameterization of curvilinear stiffeners. These results shows that in torsional load case, it is better to have straight stiffeners in comparison to the curvilinear stiffeners for reducing the overall weight of the structure. The optimal values of the design variables are tabulated in Table 4.5.
4.6. Application and Results

Table 4.6: Range of Xshape for curvilinear stiffeners

<table>
<thead>
<tr>
<th></th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_j )</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>( n_k )</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>( \alpha_j ) (degrees)</td>
<td>22.5</td>
<td>90</td>
</tr>
<tr>
<td>( \alpha_k ) (degrees)</td>
<td>22.5</td>
<td>90</td>
</tr>
<tr>
<td>( h_s ) (m)</td>
<td>0.005</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Fig. 4.10 shows the optimal configuration using curvilinear stiffeners under bending load case. In this case, the buckling constraint is again active with the buckling factor of 0.997. The maximum failure index and stress factor are 0.123 and 0.916. The optimal weight is 32.20 kg. This optimal weight is 10.8% less than the optimal weight using straight stiffeners under bending load case. This shows that the use of curvilinear stiffeners is beneficial for saving weight in the structure in comparison to the use of straight stiffeners.

Table 4.7: Optimal final values of design variables for curvilinear stiffeners

<table>
<thead>
<tr>
<th></th>
<th>Torsional Load Case</th>
<th>Bending Moment Load Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_j )</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>( n_k )</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( \alpha_j ) (degrees)</td>
<td>87.53</td>
<td>22.50</td>
</tr>
<tr>
<td>( \alpha_k ) (degrees)</td>
<td>80.32</td>
<td>68.39</td>
</tr>
<tr>
<td>( h_s ) (m)</td>
<td>4.349E-02</td>
<td>1.015E-02</td>
</tr>
<tr>
<td>( t_s ) (m)</td>
<td>1.000E-03</td>
<td>1.000E-03</td>
</tr>
<tr>
<td>( t_p ) (m)</td>
<td>7.218E-04</td>
<td>6.187E-04</td>
</tr>
<tr>
<td>Mass (kg)</td>
<td>38.26</td>
<td>32.20</td>
</tr>
</tbody>
</table>
Figure 4.9: Optimal configuration using curvilinear stiffeners under torsional load case. (buckling factor = 1.004, maximum failure index = 0.869, stress factor (stiffeners) = 0.822, mass = 38.26 kg) the optimal cross-section dimensions are given in Table 4.7.

Table 4.8: Comparison of different configuration under bending load case

<table>
<thead>
<tr>
<th>Design Case</th>
<th>Total Mass (kg)</th>
<th>% Saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orthogonal</td>
<td>36.10</td>
<td>-</td>
</tr>
<tr>
<td>Curvilinear</td>
<td>32.20</td>
<td>10.8</td>
</tr>
</tbody>
</table>

4.7 Conclusions

A BLP optimization framework for designing stiffened pipes or pressure vessels with curvilinear stiffeners is developed. An integrated optimization framework utilizing the scripting based language Python, NURBS based Rhinoceros 3D, MSC.PATRAN and MSC.NASTRAN has been developed. It is seen that placement of the stiffeners have significant influence on the buckling load of the stiffened cylindrical surface and this can be beneficial for saving
weight by optimizing the stiffener’s cross-section and composite skin laminate thickness. The optimization technique utilizes the combination of Particle Swarm Optimization (PSO) and Gradient Based Optimization (GBO). For each particle of PSO, optimal size design is obtained using GBO (MSC.NASTRAN SOL 200) subjected to buckling, stress and composite skin laminate failure constraint. The framework can be used to design curvilinear stiffeners under different applied loads at the ends of the pipes. There could be many load cases beyond the scope of this chapter. There could be other gradient free optimization techniques, like Genetic Algorithm optimization, which could be alternative to the PSO in this framework. Two different approaches, orthogonally-placed stiffeners and curvilinearly-placed stiffeners on the surface of a cylindrical pipe, under torsional and bending load cases, have been considered to show the use of the presented framework. It is seen that in the torsional load case, it is beneficial to use straight stiffeners over curvilinear stiffeners for...
reducing the overall weight of the structure. However, in bending load case, it is better to use curvilinear stiffeners over the straight stiffeners. It is seen that curvilinear stiffeners have a potential of saving weight in the composite skin laminate by 10.8% and thus saving cost of material required for composite skin laminate, as compared to the case of using straight stiffeners.
5.1 Abstract

An important objective for the aerospace industry is to design robust and fuel efficient aerospace structures. Advanced manufacturing techniques like additive manufacturing have allowed structural designers to make use of curvilinear stiffeners for achieving better designs of stiffened plate and shell structures. Finite Element Analysis (FEA) based standard optimization methods for aircraft panels with arbitrary curvilinear stiffeners are computationally expensive. The main reason for employing many of these standard optimization methods is the ease of their integration with FEA. However, each optimization requires multiple computationally expensive FEA evaluations, making their use impractical at times. To accelerate optimization, the use of Deep Neural Networks (DNNs) is proposed to approximate the FEA buckling response, computed using MSC NASTRAN. The finite element model of a plate is verified with those found in the literature. Later, a Python script is used to generate a large data-set using parallel processing; and 80%, 10% and 10% of the generated data-set are used for training, validation and testing of DNNs, respectively. The results show that
DNNs, optimized using Adam optimizer, obtained an accuracy of 95% on the test set for approximating FEA response within 10% of the actual value. To compare the efficiency of the DNN, the trained DNN is used in the optimization of curvilinearly stiffened panels by replacing the conventional FEA. The DNN accelerated the optimization by a factor of nearly 200. The presented work demonstrates the potential of DNN-based machine learning algorithms for accelerating the optimization of bio-inspired curvilinearly stiffened panels.

5.2 Introduction

In the past several decades, significant research has been conducted on improving aircraft structures. An aircraft experiences large number of load cases while in flight \(^{[49]}\) and aerospace structures must be designed to sustain these load cases. The loads could result in overall panel buckling of stiffened panels, skin buckling between stiffeners, or crippling of stiffeners. This makes stiffened panels one of the important components to be designed during design and optimization of aircraft. To design such panels, many optimization techniques have been implemented in the literature. The optimization of complex aircraft panels can be conducted with the objective of minimizing weight while satisfying buckling and stress constraints \(^{[48]}\). This optimization is not that computationally expensive when the stiffeners are designed such that they are always placed straight in a grid. However, the required computational time grows significantly if the stiffener placement and its curvature are also considered as design variables during their design. Therefore, there is a need to have an efficient framework for optimization of such panels where both the shape (stiffener placement and curvature) and size (thicknesses of panel and stiffeners) can be designed efficiently. Singh et al. \(^{[48]}\) proposed a hybrid optimization technique for curvilinearly stiffened shells where both the shape and size design variables are considered. The technique is based
on the iterative use of a gradient-free optimization technique, the particle swarm optimization (PSO), for maximizing the buckling load and gradient-based optimization (GBO) for minimizing the weight while satisfying constraints. They made use of parallel processing to reduce the CPU run time. However, the optimizer still took about six hours to design one curvilinearly stiffened panel. This computational time could be reduced further using parallel processing, but would require enormous computational resources. In this chapter, we propose to use machine learning techniques to design such panels in minutes while satisfying constraints.

In past two decades, the applications of machine learning (ML) have grown significantly. Deep learning is a subset of the ML based on artificial neural networks (ANNs). The conventional machine learning techniques, like decision trees, support vector machines, random forest, etc., have limitations in their ability to learn from raw data [50]. These techniques may need a feature extractor, generated by human engineers, which can transform the raw data into a feature vector for providing it to a machine learning based classifier. This approach requires significant resources and time to be able to get the right feature vector. For example, in the context of the aircraft panel design, the raw data would include the dimensions of the panel structure along with load cases. The designer might have to find parameters like, area, second moment of area, volume, etc., to create the feature vector for training a classifier. The accuracy of the classifier could be low if the appropriate features are not provided. Thus, it could be difficult to find the right human generated features. However, deep learning methods helped in resolving this issue. The raw data can be provided directly to the deep learning model for creating a surrogate model to make the required prediction. They can be used to build a surrogate model without worrying about creating a feature vector. The deep neural networks (DNNs) in deep learning methods are essentially ANNs, but with many hidden layers. The multiple layers of a neural network automatically
discover the features needed for accurate detection or classification from data [50].

In the literature, many case studies have been presented where biologically motivated ANNs have been used as an efficient surrogate model to solve a variety of engineering design problems. Hajela and Berke [51, 52] showed the benefits of using ANNs in structural analysis and design including non-linear structural analysis. Rogers [53] successfully used neural networks for simulating structural analysis. It is stated that a fast, inexpensive neural network can be used as an alternative to a slow, expensive structural analysis program. Jenkins [54] used neural networks for approximating the analysis of structural grillages, which are used in structural design as a conceptual representation of concrete slab bridges. Their motivation to use a neural network is that it would help a bridge designer to experiment with a number of configurations of longitudinal and transverse members and with varying structural properties of the members. Kim and Kapania [55] employed ANNs in solving inverse problems in damage detection in structures. They showed the application on ANN in damage detection problems wherein a well-trained ANN provide both the location and the extent of damage. Kim et al. [56] used ANNs for solving inverse problems in the structural optimization of a fiber optic pressure sensor. They conducted parametric studies where the relation between the design parameters and sensor response was fulfilled by using ANNs. Marín et al. [57] conducted optimization of composite stiffened panels under mechanical and hygrothermal loads using neural networks and genetic algorithms (GA). Their studies showed that the use of ANN systems can increase the speed of the optimization by reducing the computational cost about 92.8%. But, the disadvantage of their approach lies in the time required to design the ANN architecture to obtain a sufficiently accurate surrogate model.

Mallela and Upadhyay [58] used ANNs for buckling load prediction of laminated composite stiffened panels subjected to in-plane shear. They created the data-set of ANN training using FEA. Their results show that the trained ANN can accurately predict the shear buckling load
of laminated composite stiffened panels and could be very useful in optimization applications. These case studies in the literature show that ANNs have great potential to accelerate optimization of engineering structures where computational efficiency is paramount.

The benefits of using bio-inspired curvilinearly stiffened panels motivated many researchers to design and manufacture such panels. These panels have shown great potential to save significant weight in comparison to the conventionally manufactured designs in aircraft. Kapania, Li and Kapoor [59] presented the use of curvilinear stiffeners for optimal design of stiffened panels. They stated that curvilinear stiffeners enhance the design space and may lead to lightweight designs in comparison to straight stiffeners under certain loading conditions. Mulani, Slemp and Kapania [60] developed a framework, EBF3PanelOpt, for design and optimization of complex, multi-functional aircraft panels subjected to complex structural loading cases. The framework is based on the use of commercial finite element analysis (FEA) packages, thus resulting in significant time consumption for designing curvilinearly stiffened panels. Hao et al. [61] proposed an efficient optimization framework of cylindrical stiffened shells with reinforced cutouts by curvilinear stiffeners. They used a surrogate model-based optimization to reduce computation time.

Queipo et al. [62] presented fundamental issues that arise in surrogate-based analysis and optimization (SBAO), with the focus on concepts, methods, techniques, and practical implications. Viana, Haftka and Watson [63] presented an efficient approach where multiple surrogate based technique is used for global optimization algorithm. They showed that their approach reduced substantially the number of cycles required for convergence. Mulani et al. [64] presented an optimal design of curvilinearly stiffened panel using response surface approaches (RSA). They showed that the RSA have the potential to improve the optimization. But, their implementation of RSA is limited to only square panels and it does not take different load cases into consideration. Also, their parameterization has limitations of being
dependent only on the $y$-coordinates of the stiffener end points, and surrogate models are based on coarse finite element mesh of the model. Sunny et al. [65] used an ANN residual kriging-based surrogate model for shape and size optimization of a stiffened Panel. They were able to save CPU run time by making use of ANNs. However, the optimization still takes significant time as a new ANN has to be trained every time a new panel has to be designed.

In the current chapter, we propose to make use of a single deep neural network (DNN) to predict buckling response for large number of panels with curvilinear stiffeners under large number of load cases. The present work eliminates many limitations by providing a single DNN, which is based on the data generated using refined mesh of the stiffened panel, and can handle different load cases in one surrogate model. Also, the presented parameterization of the stiffened panel can handle a bigger design space. This would avoid the need to create a new neural network whenever a new panel has to be designed, therefore, bringing down the optimization time from days to minutes.

In commercial industries, finite element method (FEM) based packages, e.g., MSC.NASTRAN, etc., are used to design different structures. These packages have the in-built capability to conduct gradient-based optimization to design size variables, like thicknesses, of different components. However, designing curvilinear stiffened panels is a non-convex optimization problem and, hence, requires gradient-free optimization techniques. The main reason for employing many of the standard optimization methods is the ease of their integration with the finite element analysis (FEA). But, these optimization techniques require a significant number of function evaluations to find out the optimal design. Thus, this optimization remains to be computationally expensive due to the requirement of multiple FEAs. In this chapter, the use of deep learning is proposed to accelerate the optimization of curvilinearly stiffened panels. The proposed technique can be easily integrated with commercial FEA
packages, making it possible to design curvilinearly stiffened panels in minutes.

The current research focuses on accelerating the optimization of curvilinearly stiffened panels using DNNs. Initially, a data-set is created using Latin Hyper-Cube Sampling (LHS). For each data point in the data-set, Rhinoceros 3D [16] is used for generating the geometry of the panels, including the surface of the panel and the placement of the stiffeners. Then, MSC.PATRAN and MSC.NASTRAN are used for generating the mesh and conducting the buckling analyses respectively. Parallel processing is used to reduce the time consumption required for data generation. Once enough data-sets are generated, a Python-based machine learning library, Keras [66] with Tensorflow [67] back-end, is used for training of a DNN. The trained DNN is then used in an integrated optimization framework to design curvilinearly stiffened panels. The framework is implemented with the motivation to find optimal designs under required loading conditions. The optimization results are compared with the cases where no DNNs are used. In Section 5.3, parameterization of curvilinearly stiffened panels, methodology for developing the optimization framework and training of the DNNs have been presented. In Section 5.4, the methodology is applied on few case studies to show the potential of using DNNs for accelerating the optimization. Finally in Section IV, the conclusion is presented.

5.3 Methodology

The present work focuses on accelerating the optimization of curvilinearly stiffened panels. The scripting language Python is used to integrate the optimization framework and training of DNNs. The optimization technique is based on a hybrid optimization technique where both the shape and size design variables are considered. This section covers the details about the parameterization of a curvilinearly stiffened panel, optimization technique, validation of
FEA results with the literature and training and testing of DNNs.

5.3.1 Parameterization

In the current work, the design variables are divided into two categories: shape design variables, $X_{shape}$, and size design variables, $X_{size}$. The shape design variables include the parameters for stiffener placement and its curvature. The size design variables include cross-sectional dimensions of the stiffener and panel’s thickness. The shape design variables are optimized using a gradient-free optimization technique (PSO) while the size design variables are optimized using GBO. The details of the design variables presented in this section follows the previous work conducted by Singh et al. [48].

Shape Design Variables

The shape design variables, $X_{shape}$, include two parameters: perimeter parameter, $\epsilon$, and curvature parameter, $\alpha$, to define the placement and curvature of a stiffener on a panel [48]. The perimeter parameter varies from 0 to 1 on the boundary of the panel. The curvature parameter varies in the perpendicular direction of the line joining the end control points of the stiffener. Figure 5.1 shows the parameterization of a stiffener on a square panel. The stiffener is defined using three control points based on NURBS (Non-Uniform Rational B-Splines). The two control points (A and B) are defined using perimeter parameter ($\epsilon_A$ and $\epsilon_B$). The third control point is defined using the curvature parameter, $\alpha$. The parameter $\alpha$ is non-dimensionalized with the outer length of the panel. Thus, each stiffener shape is defined by three design variables. The parameterization is implemented in a Python script that is executed using Rhino:Python in the CAD software, Rhinoceros3D [16]. Multiple stiffeners can be defined using this parameterization.
5.3. Methodology

In this work, the convex hull property of NURBS is used. It states that a curve always lies within the control polygon as shown in Fig. 5.1. A geometry check is developed, coded in the Python script of Rhino:Python, which ensures whether the third control point is on the required surface or not. This is implemented to ensure a successful generation of the geometry. If the third control point goes outside the panel surface during optimization, then a large penalty is added to the objective function.

In the current work, two stiffeners are considered for optimization case studies. A total of six shape design variables are considered for two stiffeners.

Figure 5.1: Parameterization of stiffener placement and curvature on a panel

Size Design Variables

The size design variables, $X_{\text{size}}$, include parameters to define the cross-section of the stiffener and thickness of a panel. In the current work, a rectangular cross-section stiffener is
considered. Figure 5.2 shows the size design variables. The $h_s$ and $t_s$ define the height and thickness of the stiffener. The $h_p$ defines the panel thickness. The stiffeners are modeled as beam elements in MSC.NASTRAN in the presented case studies.

![Design variables for defining a rectangular cross-section stiffener](image)

Figure 5.2: Design variables for defining a rectangular cross-section stiffener

### 5.3.2 Global Optimization Technique

In this section, the global optimization technique is presented. Figure 5.3 shows the flowchart of the global optimization. Singh *et al.* [48] implemented this optimization technique without the use of DNNs. A brief description of the flowchart is presented in this section. The overall optimization objective is to minimize the weight of the structure. During each iteration, two sub-optimizations are conducted. First, the size design variables, $X_{size}$, of the panel are kept fixed and only shape design variables, $X_{shape}$, are optimized using shape optimization using a gradient-free optimization with an objective to maximize the buckling load without any constraints. Subsequently, the shape-optimized design is fixed during the size optimization where only $X_{size}$ is optimized, under both buckling and stress constraints. The size optimization is conducted using a gradient-based optimization. The objective of the size optimization is to minimize the weight under buckling and stress constraints. Later, the optimally-sized design variables are fed back to the shape optimization. This way, the shape and size optimizations are conducted iteratively until convergence is achieved.
The motivation of dividing the optimization into two sub-optimizations comes from the reason that optimizing a curvilinearly stiffened panel is a non-convex optimization problem. If a gradient-based optimization is used for all the design variables, then the optimizer would most likely to get stuck in a local minimum. Therefore, in this optimization scheme, the design variables are divided into two categories, depending on the requirement if they need a gradient-based or a gradient-free optimization technique. Moreover, the shape optimization provides a design by maximizing the value of the constraint, buckling load, which later helps the size optimization to find the optimal design. Singh et al. [48] found out that most of the CPU run time is spent during the shape optimization. Parallel processing was used to reduce the CPU time. But, it still took about six hours to perform the optimization. This is due to the multiple FEA evaluations required during the shape optimization. In this work, the use of DNN is proposed to replace the FEA evaluations by predicting the buckling load of a panel in shape optimization. More details about the optimization scheme are presented in this section.

Shape Optimization

In the present work, Particle Swarm Optimization (PSO) [18] is used to conduct the shape optimization where the objective is to maximize the buckling load of the panel. The objective, as shown in Eq. 5.1, of the optimization is to minimize $F(X)$ by optimizing the shape design variables $X$ under no other constraints.
Figure 5.3: Flowchart of Global Optimization Procedure

\[
\begin{align*}
\text{minimize} & : F(X) = \frac{1}{\text{Buckling factor}} = \frac{\text{Applied Load}}{\text{Buckling Load}} \\
\text{subject to} & : a_j \leq X_j \leq b_j, \quad j = 1, 2, \ldots, n
\end{align*}
\]

(5.1)

Figure 5.4 shows the flowchart of the shape optimization using PSO. PSO is a gradient-free optimization technique. In this method, a finite number of random particles (known as designs) are defined over the required domain of design variables and the objective function is evaluated for every particle. The particles’ location is improved based upon their own and social differences [60], meaning based on a particle’s own best value and the global best value in the swarm. In the current work, shape design variables, \( X_{\text{shape}} \), are optimized using...
5.3. Methodology

PSO. For more details, [18, 48, 60] can be referred.

To evaluate the buckling load of a curvilinearly stiffened panel for each particle of the PSO, there are series of steps that are taken in the current work. Initially, Rhinoceros [16] is used to create the geometry of the panel. *RhinocerosPython*, a feature available in the software, is used to write a Python script to automate geometry generation with a different set of parameters during PSO. Once the stiffeners and panel geometry are successfully generated, the geometry is exported for use in MSC.PATRAN. MSC.PATRAN is used to generate a mesh along with loading and boundary conditions using a session file. Finally, finite element analysis (FEA) is conducted in MSC.NASTRAN to evaluate the buckling load of the panel. Most of the CPU time for the optimization is spent in conducting the shape optimization because of the multiple evaluations of the FEA.

To reduce the time consumption, the use of DNN is proposed as shown in Fig. 5.4. The DNN is used such that it can automatically predict the buckling load of a panel based on the inputs from the PSO. This eliminates the need to execute the FEA multiple times, thus, saving significant computational and time resources. Once the shape optimization is complete, the optimal shaped design is fed to the size optimization.

**Size Optimization**

In this chapter, a gradient-based optimizer (GBO), MSC.NASTRAN SOL 200, is used to conduct the size optimization of the panel. The size design variables are optimized with the objective to minimize the mass of the panel. The objective, as shown in Eq.5.2, is to minimize the mass of the panel, \( G(Y) \), for a given range of size design variables, \( Y \), under buckling and stress constraints. One of the advantages of using GBO in the present optimization scheme, after the optimal shape design is provided by the DNN in the shape optimization, is
that it ensures that the final optimal design would be evaluated accurately using FEA. The DNN could have some error in the final prediction of the buckling load, but the use of FEA at the end ensures the correctness of the final results.

\[
\begin{align*}
\text{minimize} & \quad G(Y) = \text{Mass} \\
\text{subject to} & \quad c_p \leq Y_p \leq d_p, \quad p = 1, 2, \ldots, m \\
& \quad \text{Buckling Factor} \geq 1 \\
& \quad \text{von Mises stress } (\sigma) \leq \text{yield stress } (\sigma_y)
\end{align*}
\]  

(5.2)
5.3. Methodology

5.3.3 Validation of Finite Element Analysis Results

To ensure that a stiffened panel is correctly developed, the finite element model of the stiffened plate is validated with the literature. Timoshenko and Gere [68] presented the buckling parameter for a stiffened plate with a longitudinal rib in the middle of the plate. They defined three parameters as:

\[
\beta = \frac{a}{b}; \quad \gamma = \frac{EI_s}{bD}; \quad \delta = \frac{A_s}{bb_p}
\]

where, \(a\), \(b\) and \(h_p\) are length, width and thickness of the plate respectively. The \(A_s\) and \(I_s\) are the area and second moment of area of the stiffener respectively; and \(E\) and \(D\) are the Young’s modulus and flexural rigidity of the plate, respectively.

The buckling parameter is defined as:

\[
\text{Buckling Parameter} = \frac{\lambda b^2}{\pi^2 D}
\]

where, \(\lambda\) is the buckling load of the plate. Figure 5.5 shows the comparison of the buckling parameter from the FEM model and literature for different values of \(\beta\), \(\gamma\) and \(\delta\).

5.3.4 Deep Neural Networks

This section presents the details about the deep neural networks (DNNs) used in the current study. DNNs are extensively used in deep-learning methods. They have been successfully used in the literature for accurately conducting classification and regression tasks. In classification tasks, DNNs are used to predict a class label out of many classes based on required input. For example, if an image is provided as an input, the DNN could be used to classify
Verification of FEM Model with Literature

• The FEM model was verified with literature results
  • Timoshenko and Gere*:
  
  – Plate stiffened by longitudinal rib
  – Dimensional variables:
    – Compared buckling parameters

\[ \gamma = 15 \]
\[ \delta = 0.05 \]
\[ \gamma = 25 \]
\[ \delta = 0.05 \]


Figure 5.5: Validation of FEA results with literature

the image in certain required labels. In regression tasks, DNNs are used to predict continuous values of outputs. In the current study, DNNs are used to predict the buckling load of a curvilinearly stiffened panel and, thus, used as a regression task. There are many categories of deep-learning methods where DNNs are used such as supervised learning, unsupervised learning, reinforcement learning, etc. The present study focuses on the use of supervised learning to predict the buckling load.

A typical DNN has a set of inputs and outputs along with a number of hidden layers as shown in Fig. 5.6. The DNN creates a mapping between a set of inputs and output. Each layer in the DNN has a set of neurons as shown in Fig. 5.6. Each neuron in a hidden or output layer applies an activation function to a weighted sum of the outputs of the previous layer. In supervised learning, there are fixed sets of inputs and outputs for a DNN. Initially, a data set is generated, often called ground truth data, for training, validation and testing of the DNN. This data-set is used to then build a DNN.

In the current work, Keras [66] with the Tensorflow [67] backend is used for building DNNs. The Keras is a high-level neural networks application program interface (API), implemented in Python and can run with the backend of TensorFlow. Keras and Tensorflow are an open
source machine learning libraries for research and production.

There are a series of steps that need to be taken to build a DNN. Firstly, the user has to define the architecture of the network, which includes the number of hidden layers, neurons in each layer (shown in green color in Fig. 5.6), and activation functions. The network’s weights could be randomly initialized with certain values. Each neuron in hidden or output layer takes weighted sum of one or more inputs from previous layer. Then, a non-linear function, known as an activation function, is applied to the weighted sum, resulting into the neuron’s output or value. The training of the network is conducted in two phases: forward pass and backward propagation. During the forward pass, an input is provided to the network. All the values of neurons in the network are calculated based on the weights of the network. At this stage, there would be a high error in the predicted output for an input. This error or loss is then used to update the weights of the network. This step is called the backward propagation step. The mean squared error has been successfully used as a loss function in the literature for many different problems as shown in Eq. 5.5.

\[
\text{Loss} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2
\]  

(5.5)
where \( y_i \) and \( \hat{y}_i \) are the FEA based buckling parameter and DNN predicted buckling parameter respectively; \( n \) is the total data points in the dataset.

To update the weights, the gradient of the loss function is computed with respect to all the weights. The weights are then changed in the negative direction of the gradients. This helps in reducing the error between the actual output and the predicted output [50].

Once all the training examples are passed to the network (called an epoch), the average of all the gradients can be calculated to update the weights of the network. This approach has a disadvantage of having a possibility for the optimizer get stuck in a local minimum. There exists an alternative method to this approach, called mini-batch stochastic gradient descent (SGD), where instead of updating the weights at the end of all the training examples, the weights are continuously updated after small batches of training set pass through the network [50]. The SGD has been shown in the literature to be less prone to reach local minima than standard gradient descent. Recently, a new optimizer called Adam [69] has been proposed for stochastic optimization. The method is based on adaptive learning rates for different parameters and can be used with mini-batch based training. It has been shown to be robust and well-suited to a wide range of non-convex optimization problems in the field machine learning [69]. The optimizer is available in the machine learning library, Keras, for training of neural networks. In the current work, Adam is used to train the DNNs.

One of the important things to avoid during DNN training is over-fitting. To find the right architecture of the DNN, different configurations of DNN are tested. The data-set is divided into training and testing sets. The number of hidden layers and neurons in these layers is varied from small to higher values while keeping other hyper-parameters of the network fixed. During this implementation, once the neural network starts becoming bigger or denser than required, either by increasing the number of hidden layers or by increasing the number of neurons in hidden layers, the network starts to over-fit on the training set and perform badly
on the testing set. The best combination of the number of hidden layers and neurons are picked based on the accuracy on the test set. The accuracy of the DNN could also decrease beyond certain number of hidden layers or neurons because of insufficiency of the data to train all the weights.

Once the appropriate architecture is found, other hyper-parameters can be tuned, e.g., optimal number of epochs required for training of DNN. During the training phase, the weights are continuously updated to increase the accuracy over the training set. However, after certain epochs or iterations, the training accuracy starts increasing significantly such that the network would have excellent performance on the training set, but would perform very badly on any instance that is outside of the training set. Therefore, the data is split into training, validation and testing sets. The training set is used for training of the DNN. During the ongoing training of DNN, its accuracy is continuously checked on the validation set. The accuracy on the validation set helps in defining the stopping criteria for the training of DNN. Once the network’s weights are finalized, its accuracy is checked over the testing set. It should be noted that the testing set is always kept outside from training and validation.

5.3.5 Problem Description

In this section, the problem statement is discussed. The main motive of using a DNN in the current work is to predict the buckling parameter of a curvilinearly stiffened panel. From the buckling parameter, the actual buckling load of the panel can be found as shown in Eq. 5.4. The panel is considered under simply supported boundary conditions under bi-axial loading conditions as shown in Fig 5.7. The input vector for the DNN consists of the following eleven non-dimensionalized parameters:

1. Stiffener I shape design variables: $\epsilon_{A1}$; $\epsilon_{B1}$ and $\alpha_1$
2. Stiffener II shape design variables: $\epsilon_A$, $\epsilon_B$, and $\alpha_2$

3. Stiffener size design variables: $t_s/b$, $h_s/h_p$

4. Panel size design variables: $h_p/b$

5. Panel dimensions: $a/b$

6. Loading ratio: $N_y/N_x$

The output of the DNN is considered to be the buckling parameter of the panel as shown in Eq.5.4. The benefit of using this parameter as an output is that this is a non-dimensional parameter, hence, it could help the DNN to predict the buckling load of a large number of curvilinearly stiffened panels.

![Figure 5.7: Problem description for training of DNNs](image)
5.4 Application and Results

In this section, the methodology explained in Section 5.3 is applied on training of optimal DNN and optimization case studies. Initially, a data-set is generated using Latin Hypercube Sampling (LHS). Table 5.1 shows the lower and upper bounds of input variables used for creating the data-set in the present study. There are a few constraints added during the data-set generation. It can be seen from Fig. 5.7 that the stiffener shown in the red color can still be successfully made even if $\epsilon_A$ can be inter-changed with $\epsilon_B$. Therefore, a constraint is added in generating the data-set such that $\epsilon_A$ is always greater than $\epsilon_B$. If during the shape optimization, PSO provides the value of $\epsilon_B$ to be greater than $\epsilon_A$, then their values are inter-changed for providing an input to a DNN. Also, there can be cases where $\epsilon_A$ and $\epsilon_B$ are very close to each other, resulting in the failure of the geometry and mesh generation of the stiffener or resulting in very small stiffener along the corner of the panel. To avoid such an issue, another constraint is added in the data-generation such that $\epsilon_A$ and $\epsilon_B$ could not come closer than a certain value. In the current work, this value is chosen to be 0.3 with the motivation to avoid generation of undesirable small stiffeners near the corners or along the edges. This is important, otherwise it could result in generation of outliers for DNN training, thus worsening the accuracy of the network.

In the current study, a curvilinearly stiffened isotropic panel is considered under bi-axial loading conditions as shown in Fig. 5.7. For each point in the data-set, Rhinoceros 3D is used for generating the geometry, MSC.PATRAN and MSC.NASTRAN are used for generating the mesh and conducting buckling analyses respectively. To save computation time on the data generation, parallel processing is used.
<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon_A$</td>
<td>0.25</td>
<td>1.00</td>
</tr>
<tr>
<td>$\epsilon_B$</td>
<td>0.00</td>
<td>0.75</td>
</tr>
<tr>
<td>$\alpha/2(a+b)$</td>
<td>-0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>$a/b$</td>
<td>0.6</td>
<td>2.0</td>
</tr>
<tr>
<td>$h_p/b$</td>
<td>0.005</td>
<td>0.03</td>
</tr>
<tr>
<td>$h_s/h_p$</td>
<td>2.0</td>
<td>11.0</td>
</tr>
<tr>
<td>$t_s/b$</td>
<td>0.004</td>
<td>0.04</td>
</tr>
<tr>
<td>$N_y/N_x$</td>
<td>0.1</td>
<td>2.0</td>
</tr>
</tbody>
</table>

### 5.4.1 Training of Deep Neural Networks

Once all the data-sets are generated, a Python-based machine learning library, Keras [66], is used for training of DNNs. The Virginia Tech Advanced Research Computing (ARC) facilities are used to train the DNNs. The facilities have two IBM Power8 CPU (3.26 GHz) with 256 GB of memory along with four NVIDIA P100 GPU with 16 GB of memory each. The computation time for training a single DNN could vary from 0.5 hour to 1.5 hour depending upon the size of the DNN. A prediction of DNN is considered to be accurate if the predicted buckling load is within 10% of the actual value.

Initially, a data-set of 50,000 data points is generated. To find the right architecture of the DNN, the data-set is divided into training (90%) and testing set (10%). Different configurations of the network are tested by varying the number of hidden layers and the number of neurons in them; by fixing the number of epochs to be 1000. Figure 5.8 shows the variation of the accuracy of training and testing set by changing the number of hidden layers. In this configuration, the number of neurons is 90 in each of the hidden layers. If the number of neurons is increased beyond 90, the testing accuracy starts to decrease. The optimal combination is found to be 8 hidden layers with 90 neurons in each. It can be seen that the
training accuracy increases while testing accuracy decreases beyond a certain value of the number of hidden layers. The best testing accuracy is found to be 87%.

![Graph showing variation of accuracy with data set of 50,000 data points]

Figure 5.8: Variation of accuracy with data set of 50,000 data points

To see the possibility of increasing the testing accuracy, the data-set is increased to 100,000 data points. Again, different configurations of DNN are tested by fixing the number of epochs to be 1000. Figure 5.9 shows the variation of training and testing accuracy. It can be seen that by increasing the size of the data-set, the DNN is able to better adjust its weights such that testing accuracy increases. The testing accuracy doesn’t change much when the number of layers changes from 7 to 9. Thus, the optimal number of layers found here is 7. The network starts to over-fit after exceeding a certain number of hidden layers.

Finally, the data-set is increased to 190,000 data points with the motivation to see if the DNN’s accuracy can be increased further. Figure 5.10 shows the variation of training and testing accuracy with different configurations of DNN. the optimal number of layers found here are to be 6. The network starts to over-fit beyond this value. Therefore, for further analysis, the configuration with 90 neurons in each of the 6 hidden layers is fixed.

Once the architecture of the DNN is fixed, it is important to find the right number of epochs
To increase the accuracy of DNNs, a dataset increased to 100,000 cases generated using Latin Hypercube Sampling. The data set divided as: 80% for testing, 10% for validation, 10% for testing. Conducted extensive computational experiments. The accuracy of test set increased from 87% to 92%. Trainable parameters 50,311 ≈ 92%.

Figure 5.9: Variation of accuracy with data set of 100,000 data points

To increase the accuracy of DNNs even more, a dataset increased to 190,000 cases. The accuracy of test set increased to 95%. The trained neural network of 6 layers with 90 neurons in each layer can be used in optimization. Total trained parameters: 42,121 ≈ 95%.

Figure 5.10: Variation of accuracy with data set of 190,000 data points

for DNN training. Therefore, the data-set of 190,000 data points is divided into training, validation and testing set with the ratio of 80%, 10% and 10% respectively. The validation set is used as a stopping criteria for the training of DNN. Figure 5.11 shows the variation of mean squared error of the DNN over the dataset with respect to the number of epochs. The optimal number of epochs are found to be 1255. It can be seen that beyond this value of epochs, the validation loss starts to increase slowly, but training loss decreases significantly.
Therefore, the optimal weights in DNN are chosen to be the one at the optimal number of epochs. The accuracy of DNN at optimal number of epochs is about 95%. This DNN is used to conduct the case studies in the next sections.

Figure 5.11: Variation of loss function of training and validation set with number of epochs

This trained DNN is also used to evaluate the percentile based accuracy where the accuracy is evaluated for different % error criteria. In the above presented studies, a prediction is considered accurate if it is predicted within a limit of 10% of the actual value. The trained DNN is used to evaluate a percentile based accuracy where the limit of 10% is changed to a different percentage value. Table 5.2 shows the accuracy of the trained DNN over the full dataset if the limit to consider a prediction to be accurate is changed to 1.0%, 2.5%, 5.0% and 7.5%. It has been seen in the optimization case studies, presented in the next section, that the DNN trained based on 10% error limit is enough to help the optimizer come close to the global optima. Later, the GBO is used to conduct the final size optimization using actual FEA. This helped in reducing the computation time significantly.

To further analyze the accuracy of the trained DNN, contour plots are generated to compare
Table 5.2: Percentile based accuracy of DNN over the dataset of 190,000 data points

<table>
<thead>
<tr>
<th>Abs. % Error Limits</th>
<th>Training Set</th>
<th>Testing Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>24.40</td>
<td>19.06</td>
</tr>
<tr>
<td>2.5</td>
<td>54.50</td>
<td>44.56</td>
</tr>
<tr>
<td>5.0</td>
<td>82.54</td>
<td>74.19</td>
</tr>
<tr>
<td>7.5</td>
<td>93.41</td>
<td>89.22</td>
</tr>
<tr>
<td>10.0</td>
<td>97.37</td>
<td>95.21</td>
</tr>
</tbody>
</table>

the variation of trained DNN and actual FEA variation in the design space. Figure 5.12 shows the comparison of buckling parameter with respect to aspect ratio \((a/b)\) and height of the stiffener \((h_s/h_p)\). Figure 5.12(a) and (b) are generated using FEM and DNN respectively. Both the figures are plotted using same contour levels. Figure 5.12(c) shows the contour plot of % error in the prediction. The maximum error in the prediction is 10.04%. Similarly, Fig. 5.13 shows the comparison of buckling parameter with respect to the non-dimensional thickness \((t_s/b)\) and non-dimensional height of the stiffener \((h_s/h_p)\). Figure 5.13(c) shows the contour plot of % error in the prediction. The maximum error in the prediction is 9.45%. These comparison studies shows that the trained DNN is able to closely predict the actual variation of the buckling parameter.

![Figure 5.12: Variation of the buckling parameter with respect to aspect ratio of panel and height of the stiffener using FEA and DNN](image-url)
Polynomial-based Surrogate Model

To compare the benefits of using DNN over another surrogate model, polynomial based surrogate models are generated. In this study, the same dataset of previous section is used. The accuracy of training and testing set are computed by conducting polynomial regression over different degree of the polynomials. Figure 5.14 shows the comparison of training and testing accuracy with different sizes of dataset and degree of the polynomials. In Fig. 5.14(a), a dataset of 1,000 data points is used. It can be seen that as we increase the degree of the polynomial for fitting the dataset, it starts to overfit the data with 100% accuracy on training set, but the accuracy over the testing set decreased significantly. As expected, similar behaviour is seen in the Fig. 5.14(b) and Fig. 5.14(c). The best accuracy on the training set is close to 100%, but the best accuracy over testing set obtained in this study is 64.9%. This shows that the simple polynomials are not able to learn the complex variation of buckling parameter accurately using the same dataset that is used to train the DNNs. The use of non-linear activation functions help the DNNs to better approximate the complex function in comparison to the simple polynomials. With the use of DNNs, it became possible to achieve a higher testing accuracy of 95% over the same dataset.
5.4.2 Case Studies

In this section, the trained DNN is used in conducting the optimization of the curvilinearly stiffened panels. Two case studies are presented. The details about load cases and panel dimensions, taken from [60], are shown in Table 6.2. The global optimization presented in Section 5.3.2 is implemented. The optimization time is compared for the cases where the analyses are conducted sequentially, in parallel or by using DNN in the PSO iterations. Figure 5.15 shows the comparison of the CPU run-time with different methods. The DNN-based optimization accelerated the optimization by a factor of 200 in comparison to the case where FEA analyses are conducted sequentially. The approximate time for running the optimization using parallel processing is taken from the authors’ previous work [48]. The actual CPU run-time would depend on available computational resources. The parallel processing approximate time is added here to give an idea to the reader about the comparison of CPU run-time. The benefit of using a DNN is that the optimization could be ran on a simple desktop machine without the need of significant high performance computing resources.
### Table 5.3: Load cases and panel dimensions for case studies

<table>
<thead>
<tr>
<th></th>
<th>$N_x$ (kN/m)</th>
<th>$N_y$ (kN/m)</th>
<th>$a$ (m)</th>
<th>$b$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case Study #1</td>
<td>275.83</td>
<td>496.49</td>
<td>0.7112</td>
<td>0.6096</td>
</tr>
<tr>
<td>Case Study #2</td>
<td>454.50</td>
<td>513.24</td>
<td>0.5080</td>
<td>0.4064</td>
</tr>
</tbody>
</table>

### Optimization Results

- **Objective:** To compare optimization results of weight minimization of stiffened panel using DNN and using conventional FEA
- **Panel Dimensions:**
  - $a = 0.5080$ m
  - $b = 0.4064$ m
- **Applied Loading:**
  - $N_x = 454.50$ kN/m
  - $N_y = 513.24$ kN/m
- **Time consumption comparison:**
  - Sequential: $\approx 0.5$ hr
  - Parallel: $\approx 6$ hr
  - With DNN: $\approx 0.5$ hr

![Figure 5.15: Comparison of CPU run time for conducting optimization sequentially, with parallel processing and with use of a DNN](image)

**Case Study - I**

In this section, the results from the first case study are presented. The load case, as shown in Table 6.2, is considered. During shape optimization, six design variables related to the stiffener placement and curvature are considered. A swarm of 120 particles is considered for PSO optimization. Figure 5.16 shows the comparison of the optimization history for the cases where actual FEA is used or if the trained DNN is used in the PSO iterations. It can be seen that in the case where actual FEA is used in PSO, the optimal design is found at the 4th global iteration. The optimal mass of this configuration is 7.74 kg. The buckling constraint is active, meaning buckling factor is close to 1.0, and the stress constraint is inactive for the optimal design. On the right side of the Fig. 5.16, the DNN-PSO based iteration history shows that the optimal design is again found after 4th global iteration. The optimal design has a mass of 7.99 kg. The stiffener placement is very similar to the case where actual FEA is used in the PSO. As the loading is higher in magnitude in $y$-direction, it is understandable...
that the optimizer tried to put both the stiffener in that direction. In this design also, the buckling constraint is active while stress constraint is not. Figures 5.17 and 5.18 shows the first buckling model and the von Mises stress distribution of optimal design found by using FEA and DNN in the PSO iterations respectively. The maximum von Mises stresses are 71.9 MPa and 70.6 MPa in the optimal designs found by using FEA and DNN. This shows that the trained DNN has the potential to save significant computational resources while providing a very close optimal design.

![Graph showing optimization history]

Figure 5.16: Comparison of optimization history of case study I: (Left) Using FEA in PSO iterations, (Right) Using DNN in PSO iterations

Case Study - II

In this section, the results from the second case study are presented. The 2\textsuperscript{nd} load case, as shown in Table 6.2, is considered. For shape optimization, a swarm of 120 particles is considered for PSO optimization. Figure 5.19 shows the comparison of the optimization history for the cases of using FEA or DNN in the PSO iterations. The optimal design is found at the 7\textsuperscript{th} global iteration if FEA is used in the PSO. The optimal mass of this configuration is 3.26 kg. The buckling constraint is active while the stress constraint is inactive for the
Comparison of Optimal Design

Buckling Factor: 0.995

Buckling Factor: 0.993

Max von Mises Stress: 71.9 MPa

Max von Mises Stress: 70.6 MPa

Mass: 7.74 kg

Mass: 7.99 kg

Figure 5.17: Optimal design in case study I by using FEA in PSO iterations. Left: Geometry, Middle: 1st Buckling mode, Right: von Mises stress

Figure 5.18: Optimal design in case study I by using DNN in PSO iterations. Left: Geometry, Middle: 1st Buckling mode, Right: von Mises stress

optimal design. Figure 5.19 also shows the DNN-PSO based iteration history. The optimal design is found after the 5th global iteration. The optimal design has a mass of 3.27 kg. The optimal stiffener placement is not very similar to the case where actual FEA is used in PSO, but the optimal mass is almost equal. In this design also, the buckling constraint is active while stress constraint is not. Figures 5.20 and 5.21 shows the first buckling modes and the von Mises stress distribution of the optimal design found by using FEA and DNN in the PSO iterations, respectively. The maximum von Mises stresses is 112 MPa in both
the optimal designs found by using FEA and DNN; and the buckling factor is close to 1.0 in both the designs.

Figure 5.19: Comparison of optimization history of case study II: (Left) Using FEA in PSO iterations, (Right) Using DNN in PSO iterations

Figure 5.20: Optimal design in case study II by using FEA in PSO iterations. Left: Geometry, Middle: 1st Buckling mode, Right: von Mises stress

The cases studies conducted in this section shows that DNNs can be successfully used for optimization of curvilinearly stiffened panels. They not only save significant computational resources, but also help in getting designs close to optimal designs. Moreover, the presented approach makes sure that all the constraints are satisfied in the final design as GBO is conducted in the final step of every PSO iteration to satisfy all the required constraints.
In this chapter, a DNN-based integrated optimization framework is presented for optimization of curvilinearly stiffened panels. The framework is based on a hybrid optimization technique of gradient-free optimization (PSO) and gradient-based optimization (GBO) techniques. The PSO optimizes the placement and the curvature of the stiffeners on a panel with the objective of maximizing the buckling load of the panel. This design is fed into GBO for optimizing the size design variables. This back and forth iteration between PSO and GBO continues until the required convergence is achieved. Most of the CPU time consumption during the optimization takes place by PSO for conducting multiple FEA evaluations. In the current work, the use of DNN is proposed to replace the FEA analyses evaluations.

A large data-set is created for training, validation and testing of DNNs. The optimal DNN’s testing accuracy is found to be 95%. Even though this DNN took significant computational resources, it was a one time cost and this DNN could be used for evaluating the buckling load of large number of curvilinearly stiffened panels under large number of load cases. Two case studies are presented where a panel, stiffened by two curvilinear stiffeners, is optimized.
The optimization studies show that DNN can be successfully used in the optimization. The DNNs accelerated the optimization by the factor of nearly 200 while providing a design that is very close to the FEA-derived optimal design.
Chapter 6

Accelerated Structural Design and Optimization using Active Learning

6.1 Abstract

Optimization of complex structures using standard evolutionary algorithms, like Genetic Algorithm (GA), is known to be computationally expensive, since a very large number of Finite Element Analyses (FEA) must be conducted for each possible structural design during the optimization. In the literature, different surrogate models have been used to replace the FEA for improving the computational performance of the structural optimization. However, the optimal solution found by the optimizer often depends on the accuracy of the surrogate model itself. Moreover, large datasets may be needed for getting the required accuracy of the surrogate model. An adaptive machine learning technique called active learning is used in this chapter to accelerate the evolutionary optimization of complex structures. An active learner is a machine learning-based model which can interactively query the outputs of certain data points, whenever the model would be uncertain about those outputs. In the presented approach, the active learner helps the GA by predicting whether the possible design is going to be a feasible one or not, meaning whether it satisfies the required constraints or not. In an active learning based approach, If the learner is uncertain about the output, an actual FEA is conducted, and the results are used to improve the learner’s accuracy for future
evaluations. The approach does not need a trained surrogate model prior to the optimization. The active learner adaptively learns about the structure during the optimization to improve the computational performance. The approach is used to optimize: a ten-bar truss problem, the Hesse function, and buckling of curvilinearly stiffened panels under in-plane loads. The results show that the approach has the potential to reduce the total required constraint evaluations by more than 50%.

Nomenclature

\( k \) = Number of constraints
\( \xi \) = Number of design variables
\( \mu \) = Population size of GA
\( \lambda \) = Offspring size of GA
\( p \) = Probability of a prediction
\( \gamma \) = Confidence parameter
\( \beta \) = Threshold value for \( \gamma \)
\( \phi \) = Number of estimators in active learner
\( m \) = Queries data-size collected before re-training active learner
\( P \) = Applied load on the ten-bar truss

6.2 Introduction

Evolutionary algorithms (EAs), like Genetic Algorithm (GAs) and Particle Swarm Optimization (PSO), have been successfully used in the literature for carrying out large scale
non-convex optimization problems. The GAs were developed by John Holland, his colleagues, and his students at the University of Michigan [70, 71, 72, 73]. The GA’s are capable of finding near-optimal solutions in large design spaces. They are essentially search algorithms based on the concept of natural selection and natural genetics. They have been successfully used in structural design and optimization problems. Jenkins [74] investigated the use of GAs towards structural optimization. Riche and Haftka [75] conducted buckling load maximization of composite laminate by optimizing laminate stacking sequence using GA. Rajeev and Krishnamoorthy [76] used GAs for optimizing structures with discrete design variables. They stated that GAs are very useful for structural optimization problems where it is difficult to compute gradients of the objective functions and constraints. However, GAs could have the disadvantage of being slow if the evaluation of the objective function or constraints is computationally expensive. For example, GAs can be integrated with Finite Element Analysis (FEA) software packages for evaluating the FEA responses. This integration with the FEA results in the requirement of enormous computational and time resources to conduct the simulation responses.

Ong, Nair and Keane [77] stated that high computational costs due to high-fidelity simulation models can have issues for the successful implementation of EAs for engineering design optimization. They conducted optimization of computationally expensive problems by making use of surrogate models. Zhou et al. [78] presented an approach where global and local surrogate models are combined to accelerate evolutionary optimization algorithms. At a local level, they used computationally cheap radial basis function (RBF) surrogate models to conduct the gradient-based search. Nik et al. [79] conducted surrogate-based evolutionary optimization of a composite laminate with curvilinear fibers. Mulani et al. [64] used response surface approaches for evolutionary optimization of stiffened panels with curvilinearly stiffeners. Salajegheh and Gholizadeh [80] presented an approach for finding
the optimal design of structures by an improved GA. They presented a modified version of standard GA, called virtual sub-population method (VSP), along with the use of artificial neural networks (ANNs). Jansson, Wakeman, and Månson [81] conducted optimization of hybrid thermoplastic composite structures using surrogate models and GAs. They used an approximate model to replace the FEA in the optimization. The surrogate model is pre-trained on the dataset created using FEA simulations. They stated that the successful implementation of their approach requires the right approximation algorithm as well as the right size of training data for the training of the surrogate model.

The pre-training of surrogate model for use in the evolutionary optimization could also be computationally expensive due to the requirement of large labelled datasets for training of the model. Moreover, it could be difficult to get a surrogate model which is highly accurate over a large design space, if the design space is non-convex with many local minima. An alternative to this could be an adaptive approach where the surrogate model could adaptively improve during the optimization to improve the computational performance. In the literature, different adaptive optimization schemes have been implemented to accelerate computationally expensive optimizations. Wang, Dong and Aitchison [82] presented an adaptive response surface method (ARSM)-based global optimization scheme for computation-intensive design problems. The ARSM creates quadratic approximation models for the objective function in a gradually shrinking design space. The method showed the benefits of not getting stuck in one of the local minima. Steenackers, Presezniak and Guillaume [83] also developed an ARSM method for high-dimensional design problems. The response model to be optimized is not built from a pre-defined number of design experiments, but is adapted and refined during the ongoing optimization. They tested their approach on a FEA based optimization while showing significant potential to save CPU run time. Gurav and Kapania [84] developed response surface method for optimization of curvilinearly stiffened panels. They update the
response surface during the EA-based optimization. Lagaros, Charmpis, and Papadrakakis [85] presented an adaptive neural network strategy for improving the computational performance of GA-based structural optimization. They showed that the use of ANNs to predict the feasibility or infeasibility of structural designs while avoiding expensive FEA evaluations. Their approach is adaptive in a way that the utilized ANN configuration is updated as the evolutionary optimization evolves by performing ANN re-training using the data gradually accumulated during the optimization. During their optimization, if an ANN makes a prediction for a design to be feasible, it is verified with an actual FEA evaluation. But, if the ANN predicted a design to be infeasible, they discard the design without verifying with the actual FEA evaluation. Their method may have a disadvantage that the approach might discard a good design if the ANN made a mistake in its prediction.

The current work focuses on developing a new method which may help in eliminating the disadvantages of the approaches mentioned in the literature. We propose to make use of an adaptive machine learning technique, called active learning, to accelerate a constrained GA-based structural optimization. An active learner is a machine learning-based model which can interactively query the outputs of certain data points whenever the model would be uncertain about those outputs. This is different from supervised learning where the model has access to all the available labelled datasets, i.e. the input-output pairs. It has an advantage over supervised learning in which the creation of the labelled dataset for training of the model could be computationally expensive [86, 87]. In this approach, the active learner has the freedom to actively choose the training data. The learner only queries the output of certain data-points when it can learn something new and improve its performance. This helps in eliminating the need for creating large training datasets.

There are different strategies in active learning to query the required output of the unlabelled dataset. *Query by committee* is one of the many strategies used in the literature for active
learning. This strategy is based on the use of an active learner which is an ensemble or committee of many surrogate models. The learner only queries the output for those data-points which have the most disagreement among the committee [88]. Abe and Mamitsuka [89] presented a query by committee strategy using bagging approach and called it query by bagging. Bagging [90, 91] is an ensemble-based learning which has multiple base surrogate models, each trained on random subsets of the original dataset, and the final prediction is made by taking aggregated votes from all the surrogate models. It has been shown in the literature that bagging significantly improves the predictive power of the machine learning models. Therefore, in this chapter, the query by bagging strategy is used for the active learner. Goel et al. [92] also presented an ensemble based surrogate modeling. They showed the benefit of the approach for identifying regions of high errors at locations where predictions of the surrogate models widely differ.

In the current work, active learning is used for accelerating the structural design and optimization. Initially, an active learner is trained using a small dataset collected during the first generation of the GA by conducting actual constraint evaluation. Later, during the optimization, the active learner helps the GA by predicting if the required design would be feasible or not. If the learner is uncertain about its output, then an actual constraint evaluation is conducted to check the feasibility of the design. The methodology for determining the uncertain points is presented in Section 6.3.3. These uncertain data points are collected for retraining of the learner, so that it can become more confident in its predictions for future evaluations in the optimization. Section 6.3 presents the details of the methodology of the proposed approach. Section 6.4 shows the application of the approach on three different constrained optimization problems: a ten-bar truss problem, the Hesse function, and curvilinearly stiffened panels. The results show that the approach has the potential to reduce the total required constraint evaluations by more than 50%.
6.3 Methodology

This section presents the methodology for the proposed approach of using active learning for improving the computational performance of the GA. Initially, a brief description of the implemented GA without active learning is shown. Later in the section, the new proposed approach of using active learning for helping the GA is presented.

6.3.1 Genetic Algorithm for Constrained Structural Optimization Problem

A constrained structural optimization problem can be formulated as:

\[
\begin{align*}
\text{minimize} & : F(X) = \text{Mass} \\
\text{subject to} & : g_j(X) \leq 0, \quad j = 1, 2, \ldots, k \\
& : c_h \leq X_h \leq d_h, \quad h = 1, 2, \ldots, \xi
\end{align*}
\]

where \( F(X) \) and \( g_j(X) \) are the objective function and \( k \) constraints respectively, and \( \xi \) is the total number of design variables. There are different methods for handing constraints in this optimization problem. One of the many methods is the penalty method where the optimization is converted into unconstrained optimization by including the constraints in the objective function. However, Lagaros, Papadrakakis and Kokossalakis [93] presented the so-called death penalty method. In this method, every infeasible design is discarded and it works well in EAs. In the current work, death penalty method is used. Also, there are different methods of implementing a GA. The details of the method implemented in this
The optimization starts with a set of parent vectors representing possible designs of the structure. These vectors are the initial population which are randomly generated in the design space. The objective function and constraints are evaluated for all these vectors. If any of these parent vectors do not satisfy the constraints (meaning that they are infeasible), it is replaced with a new vector until the entire set of parent vectors becomes feasible. This process is continued until there are $\mu$ feasible parent vectors. Later, re-combinations and mutation operations are used to form $\lambda$ offspring for each generation [85], which are again checked if they are checked for their feasibility. In the current work, all the parent vectors are recombined and mutated to form offspring in every generation ($\mu = \lambda$). The parent selection of the next generation is conducted based on $(\mu, \lambda)$ evolution strategies (ES) scheme [93, 94], where the best ($\mu/2$) parents of the previous generation and the best ($\lambda/2 = \mu/2$) feasible offspring of the current generation become the parent vectors for the next generation. The total number of parent vectors remains the same as $\mu$ in every generation. Figure 6.1 shows the flowchart of the optimization procedure. The steps for the optimization, similar to what Lagaros et al. [85] presented, can be written in detail as:

1. **Parent Initialization**: Initialization of $\mu$ parent vectors.

2. **Evaluation**: Evaluate objective function and constraints for parent vectors.

3. **Constraint Check**: Modify/replace infeasible parent vectors until all become feasible.

4. **Offspring generation**: Use recombination and mutation operations on parent vectors to form offspring vectors.

5. **Evaluation**: Evaluate objective function and constraints for offspring vectors.
6. **Constraint Check**: Continue if there are at least $\mu/2$ feasible offspring, else go to step 4.

7. **Parent Selection**: Best $(\mu/2)$ parents of previous generation and the best $(\lambda/2 = \mu/2)$ feasible offspring of current generation become the parent vectors for next generation.

8. **Convergence**: Stop if converged, else go to step 4.

In the current work, the optimization is stopped when the best objective function value is improved by less than 0.01% for $2*\xi$ generations. The optimization is implemented in Python using a Python-based library: Distributed Evolutionary Algorithms in Python (DEAP) [95]. It is an evolutionary computation framework for rapid prototyping and testing of different ideas.

![Flowchart of the constrained optimization procedure.](image-url)
6.3.2 Active Learning

Active learning, also called “query learning,” or “optimal experimental design”, is a sub-field of machine learning [96]. The main idea is that there may not be the need of large training datasets if the learning algorithm has the freedom to choose its training data. The idea comes from the motivation that sometimes there are huge unlabelled datasets available, and it is very expensive to label those datasets (meaning finding the required outputs of all the inputs). Here, the active learner is trained on a small labelled dataset. Later, the learner goes through the whole unlabelled dataset and it interactively queries the outputs of certain data points, whenever the model would be uncertain about those outputs. The learner only queries the output of those data-points when it can learn something new and improve its performance. The output of these queries can be provided by a human user and the learner can be re-trained for improving its performance. This helps in reducing the time and resources for labeling the data.

The active learner can query the required output of the unlabelled dataset with different strategies. In the current work, query by bagging [89] is used. It is based on the use of an active learner, which has a committee of many base estimators i.e., surrogate models. Each of the base estimators is trained on random subsets of the original training dataset, and a final prediction is made by taking aggregated votes from all the estimators. The base estimator in the committee can be different classifiers, like decision trees, ANNs, support vector machines (SVMs), k-nearest neighbours (kNNs), etc. If the task of the learner is to do binary classification, meaning predicting either 0 or 1, then certainty of a prediction can be computed as [97]:
6.3. Methodology

\[
p(0|x) = \frac{\text{Number of Estimators predicted class } 0 \text{ for input } x}{\text{Total Number of Estimators in Active Learner}} \quad (6.1)
\]

\[
p(1|x) = 1 - p(0|x) \quad (6.2)
\]

If the learner is not certain between the two classes, then the probabilities of predicting two classes would be close to 0.5 and the difference between those probabilities would be close to 0. Similarly, if the learner is confident about its output for an input, then the difference between the probabilities would be higher and close to 1.0. In this chapter, a confidence parameter, \( \gamma \), is defined. It denotes the difference between the probabilities of predicting two classes in a binary classification. It quantifies the confidence level of the learner in its prediction. It can be written as:

\[
\gamma = |p(0|x) - p(1|x)| = |1 - 2p(0|x)| \quad (6.3)
\]

The active learner issues a query for a prediction if \( \gamma \) is less than a threshold value. This threshold parameter is defined as \( \beta \) in the current chapter. The value of \( \beta \) is important for the performance of the active learner. The active learner only believes in its prediction if \( \gamma \) is greater than \( \beta \), otherwise it would issue a query. Therefore, a higher value of \( \beta \) could make the learner issue more queries. On the other hand, if \( \beta \) is set to a lower value, the performance of the active learner could get worse because the learner would not even issue a query even if it was uncertain about its prediction. Another important parameter of the active learner is its number of estimators, \( \phi \). A lower number of estimators in the learner could result in an inaccurate prediction even with a higher value of \( \gamma \). Therefore, the learner should have enough estimators for good performance. Different case studies are conducted.
in Section 6.4 to study the impact of $\beta$ and $\phi$ values on the performance of the optimization.

![Dataset of 600 Instances](image1)

![Random Instances from Dataset](image2)

Figure 6.2: (a) A toy data set of 600 instances available from scikit-learn [2]; (b) 50 labeled instances randomly picked from the dataset for initial training of active learner

An illustrative example is shown in this section to show the application of active learning. A toy dataset of 600 instances is used from the scikit-learn Python library [2]. The dataset has two classes along with some noise, meaning that there is no clear division between the region of the two classes as shown in Fig. 6.2(a). Initially, 50 labelled instances are randomly picked from the dataset for initial training of the active learner as shown in Fig. 6.2(b). Later, different types of active learners are studied to see their performance. The scikit-learn Python library [2] is used in the current chapter to train machine learning models.

Firstly, the active learner based on decision trees is studied. Figure 6.3 shows the performance of the active learner based on a committee of 20 different decision trees. The learner is trained on the initial training dataset of 50 instances. As stated earlier, each of the decision trees in the learner is trained on random subsets of the 50 instances. Figure 6.3(a) shows a contour plot of the confidence parameter ($\gamma$) over the whole domain. It shows the regions where the learner is least certain about its output. The learner is most uncertain in the first and fourth quadrant of the plot. At this stage, the accuracy of the learner over the whole dataset
is 68%. This learner is then provided with the remaining 550 data points of the dataset to issue queries. It is allowed to issue a query if $\gamma$ for any prediction is less than the threshold value of $\beta = 0.5$. Figure 6.3(b) shows 183 queries issued by the learner. As the learner is most uncertain in the first and fourth quadrant, it issued maximum queries in that region. Finally, these queries are added to the initial training dataset and the learner is retrained. The accuracy of the learner over the complete dataset increased to 93%. Figure 6.3(c) shows the improved confidence contour plot of the active learner after retraining. It should be noted that the retrained active learner can again be used to issue queries from the remaining data-points for more improvement. This case study is only conducted for one iteration of issuing queries to show the application of active learning. This study shows that it is possible to attain high accuracy without the need to train the machine learning model over the whole dataset.

Secondly, the active learner based on ANNs is studied. Figure 6.4 shows the performance of the active learner based on a committee of 20 different ANNs. The architecture of the ANNs consists of three hidden layers with six neurons in each. Each of the ANNs is ini-
tialized with different weights and biases, and is trained on the random subsets of the 50 instances. Figure 6.4(a) shows a contour plot of the confidence parameter ($\gamma$) of the learner over the whole domain after training on an initial training dataset of 50 instances (shown in Fig. 6.2(b)). The learner has the least confidence in the first and fourth quadrant of the plot. The accuracy of the learner at this stage is 85%. Figure 6.4(b) shows 144 queries issued by the learner from the remaining 550 data points. The accuracy of the learner increased to 91% after adding the new queries in the initial training dataset. Figure 6.4(c) shows the improved confidence contour plot of the active learner after retraining. This case study is also conducted again, but with different architecture of the ANNs in the committee. The architecture of the ANNs is changed to three neurons in the three hidden layers. Figure 6.5 shows the performance of the active learner based on a committee of 20 different ANNs with the new architecture. Here, the Fig. 6.5(a) shows that the learner is uncertain about almost the entire domain. Therefore, it issued 419 queries from the 550 data points, as shown in Fig. 6.5(b), which is much higher than in the previous case studies.

![Confidence Plot](image)

Figure 6.4: (a) Confidence plot after initial training of active learner made of ANNs (accuracy: 85%); (b) 144 queries issued by the learner; (c) confidence plot after including queries in the training dataset (accuracy: 91%)

This case study points out one of the main disadvantages of ANNs. ANNs require tuning
of the number of hyper-parameters such as the number of neurons, hidden layers, and the number of training epochs. Therefore, it is important to pick the right architecture of ANNs for getting good performance in active learning. Decision tree-based active learners have advantages over ANN-based learners in the current implementation for being more robust and needing less user involvement. The main motivation of using active learning, in the current research, is to help GA to save computational time and resources. It would be difficult to know the right architecture of the ANNs in the active learner before performing the optimization. Therefore, in the current work, decision tree-based active learners are studied in Section 6.4. It is seen that the approach shows great potential to save computational resources. More details are provided in Section 6.4.

![Confidence Plot](image1)

**Figure 6.5:** (a) Confidence plot after initial training of active learner made of ANNs (accuracy: 78%); (b) 419 queries issued by the learner; (c) confidence plot after including queries in the training dataset (accuracy: 89%)

### 6.3.3 Active Learning-based Genetic Algorithm (ALGA)

This section presents the details about the use of active learning in the GA. It has been shown in the Section 6.3.2 that an active learner can attain high accuracy over a domain without the need of a large dataset. This advantage of active learning is used in the GA
for improving its computational performance. During the first parent vector generation, the GA generates a small dataset of feasible and infeasible designs. The GA only considers the feasible parent vectors for offspring generation and discards all the infeasible designs, as shown in Section 6.3.1. However, this data can be used to initially train the active learner. Later, during the optimization, instead of actually evaluating constraint functions, the GA utilizes the active learner to predict if the possible design is feasible or not. If the active learner is uncertain about its prediction, meaning if $\gamma$ for the prediction is low, then the active learner issues a query and the actual constraint functions are evaluated. These queries are saved for later retraining of the active learner. Once certain number of queries are collected, the active learner is retrained for improving its predictions for future generations during the optimization. The steps for the optimization using ALGA can be written as:

1. **Parent Initialization**: Initialization of $\mu$ parent vectors.
2. **Evaluation**: Evaluate objective function and constraints for parent vectors.
3. **Constraint Check**: Modify/replace infeasible parent vectors until all become feasible.
4. **Active Learner Initial Training**: Use feasible and infeasible vectors dataset for initial training of active learning.
5. **Offspring generation**: Use recombination and mutation operations on parent vectors to form offspring vectors.
6. Initialize variable query-counter = 0.
7. **Offspring Evaluation**:
   
   For every vector in offspring generation:
   
   a. Evaluate objective function. Use active learner to predict the feasibility of the vector (meaning if it satisfies constraints or not).
b. If $\gamma$ for the vector $< \beta$:

- Issue a query and evaluate actual constraint function for the vector.
- Add the vector to training dataset.
- query-counter = query-counter + 1.
- If query-counter > $m$: retrain active learner and set query-counter = 0; else: continue.

else: continue.

8. **Constraint Check**: Continue if there are at least $\mu/2$ feasible offspring, else go to step 5.

9. **Parent Selection**: Best $(\mu/2)$ parents of previous generation and the best $(\lambda/2 = \mu/2)$ feasible offspring of current generation become the parent vectors for next generation.

10. **Convergence**: Stop if converged, else go to step 5.

There are some important parameters for the successful execution of the active learning-based GA. The parameters related to the learner are the number of estimators in the learner ($\phi$), threshold value ($\beta$) for the confidence parameter, and the total number of queries before the learner should be retrained ($m$). These parameters are studied in Section 6.4 for different number of population sizes ($\mu$) for three different optimization problems.

### 6.4 Application and Results

This section presents the application of the methodology presented in Section 6.3. Three different optimization case studies are conducted. Initially, two optimization problems are
considered from the literature to show that 1) the optimizer is able to find near-optimal solutions as mentioned in the literature and 2) active learning-based genetic algorithm (ALGA) (shown in Section 6.3.3) can help in reducing the required number of constraint evaluations, in comparison to the constrained genetic algorithm (GA) (shown in Section 6.3.1) and penalty method-based GA. Later, ALGA is used to optimize curvilinearly stiffened aircraft panels. The authors have previously conducted optimization of these panels [48], and have found that the problem is a non-convex optimization and takes significantly high CPU time because of the enormous number of FEA evaluations. ALGA is used to conduct the optimization of such panels to see if there could be any improvement in the computational
6.4. Application and Results

Table 6.1: Data for ten bar truss [3]

<table>
<thead>
<tr>
<th>Material</th>
<th>aluminium</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young’s modulus</td>
<td>$10^7$ psi</td>
</tr>
<tr>
<td>Minimum area</td>
<td>0.1 in$^2$</td>
</tr>
<tr>
<td>Specific mass</td>
<td>0.1 lbm/in$^2$</td>
</tr>
<tr>
<td>Allowable stress</td>
<td>$\pm 25,000$ psi</td>
</tr>
<tr>
<td>Allowable displacement</td>
<td>$\pm 2$ in</td>
</tr>
</tbody>
</table>

performance.

6.4.1 Case Study I: Ten-Bar Truss Problem

The ten bar truss model was developed by Venkayya [98]. Haftka and Gürdal [3] conducted an optimization of a ten-bar truss structure as shown in Fig. 6.7. The objective is to minimize the weight of the structure by designing the truss areas. This corresponds to ten design variables. The structure is designed under two loads applied at Node 2 and 4. The magnitude of the loads is 100 Kips. There are stress and displacement constraints. The stress in a truss should be within $\pm 25,000$ psi. The displacement of Node 1-4 in the vertical direction should be within $\pm 2$ in. The optimal weight found by Haftka and Gürdal [3] was 5060.85 lbs.

The ALGA approach proposed in this chapter is used to optimize the ten-bar truss structure. The motivation of conducting this optimization is to show that ALGA is able to find a near-optimal solution by saving some constraint evaluations as compared to the constrained GA and the penalty method-based GA.

It is important to choose the right parameters of the active learner during the optimization. As mentioned in Section 6.3.3, if the right parameters are not chosen, the active learner could make a mistake in its prediction of the feasibility of a design. The population size ($\mu$) of the GA is also important and is related to the complexity of the problem. If the optimization is non-convex and has multiple local minima, then a larger population helps
in finding the global minimum in the optimization. A parametric study is conducted using various values of $\mu$, $\beta$, $\phi$ and $m$ as shown in Fig. 6.8. As the optimization is stochastic, three independent runs of the optimization are conducted. Figure 6.8 shows, for each combination of parameters, how many optimization runs resulted in a feasible solution. The population size, $\mu$, is defined in terms of the number of design variables ($\xi$). The number of estimators, $\phi$, are chosen to be 25, 50, 100 and 150. The values of $\beta$ are chosen to be 0.2, 0.4, 0.6 and 0.8. It is seen that with the lower value of $\beta = 0.2$, the optimization always converged to infeasible designs. This is because of the reason that the active learner only issues queries if the $\gamma$ value of the prediction is less than 0.2. This resulted in the wrong feasibility predictions by the learner. Similarly, if $\phi$ is set to a lower value of 25, there is a chance that the active learner might make mistake, as shown in Fig. 6.8. However, if a higher number of estimators and a higher value of $\beta$ are chosen, then the converged solution is always feasible. Figure 6.8 also shows the analysis of $m$ on the feasibility of the converged solution. This parameter, $m$, is defined as a percentage of the population size $\mu$. In the current case study, $m$ does not have much impact on the feasibility of the converged solution. The higher value of $m$
essentially means that the active learner will wait longer before retraining itself. It should be preferred to choose a lower value of \( m \), so that the learner is updated more frequently about the optimizer search domain. The values of 0.6 and 0.8 are selected for \( \beta \), and the values of 50, 100 and 150 are selected for \( \phi \), for the next case studies in this section.

Figure 6.8: Analysis of different parameters of active learner on feasibility of optimal design in the ten-bar truss problem.

Figure 6.9 shows the comparison of the optimal solution found using different parameters in ALGA. The optimal solutions are compared with Haftka and Gürdal’s [3] solution. The EAs tend to find local optima if the population size is too small. Figure. 6.9 shows that all the combinations of parameters resulted in optimal solutions which were within 0.5% of Haftka’s solution, except for the case where \( \mu / \xi = 10 \), the error could be beyond 3%.

Figure 6.10 shows the comparison of percentage savings in the total constraint evaluations during the optimization. It shows that, for \( \beta = 0.8 \), the active learner issued more queries in comparison to the case where \( \beta = 0.6 \), and thus saved fewer total constraint evaluations. It
is seen that, approximately 55% of the total constraint evaluations are saved by the active learner if $\beta$ is chosen to be 0.6. Similarly, about 35% of the total constraint evaluations are saved by the learner when $\beta$ is 0.8. These results show that ALGA has great potential to save expensive FEA evaluations.

The optimal solutions found using ALGA are also compared with the optimal solutions found using constrained GA (Section 6.3.1) where no active learning is used, and also with the case where penalty method is used in the GA. This case study is conducted to show the benefit of using ALGA over other conventional approaches. Figure 6.11 shows the performance of ALGA with other methods. It shows that the optimal solution is very close to the Haftka [3] solution if a larger population size is chosen. Also it shows that approximately 60% fewer FEA evaluations are required if ALGA is used over the other two methods. This case study shows that ALGA can find the optimal solution while having improvement in the computational performance.
During the initial generations of the optimization, there is less information available for the active learner. However, as the optimization proceeds, more information about the design space becomes available to the active learner. This helps in increasing the confidence in the active learner’s results.
The total number of iterations of the optimization can be divided into five equal intervals for analyzing the value of employing the active learning for analysis. Figure 6.12 shows the % savings in the total evaluations in five intervals of the optimization iterations. It is seen that during the first 20% of the iterations, the active learner saves about 50% of the FEA evaluations. However, as the optimization proceeds, the % savings using active learning increases. During the last 20% of the iterations of the optimization, the active learner saves about 65% of the total evaluations. This case study shows that as the optimization proceeds, the active learner becomes more confident about the design space and can help the optimizer better in the later iterations of the optimization.

Figure 6.12: Comparison of % savings in total evaluations during the optimization of ten-bar structure
6.4.2 Case Study II: The Hesse Function

The second case study is conducted for the optimization of the Hesse function [82]. The function has 18 local minima in the design space. The function can be written as:

\[- f_{HE} = 25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2 + (x_6 - 4)^2 \quad (6.4)\]

The constraints of the optimization problems are written as:

\[
\begin{align*}
  x_{1,2} & \geq 0 \quad (6.5) \\
  1 & \leq x_{3,5} \leq 5 \quad (6.6) \\
  0 & \leq x_4 \leq 6 \quad (6.7) \\
  0 & \leq x_6 \leq 10 \quad (6.8) \\
  2 & \leq x_1 + x_2 \leq 6 \quad (6.9) \\
  x_1 - 3x_2 & \leq 2 \quad (6.10) \\
  4 & \leq (x_3 - 3)^2 + x_4 \quad (6.11) \\
  4 & \leq (x_5 - 3)^2 + x_6 \quad (6.12)
\end{align*}
\]

The optimization of the function is conducted with the motivation to see if ALGA can find the optimal solution with fewer constraint evaluations in comparison to both the constrained GA and a penalty method-based GA. The analytical solution to this optimization is -310. As was done in Section 6.4.1, the effect of the different parameters of the active learner on the feasibility of the optimization are studied. Figure 6.13 shows the analysis of the different combinations of $\mu$, $\beta$, $\phi$ and $m$ using three different independent runs of the optimization.
It also shows that the converged solution is feasible for higher values of $\phi$ and $\beta$. Again, two values, 0.6 and 0.8 are selected for $\beta$, and that of 50, 100 and 150 for $\phi$.

Figure 6.13: Analysis of different parameters of active learner on feasibility of optimal solution in Hesse function.

Figure 6.14 shows the comparison of the optimal solution with respect to the analytical solution using different combinations of the parameters of the active learner. It is seen that the optimal solution found in the optimization is close to the analytical solution. The maximum error is seen to be greater than 1% for $\mu/\xi$ equal to 20 or 30.

Figure 6.15 shows the comparison of the percentage savings in the total constraint evaluations during the optimization. It shows that, with the value of $\beta = 0.8$, ALGA saved about 65% of the total constraint evaluations. For the value of $\beta = 0.6$, it saved about 50% of the total constraint evaluations. This case study also shows that active learning has great potential to save expensive constraint evaluations.

The optimal solutions found using ALGA are again compared with the optimal solutions found using constrained GA (Section 6.3.1) where no active learning is used, and with penalty
method-based GA. Figure 6.16 shows the performance of ALGA in comparison to the other methods. It shows that the optimal solution is very close to the analytical solution if a larger population size is chosen and that significantly fewer constraint evaluations are required if ALGA is used over other two methods. This case study also shows that ALGA can find optimal solution while accelerating the optimization.

As shown in Section 6.4.1, the total number of iterations of the optimization can be divided into five equal intervals for analyzing the use of active learning. Figure 6.17 shows the % savings in the total constraint evaluations in the five intervals of the optimization iterations. It is seen that during the first 20% of the iterations, the active learner saves about 65% of the constraint evaluations. Moreover, as the optimization moves forward, the % savings using active learning also increases. During the last 20% of the iterations of the optimization, the
active learner saves about 85% of the total constraint evaluations. This case study also shows that the active learner becomes more confident about the design space as the optimization proceeds.
6.4.3 Case Study III: Curvilinearly Stiffened Panels

Case studies I and II showed that active learning has the potential to accelerate the optimization of complex problems with several local minima. In case study III, curvilinearly stiffened panels are optimized.

Advanced manufacturing techniques like additive manufacturing have made it possible for structural designers to make use of curvilinear stiffeners for achieving better designs of aircraft panels. However, optimization of such panels using expensive FEA evaluations makes that optimization very inefficient. The approach presented, ALGA, in this chapter is used to optimize these panels. Figure 6.18 shows the parameterization of curvilinear stiffener placement on a panel and applied loads. Singh et al. [48] defined two parameters: perimeter parameter, $\epsilon$, and curvature parameter, $\alpha$, for defining the placement and curvature of a stiffener on a panel, as shown in the Fig. 6.18. NURBS (Non-Uniform Rational B-Splines) is used to define the stiffener placement. The two control points ($\epsilon_A$ and $\epsilon_B$) are based on
the perimeter parameter and $\alpha$ is based on curvature parameter. For more details, [48] can be referred.

In this case study, there are nine design variables including six design variables for placement of two stiffeners, and three design variables for the stiffener cross-section and panel thickness. The objective of the optimization is to minimize the mass of the panel. There are two constraints based on the buckling factor and von Mises stress in a panel as shown in Eq. 6.13. The panel is subjected to the load case as mentioned in Table 6.2.

\[
\text{minimize } F(X) = \text{Mass} \\
\text{subject to } c_h \leq X_h \leq d_h, \quad h = 1, 2, ..., \xi \\
: \text{Buckling Factor} \geq 1 \\
: \text{von Mises stress } (\sigma) \leq \text{yield stress } (\sigma_y) \quad (6.13)
\]

<table>
<thead>
<tr>
<th>$N_x$ (kN/m)</th>
<th>$N_y$ (kN/m)</th>
<th>$a$ (m)</th>
<th>$b$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>454.50</td>
<td>513.24</td>
<td>0.5080</td>
<td>0.4064</td>
</tr>
</tbody>
</table>

The optimization is conducted using a population size of 300 using an active learner based on $\beta = 0.6$. ALGA is used with two combinations of $m$ and $\phi$: 1) $m/\mu = 0.2$; $\phi = 100$; 2) $m/\mu = 0.8$; $\phi = 50$. The optimization results are compared with constrained GA and penalty method-based GA using the same population size. Figure 6.19 shows the comparison of the masses of the optimal designs using different approaches. It can be seen that ALGA is able to find near-optimal designs. Figure 6.20 shows the percentage savings in total FEA evaluations.
in comparison to the penalty method and constrained GA. It can be seen that ALGA can save approximately 60% of FEA evaluations in comparison to the penalty method-based GA, and about 70% FEA evaluations in comparison to the constrained GA that does not employ active learning.

Figure 6.19: Comparison of optimal weights in optimization in curvilinearly stiffened panel.
Figure 6.20: Comparison of % savings in the total evaluations in optimization in curvilinearly stiffened panel.

Figure 6.21 shows the percentage savings in FEA evaluations during the optimization. The total optimization iterations are divided into five intervals. The active learner saves about 30% of the function evaluations during the first 20% of the optimization iterations. However, as the optimization proceeds, the active learner gains more information about the design space. The learner saved about 70% of the FEA evaluations in the last 20% of the optimization iterations.

Figure 6.21: Comparison of % savings in the total evaluations in optimization in curvilinearly stiffened panel.
Figures 6.22, 6.23, 6.24 shows the optimal designs found using ALGA, constrained GA and penalty method-based GA. The optimal masses are 3.21 kg, 3.23 kg and 3.28 kg respectively. The buckling factors of all the optimal designs are 1.00. The buckling constraint is active in all the optimal designs. However, the stress constraint is not active. The maximum von Mises stresses are 91.3 MPa, 89.4 MPa and 86.5 MPa for the optimal designs found using ALGA, constrained GA and the penalty method, respectively. These optimal designs show that ALGA can be successfully used to conduct evolutionary based optimization if a higher number of estimators are used along with a higher value of $\beta$.

![Optimal design using ALGA](image)

Figure 6.22: Optimal design using ALGA. Left: Geometry, Middle: 1st Buckling mode, Right: von Mises stress. (Mass = 3.21 kg)

### 6.5 Conclusions

This chapter presents a new approach for accelerating genetic algorithms. The approach is based on an adaptive machine learning technique, called active learning. The active learner helps in accelerating the GA by predicting whether a design is going to be feasible or infeasible. The approach is used to optimize: the well-known ten-bar truss problem, the Hesse function, and curvilinearly stiffened panels under buckling and stress constraints.
Figure 6.23: Optimal design using constrained GA. Left: Geometry, Middle: 1st Buckling mode, Right: von Mises stress. (Mass = 3.23 kg)

Figure 6.24: Optimal design using penalty method based GA. Left: Geometry, Middle: 1st Buckling mode, Right: von Mises stress. (Mass = 3.28 kg)

The results show that the approach has the potential to reduce between 50% to 70%, the number of total required constraint evaluations. It is, however, important to choose the right parameters for the active learner for its use in successful execution of the optimization. The value of the $\beta$ for optimization is recommended to be greater than 0.6. The decision tree-based active learner are more robust and need less user-involvement as compared to the neural network-based active learner. The number of estimators in the active learner are recommended to be greater than 100 for correct predictions made by the active learner.
In future, parallel processing could be incorporated in ALGA to improve computational performance. The active learner based on different combinations of estimators could also be evaluated, like, decision trees, ANNs, support vector machines (SVMs), k-nearest neighbours (kNNs), etc.
Chapter 7

Machine Learning Approaches for Finite Element Modeling Recommendations

7.1 Abstract

The finite element method has been well-established for modern structural design and analysis. However, significant modeling choices must be made to achieve valid answers sufficient for engineering decisions, but without employing high computational effort. Many situations may arise where a reduced modeling may be insufficient and a full 3-D modeling approach is required to get valid results. Decisions about where to apply each type of modeling approach are typically based upon user experience and are only periodically validated using experiments. In this chapter, subjective nature of such decision making is removed via comprehensive multi-level qualification scheme where high-fidelity models are used to judge the sufficiency of low and mid-fidelity models. These modeling assessment results are used to train machine-learning software to make recommendations for creating valid models for arbitrary geometric and load parameters. Artificial Neural networks for automated mid-fidelity modeling recommendations have been created for T-joint and stepped-plate problems under different loads.
7.2 Introduction

Numerical techniques like Finite Element Method (FEM) and the rapid increase in computational resources have made it possible to simulate the physical behavior of ever most complex structural components [99]. The FEM has been well-established for structural design and analysis with comprehensive commercial packages such as MSC.NASTRAN, ABAQUS, ANSYS, etc. One of the important concerns in using FEM is that the modeling techniques, including the choice of element type, size, and boundary conditions, could become critical for evaluating stress, strain and displacements accurately while designing a structure. Model efficiency and turn-around time can become critical for practical design optimization where many candidate structures will be evaluated. Of course, such models must retain sufficient accuracy for making engineering decisions. The present work seeks to develop an approach for creating modeling recommendations to provide fast models which are accurate enough for their intended purpose.

While ultimately modeling validation involves comparisons with experimental data, a multi-fidelity modeling approach can be used to develop modeling recommendations that can be used before such test data is available and to fill-in modeling assessments/recommendations for regions not fully covered by an executed testing program. In its simplest form, a fully converged high-fidelity model which is robust, but not necessarily efficient, provides results which can be employed to access the efficacy of other simpler models. Extensive research has been conducted on this topic in the literature with the motivation to find best modeling choices for modeling complex structures. Romeed et al. [100] conducted comparison of 2-D and 3-D finite element analysis of a restored tooth. They concluded that 2-D finite element analysis may help in investigating key aspects of the mechanical behaviour of a dental restoration in a single tooth unit, but combinations of 2-D and 3-D FEA may offer the best understanding of the bio-mechanical behaviour of complex dental structures. Krueger
and Minguet [101] studied analysis of composite skin–stiffener debond specimens using a 2-D/3-D modeling technique. They stated that the computed total strain energy release rates obtained from 2-D/3-D simulations were in good agreement with results obtained from full 3-D models. Krueger and Brien [102] used 2-D/3-D modeling technique for the analysis of delaminated composite laminates. They used multi-point constraints to provide a kinematically compatible interface between the local 3-D model and the global structural model meshed with 2-D finite elements. They showed that the combination of 2-D/3-D modeling technique has an advantage for reducing the model size, because only a relatively small section in the structure needs to be modeled with 3-D elements.

Typically, the decisions about where to apply which type of modeling approach in various regions are based upon user experience and judgment and are only periodically validated against experimental data. The engineering effort for execution of these analyses and calculations results in a significant portion of resources spent by different industries. Variations in approaches can also lead to costly delays. Glickman and Romero [103] presented a case study on analyst-to-analyst variability in simulation-based prediction by interviewing multiple analysts. Their observations demonstrated a linkage between analysts’ background and experience and their particular choices of methods. This could result in different choices being made to model/simulate a particular problem. Oberkampf, Pilch, and Trucano [104] proposed a new model, called The Predictive Capability Maturity Model (PCMM), that can be used to assess the level of maturity of computational modeling and simulation (M&S) efforts. Its development is based on different analysis of similar investigations. The purpose of the PCMM is to improve the decision making for some engineering system applications.

The six M&S elements used to assess maturity in this model are (1) representation and geometric fidelity, (2) physics and material model fidelity, (3) code verification, (4) solution verification, (5) model validation, and (6) uncertainty quantification and sensitivity analysis.
In the present work, the subjective nature of such decision making in building a model is removed via a comprehensive multi-level qualification scheme where high-fidelity models provide the benchmark results to assess low and mid-fidelity models for a broad range of geometric and loading parameters. A design tool is developed with the motivation to provide an automated design agent to provide the specific modeling and fidelity requirements for particular parameter choices in a given problem. This approach can not only save computational effort in day-to-day modeling of common geometries, but it also can provide a standard modeling recommendation that can help eliminate concerns and delays associated with analyst-to-analyst variability.

Herein, machine learning will be used to develop automated modeling recommendations. In the past decades, the field of machine learning has seen significant development. For instance, artificial neural networks (ANNs) and decision trees (DTs) have shown great potential as computationally efficient surrogate models. Fahmy et al. [105] used ANNs for conceptual design of orthotropic steel-deck bridge. Their motivation is that the professional software tools do not aid the designer in choosing a preliminary economic layout at the conceptual design stage. The designers have to go through iterative, lengthy and expensive procedures to reach the best configuration and, thus, they used a machine learning based approach to reduce the computational and time resources required. Tong and Liu [106] presented the use of ANN in the preliminary structural design of shells. Rogers [53] used ANNs for simulating structural analysis. It is stated that a fast, inexpensive neural network can be used as an alternative to a slow, expensive structural analysis program. In the current work, the benefits of machine learning are leveraged. The results of the modeling assessments of different geometries are used to train machine learning software to make automatic modeling recommendations for creating accurate and efficient models for arbitrary geometric and load parameters. This approach provides accurate modeling recommendations to a user within
seconds, thus saving significant amount of time spent in iterating the analysis choices.

In Section 7.3, the methodology for creating and validating multi-fidelity models of two geometries of interest: T-joint and stepped plate is presented. The machine learning approaches along with the different modeling recommendations schemes are presented. In Section 7.4, the methodology is applied on two case studies where ANNs are used to provide modeling recommendations. It is observed that this approach has a great potential to save significant computational resources for accurately providing modeling recommendations.

7.3 Methodology

In this work, two geometries of interest, a T-joint and stepped plate structures, are studied. The motivation behind selection of these geometries is that one commonly come across these two geometries while designing different real-world structures like aircraft, submarines, buildings, etc. One particular area of concern while designing these geometries is the selection of modeling topology. For regions relatively far away from the joint of vertical and horizontal members in a structure, shell elements can be safely used. However, near those transition regions of the joint, using standard shell elements leads to significant errors as important through-the-thickness 3-D effects would not be captured. This necessitates the use of solid continuum elements in those regions. However, modeling such regions or entire structures using solid continuum elements requires multiple layers of elements through the thickness to capture the bending response. Maintaining proper element aspect ratios can drive the element count very high. Hence, utilizing a blend of both approaches is ideal for modeling efficiency and accuracy. A multi-fidelity modeling approach is presented for both geometries which provides a rational way to create the desired modeling recommendations by varying the regions of each type of modeling and comparing their results against those from a full
7.3. Methodology

3-D continuum element model.

7.3.1 Validation Scheme for Multi-fidelity Models

The ASME V&V guide [107] defines verification and validation as follows:

**Verification:** the process of determining that a computational model accurately represents the underlying mathematical model and its solution.

**Validation:** the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.

The real-world assessment of computational model results for validation requires comparison with physical test data. Such physical test data is not always readily available. Here, the concept of validation is augmented to include qualification assessments against results from computational models with higher physical fidelity. Such higher physical fidelity may include additional material physics, more detailed geometric representation, inclusion of large strains and/or large rotations, better contact representation, more fundamental element topologies (e.g., 3-D continuum vs. 2-D shell), etc. Such higher fidelity correspond to improvements not only in the computational model per se, but the underlying mathematical model as well. For instance, a linear elastic beam model may be partially validated by comparing its predictions against those from a fully 3-D continuum model of the same geometry using an elastic-plastic material model including geometric non-linearity.

Here the term hierarchical model qualification is used to denote the concept of evaluating a model’s sufficiency using a family of mathematical models of increasing physical fidelity. Such higher fidelity mathematical models themselves will lead to higher fidelity computational models which themselves are subject to verification and validation with ultimate validation via comparison with experimental data. Hence, ultimately all models must be subjected to
classical validation as shown in Fig. 7.1, but in the absence of available physical test data, hierarchical modeling qualification can be used to provide significant confidence in modeling results.

Figure 7.1: Combined hierarchical qualification/classical validation approach.

In the current work, five different quantities are used as metrics to validate multi-fidelity models of different geometries with respect to their respective full 3-D models. These five quantities are maximum displacement, total strain energy, maximum principal stress, maximum principal strain, and maximum von Mises stress. These quantities are chosen because they are typically important while conducting structural design and optimization.
Table 7.1: Dimensions of a T-joint.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Value (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>10.00</td>
</tr>
<tr>
<td>$L$</td>
<td>5.00</td>
</tr>
<tr>
<td>$W$</td>
<td>5.00</td>
</tr>
<tr>
<td>$t$</td>
<td>0.20</td>
</tr>
<tr>
<td>$r$</td>
<td>0.20</td>
</tr>
</tbody>
</table>

7.3.2 Example Structures

In this chapter, two geometries, T-joint and stepped plate structures, under bending and shear loads are selected for study. This section presents the parameterization of these structures and their analysis results.

T-Joint

A T-joint structure is analyzed at the extremes of using all 2-D or all 3-D elements in the mesh. The results from static analysis of the structure are presented. Figure 7.2 shows an example full 3-D T-joint structure with geometric parameters and applied traction. A fillet of radius $r$ is considered in the structure. The dimensions of the structure are given in Table 7.1. In order to avoid any significant effect of boundary conditions on the five quantities of interest, the results are only compared in the region away from the edges as shown in blue color in Fig. 7.2. The thicknesses of the vertical and horizontal members are considered to be 0.2 m. A vertical total load of 5 N ($F_y$) along with total shear load of 2.5 N ($F_z$) are applied as the surface tractions on the horizontal edge surface of the full 3-D model. In the full 2-D structural model, a vertical load of 1 N/m ($F_y$) along with 0.5 N/m ($F_z$) are applied as a distributed load on the edge of the full-2D model. It should be noted that the total applied load in each case is equivalent.

Initially, for comparison, the structure is analyzed using 3-D continuum elements (Abaqus/S-
Figure 7.2: Example T-joint structure under applied traction with key geometric sizing parameters indicated.

standard C3D8R element which utilizes reduced integration with hourglass stiffness) to model the entire domain. In order to select a correct element size, a mesh convergence study using total strain energy is presented in Fig. 7.3. In the mesh, elements having 3:3:1 aspect ratio are used. The element length along the thickness direction is varied to give 1 to 5 elements corresponding to 616 and 29,946 total elements, respectively. For the results presented in this section, 4 elements through the thickness (0.05 m long in thickness direction) and 0.15 m of edge length is recommended to the Abaqus/CAE mesher to use globally outside this dimension. At the other extreme, a limiting 2-D model is created by using 2-D shell elements everywhere (Abaqus/Standard S4R element which employs reduced integration with hourglass stiffness). The global element edge length is again specified as 0.15 m. The results are compared with the pure solid model with and without fillet as detailed in Table 7.2. Recall, the results are only compared in the regions away from the edge boundary conditions.
Clearly the results show that the analysis model using only 2-D elements has significant error compared to the models using only 3-D elements. It can be observed that a full 2-D model gives significant differences in the stresses and strains. Hence, the opportunity to use a multi-fidelity model to achieve accurate results with improved efficiency is indicated. Of course, such a multi-fidelity model will require determining an optimal distribution of the 2-D and 3-D regions.

The Figs 7.4, 7.5, 7.6, 7.7 and 7.8 compares the five quantities of interest in full 3-D and full 2-D model. It shows that there could be significant differences in these quantities.

![Figure 7.3: Mesh convergence study for a full 3-D model of a T-joint.](image)

<table>
<thead>
<tr>
<th>Model</th>
<th>Full 3-D (with fillet)</th>
<th>Full 3-D (without fillet)</th>
<th>Full 2-D (without fillet)</th>
<th>% Abs. Diff Full 2-D &amp; Full 3-D (w/ fillet)</th>
<th>% Abs. Diff Full 2-D &amp; Full 3-D (w/o fillet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Strain Energy (J)</td>
<td>2.09E-06</td>
<td>2.24E-06</td>
<td>2.19E-06</td>
<td>4.73</td>
<td>2.32</td>
</tr>
<tr>
<td>Max. Disp. (m)</td>
<td>8.78E-07</td>
<td>9.45E-07</td>
<td>9.19E-07</td>
<td>4.62</td>
<td>2.73</td>
</tr>
<tr>
<td>Max. Abs. Principal Stress (N/m²)</td>
<td>592.313</td>
<td>634.942</td>
<td>795.087</td>
<td>34.23</td>
<td>25.22</td>
</tr>
<tr>
<td>Max. Abs. Principal Strain</td>
<td>7.27E-09</td>
<td>7.44E-09</td>
<td>9.71E-09</td>
<td>33.53</td>
<td>30.48</td>
</tr>
<tr>
<td>Max. von Mises Stress (N/m²)</td>
<td>527.639</td>
<td>539.712</td>
<td>701.897</td>
<td>33.03</td>
<td>30.05</td>
</tr>
</tbody>
</table>
Different computational experiments are conducted to find the reason for differences in the results in full 3-D and full 2-D. A cantilever beam is considered under unit tip moment. As shown in Fig. 7.9, four different cantilever beam models with number of elements through the thickness from 3 to 6 are considered.
The shell element S4R in ABAQUS uses reduced integration with a single integration point at the element centroid. It employs 5 section points thru-the-thickness to integrate in that direction. Hence, the S4R elements naturally have accurate stresses at the top and bottom surfaces for bending. On the other hand, the solid element C3D8R truly has a single integra-
Figure 7.8: Contour plot of von Mises stress in T-Joint: (Left) Full 3-D, (Right) Full 2-D.

Figure 7.9: Different models of cantilever beam with different number of through-thickness elements.

...tion point at the element centroid. Using a stack of C3D8R elements through the thickness will not give an integration point stress locations at the top and bottom surfaces for bending. Hence, they do not get the full moment arm in determining the section bending moment.
Figure 7.10 shows the convergence study of using different number of C3D8R (Reduced Integration) and C3D8 (Full Integration) elements thru-the-thickness of the cantilever beam. It can be seen that we can expect about 20% error for the peak bending stress (note that the section bending moment is constant along the length in this case) using four C3D8R solid elements thru-the-thickness compared to the exact solution.

![Graph showing error in peak stress with varying number of elements](image)

**Figure 7.10**: Peak bending stress convergence with varying number of thru-the-thickness elements (Elm denotes element integration point values, while Nodal refers to interpolated nodal values).

**Nominal Multi-fidelity Model of T-joint**: Both 2-D shell and 3-D continuum elements are employed in the nominal multi-fidelity model. Figure 7.11 shows the model created using both the 3-D and 2-D elements where the 3-D region covers the actual welded joint. Two parameters, $\alpha = D_1/H$ and $\beta = D_2/L$, are used to define the 3-D region as shown in Fig. 7.12. The shell thickness of the 2-D region is added to show that the thickness is kept uniform in the structure. The shell-to-solid coupling feature of Abaqus/Standard is used to
join the 3-D and the 2-D element regions. The model is generated with $\alpha = \beta = 0.4$ as an example to show benefits of multi-fidelity model. The model is analyzed under the same load and dimensions as shown in Fig. 7.2 with the results listed in Table 7.3. The mixed fidelity model has only 8,118 elements in comparison to 14,124 elements in the pure 3-D model. In the present work, a framework is implemented for finding the optimum distribution of 2-D and 3-D regions in the multi-fidelity model such that the required results (total strain energy, max von Mises stress, etc.) remain close with the results produced using model with all 3-D elements.

Figure 7.11: T-Joint modeling: (a) multi-fidelity model with 2-D and 3-D element topologies and (b) results region of interest with grey color showing edge regions near applied traction and kinematic boundary conditions that are not considered in the results comparisons.

**Importance of Sufficient 3-D Region for T-Joint**  Results from multi-fidelity models with different distribution of 3-D and 2-D regions are now presented. Table 7.5 shows the comparison of results of multi-fidelity T-Joint model with a smaller 3-D region. It is determined that the multi-fidelity results with a small 3-D region ($\alpha = \beta = 0.1$) near the joint still has significant errors compared to full 3-D results. However, the multi-fidelity results from using a large enough 3-D region ($\alpha = \beta = 0.4$) near the joint lead to very good
Figure 7.12: Parameterization of the multi-fidelity T-Joint model with 2-D and 3-D regions indicated in orange and blue, respectively.

Table 7.3: Comparison of multi-fidelity model and high-fidelity (full 3-D) results for a T-joint ($\alpha = \beta = 0.4$).

<table>
<thead>
<tr>
<th>Model</th>
<th>Full 3-D</th>
<th>Multi-fidelity</th>
<th>% Abs. Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Strain Energy (J)</td>
<td>2.09E-06</td>
<td>2.07E-06</td>
<td>0.78</td>
</tr>
<tr>
<td>Max. Disp. (m)</td>
<td>8.78E-07</td>
<td>8.68E-07</td>
<td>1.11</td>
</tr>
<tr>
<td>Max. Abs. Principal Stress (N/m²)</td>
<td>592.313</td>
<td>591.842</td>
<td>0.08</td>
</tr>
<tr>
<td>Max. Abs. Principal Strain</td>
<td>7.27E-09</td>
<td>7.27E-09</td>
<td>0.08</td>
</tr>
<tr>
<td>Max. von Mises Stress (N/m²)</td>
<td>527.639</td>
<td>527.239</td>
<td>0.08</td>
</tr>
</tbody>
</table>

accuracy as compared to the full 3-D results as shown in Table 7.3. This demonstrates that there is a need for sufficient 3-D region near the joint of T-joint structure to produce results within the user acceptable error limits for engineering judgments.
Table 7.4: Comparison of computational cost of the models shown in Table 7.3.

<table>
<thead>
<tr>
<th>Model</th>
<th>Full 3-D</th>
<th>Multi-fidelity</th>
<th>% Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Elements</td>
<td>14,124</td>
<td>8,220</td>
<td>41.80</td>
</tr>
<tr>
<td>Number of DOF</td>
<td>55,284</td>
<td>37,536</td>
<td>32.10</td>
</tr>
</tbody>
</table>

Figure 7.13: Contour plots with respect to $\alpha$ and $\beta$ for %error in a) Total Strain Energy b) Max. Disp. c) Max. Abs. Principal Stress d) Max. Principal Strain e) Max. von Mises Stress in T-joint problem.

Stepped Plate

A stepped plate structure is analyzed statically using all 2-D shell and all 3-D continuum element representations. Figure 7.14 shows an example of full 3-D stepped plate structure with geometric parameters and applied traction. A vertical total load of 4.878 N ($F_y$) and total shear load of 3.506 N ($F_z$) are applied as surface traction on the edge surface of the
Table 7.5: Comparison of multi-fidelity model and high-fidelity (full 3-D) results for a T-joint ($\alpha = \beta = 0.1$).

<table>
<thead>
<tr>
<th>Model</th>
<th>Full 3-D</th>
<th>Multi-fidelity</th>
<th>% Abs. Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Strain Energy (J)</td>
<td>2.09E-06</td>
<td>2.01E-06</td>
<td>3.85</td>
</tr>
<tr>
<td>Max. Disp. (m)</td>
<td>8.78E-07</td>
<td>8.42E-07</td>
<td>4.15</td>
</tr>
<tr>
<td>Max. Abs. Principal Stress (N/m²)</td>
<td>592.313</td>
<td>669.771</td>
<td>13.08</td>
</tr>
<tr>
<td>Max. Principal Strain</td>
<td>7.27E-09</td>
<td>8.20E-09</td>
<td>12.81</td>
</tr>
<tr>
<td>Max. Abs. von Mises Stress (N/m²)</td>
<td>527.639</td>
<td>592.29</td>
<td>12.25</td>
</tr>
</tbody>
</table>

Initially, for comparison, the structure is analyzed using 3-D elements (C3D8R) to model the entire domain. A mesh convergence study using total strain energy is presented in Fig. 7.15. As stated in the previous section, the element size/length along the thickness is varied from 2 element through thickness (4,161 elements) to 5 elements through thickness (41,136 elements). For the results presented in this section, 4 elements through thickness are used. The full 3-D model results are compared with the corresponding full 2-D (S4R) model results, as summarized in Table 7.7. It can be observed that the full 2-D model results have significant differences as compared to the full 3-D model. Hence, similar to the T-joint, there is an opportunity to use a multi-fidelity model to achieve accurate results here.

The Figs 7.16, 7.17, 7.18, 7.19 and 7.20 compares the five quantities of interest in full 3-D and full 2-D model. It shows that there could be significant differences in these quantities.
Table 7.6: Dimensions of a stepped plate.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Value (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>10.000</td>
</tr>
<tr>
<td>$L_2$</td>
<td>7.175</td>
</tr>
<tr>
<td>$t_2$</td>
<td>0.205</td>
</tr>
<tr>
<td>$t_1$</td>
<td>0.5337175</td>
</tr>
<tr>
<td>$W$</td>
<td>5.825</td>
</tr>
<tr>
<td>$r$</td>
<td>0.100</td>
</tr>
</tbody>
</table>

Figure 7.14: Stepped plate example structure with geometric parameters and applied traction.

Table 7.7: Comparison of 2-D and 3-D results for a stepped plate structure.

<table>
<thead>
<tr>
<th>Model</th>
<th>Full 3-D</th>
<th>Full 2-D</th>
<th>% Abs. Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Strain Energy (J)</td>
<td>8.376E-06</td>
<td>7.83451E-06</td>
<td>6.46</td>
</tr>
<tr>
<td>Max. Disp. (m)</td>
<td>3.056E-06</td>
<td>2.87036E-06</td>
<td>6.07</td>
</tr>
<tr>
<td>Max. Abs. Principal Stress (N/m²)</td>
<td>982.324</td>
<td>1060.72</td>
<td>7.98</td>
</tr>
<tr>
<td>Max. Abs. Principal Strain</td>
<td>11.209E-09</td>
<td>13.2877E-09</td>
<td>18.54</td>
</tr>
<tr>
<td>Max. von Mises Stress (N/m²)</td>
<td>770.819</td>
<td>953.46</td>
<td>23.69</td>
</tr>
</tbody>
</table>
Figure 7.15: Mesh convergence study for full 3-D stepped plate model.

Figure 7.16: Contour plot of strain energy density in a stepped plate: (Top) Full 3-D, (Bottom) Full 2-D.
Figure 7.17: Contour plot of displacement in a stepped plate: (Top) Full 3-D, (Bottom) Full 2-D.

Figure 7.18: Contour plot of maximum principal stress in a stepped plate: (Top) Full 3-D, (Bottom) Full 2-D.
Figure 7.19: Contour plot of maximum principal strain in a stepped plate: (Top) Full 3-D, (Bottom) Full 2-D.

Figure 7.20: Contour plot of von Mises stress in a stepped plate: (Top) Full 3-D, (Bottom) Full 2-D.
Nominal Multi-fidelity Model of Stepped Plate: Two parameters ($\alpha = D_1/L_1$ and $\beta = D_2/L_2$) are defined to define the 3-D and 2-D regions in the structure as shown in Fig. 7.21 and 7.22. A fillet is also added at the step change in the 3-D region. The shell thickness of the 2-D region is added to show that thickness is kept uniform away from the step in the structure. The shell-to-solid coupling feature of Abaqus/Standard is used to join the transition between 3-D and the 2-D elements. The model is generated with $\alpha = \beta = 0.3$ as an example to show advantages of multi-fidelity model. The model is analyzed under the same load and dimensions as shown in Fig. 7.14 with the results tabulated in Table 7.8 and 7.9. The model has only 11,590 elements in comparison to 22,192 elements in the pure 3-D model, while having all errors for the quantities of interest less than 5%. This shows that optimal distribution of the 3-D and 2-D region can be used to save the computational cost without significant loss of accuracy while analyzing such structures.

![Figure 7.21: Stepped plate modeling: (a) multi-fidelity model with 2-D and 3-D element topology and (b) results region of interest with grey areas near kinematic and traction boundary conditions not considered.](image)

Importance of Sufficient 3-D region for Stepped Plate: Results from multi-fidelity models of stepped plate with different distribution of 3-D and 2-D regions are now examined. Table 7.10 shows the comparison of multi-fidelity stepped plate model having smaller 3-D region with respect to the full 3-D model. It is apparent that the multi-fidelity results with
Figure 7.22: Stepped plate parameterization of multi-fidelity model with 2-D and 3-D regions indicated in orange and blue, respectively.

Table 7.8: Comparison of multi-fidelity model and high-fidelity (full 3-D) results for a stepped plate ($\alpha = \beta = 0.3$)

<table>
<thead>
<tr>
<th>Model</th>
<th>Full 3-D</th>
<th>Multi-Fidelity</th>
<th>% Abs. Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Strain Energy (J)</td>
<td>8.376E-06</td>
<td>8.184E-06</td>
<td>2.28</td>
</tr>
<tr>
<td>Max. Disp. (m)</td>
<td>3.056E-06</td>
<td>2.983E-06</td>
<td>2.37</td>
</tr>
<tr>
<td>Max. Abs. Principal Stress (N/m$^2$)</td>
<td>982.324</td>
<td>981.584</td>
<td>0.07</td>
</tr>
<tr>
<td>Max. Abs. Principal Strain</td>
<td>11.209E-09</td>
<td>11.179E-09</td>
<td>0.26</td>
</tr>
<tr>
<td>Max. von Mises Stress (N/m$^2$)</td>
<td>770.819</td>
<td>768.541</td>
<td>0.29</td>
</tr>
</tbody>
</table>

Table 7.9: Comparison of computational cost of the models shown in Table 7.8

<table>
<thead>
<tr>
<th>Model</th>
<th>Full 3-D</th>
<th>Multi-Fidelity</th>
<th>% Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Elements</td>
<td>22,192</td>
<td>11,668</td>
<td>47.42</td>
</tr>
<tr>
<td>Number of DOF</td>
<td>83,655</td>
<td>51,675</td>
<td>38.22</td>
</tr>
</tbody>
</table>

small 3-D region ($\alpha = \beta = 0.1$) near the joint have significant errors in comparison to full 3-D model. However, as it was observed for the T-joint structure, the multi-fidelity results with large enough 3-D region ($\alpha = \beta = 0.3$) show good results compared to results from
the full 3-D model as detailed in Table 7.8. As demonstrated previously for the T-joint structures, this study shows that there is a need for sufficient 3-D region near the joint of stepped plate structure to produce results of sufficient engineering accuracy as specified by the user.

7.3.3 Machine Learning Approaches

The machine learning methods provide well-suited technique for automated finite element modeling recommendations. In this work, Artificial Neural Networks (ANNs) are implemented with a motivation to provide modeling recommendations within seconds.
Table 7.10: Comparison of multi-fidelity model and high-fidelity (full 3-D) results for a stepped plate ($\alpha = \beta = 0.1$).

<table>
<thead>
<tr>
<th>Model</th>
<th>Full 3-D</th>
<th>Multi-Fidelity</th>
<th>% Abs. Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Strain Energy (J)</td>
<td>8.376E-06</td>
<td>8.042E-06</td>
<td>3.98</td>
</tr>
<tr>
<td>Max. Disp. (m)</td>
<td>3.056E-06</td>
<td>2.939E-06</td>
<td>3.82</td>
</tr>
<tr>
<td>Max. Abs. Principal Stress (N/m$^2$)</td>
<td>982.324</td>
<td>986.325</td>
<td>0.41</td>
</tr>
<tr>
<td>Max. Abs. Principal Strain</td>
<td>11.209E-09</td>
<td>11.793E-09</td>
<td>5.21</td>
</tr>
<tr>
<td>Max. von Mises Stress (N/m$^2$)</td>
<td>770.819</td>
<td>848.331</td>
<td>10.05</td>
</tr>
</tbody>
</table>

**Artificial Neural Networks (ANNs)**

Using biological neural networks as an inspiration, Artificial Neural Networks (ANNs) represent the coupled interactions between the inputs and outputs using one or more hidden layers of neurons as shown in Fig. 7.24. Such ANNs are used to make optimal modeling recommendations in the geometries presented in Section 7.3.2.

A feedforward neural network consists of input, hidden and output layers of neurons as shown in Fig. 7.24. Each neuron in a hidden layer applies an activation function to its input. In the feedforward network, output of any given layer is provided as an input to the next layer. More specifically, the input into any given neuron is a weighted sum of the outputs of the neurons in the previous layers. There are no cyclic connections existing between the neurons and, thus, information always flows in the forward direction. Later, a backpropagation algorithm is used to determine the individual weights to minimize the error in the predictions and thereby “train” the neural network.

In order to train ANNs, a dataset of known inputs-outputs is created. It is divided into training, validation and testing sets. The training set is used for training of the ANN by finding the synapse weights. The validation set is used to decide when to stop the training of
the ANN, otherwise the ANN could overfit on the training set and would have low accuracy on the test set. Therefore, a validation set is provided to pick the optimal ANN. Finally, the testing set is used to check the accuracy of the trained ANN. In the presented case studies in this chapter, the ratio of training, validation and testing sets is 60%, 20% and 20%.

In the present work, Neural Network Toolbox of MATLAB is utilized for developing and implementing the use of ANN. The Neural Network Toolbox in MATLAB [108] includes various functions and algorithms to create, train and visualize and simulate neural networks. It can perform regression, clustering, classification and dynamic system modeling and control. In MATLAB, there is a set of steps that a user has to take to train ANNs. Initially a dataset is collected. Then, a neural network is created and configured. Later, weights and biases are initialized. Finally, the network is trained, validated and tested.

![Figure 7.24: Multi-Input/Multi-Output (MIMO) neural network with two hidden layers.](image)

7.3.4 Modeling Recommendations Schemes

Two modeling recommendations schemes are presented in the current work consisting of the following:
1. Modeling Recommendations Using Fixed Error Level

2. Modeling Recommendations Using Variable Error Level

The objective of these schemes is to provide the modeling recommendations such that 3-D region could be minimized for saving computational cost, while maintaining the accuracy of the results within certain required limit. The modeling recommendation can be provided either based upon a specific fixed limit (e.g., 5%) considered during machine learning training, or a user-desired variable limit specified after the machine learning has been achieved.

Restated as a minimization problem, the objective function, $f(x)$, for choosing optimal distribution is defined for T-joint as follows:

$$f(x) = \frac{1}{2-D \text{ Region Area}} = \frac{1}{\left(\frac{H - D_1}{(H - D_1) + (L - D_2)}\right)W} \quad (7.1)$$

where the geometric parameters are shown in Fig. 7.2 and Fig. 7.12.

Similarly, the objective function can be defined for stepped plate as follows:

$$f(x) = \frac{1}{2-D \text{ Region Area}} = \frac{1}{\left(\frac{L_1 - D_1}{(L_1 - D_1) + (L_2 - D_2)}\right)W} \quad (7.2)$$

where the geometric parameters are shown in Fig. 7.14 and Fig. 7.22.

Constraints on the errors of the following five result quantities are used: (1) maximum von Mises stress; (2) maximum absolute principal stress; (3) maximum absolute principal strain; (4) maximum displacement; and (5) total strain energy. Each constraint can be written as

$$g_i = \frac{E_i}{X_{user}} - 1 \leq 0 \quad (7.3)$$
where $E_i$ is the $i^{th}$ absolute % error with respect to the respective full 3-D model and $X_{user}$ is the percentage error bound. A penalty approach is used to convert constrained optimization problem to an unconstrained optimization with an objective function

$$h(x) = f(x) + K \left( \sum \max(g_i, 0)^2 \right)$$

(7.4)

where $K$ is a very large number.

**Modeling Recommendations Using a Fixed Error Level**

In this approach, a reduced grid-based optimization framework is implemented to find the optimal 2-D and 3-D region. Such a discrete approach is practical and prudent, as the end users are likely unconcerned with achieving the precise optimum model so long as their model is close enough. The objective of the optimization, as shown in Eqn. 7.4, is used to find optimal $\alpha$ and $\beta$. Instead of finding the true optimum which itself could take significant time, the near optimal point was determined by calculating the unconstrained objective function at discrete values of $\alpha$ and $\beta$ chosen so that T-Joint welded region and stepped plate’s step region is always modeled using 3-D elements. Initially, a dataset is created based on Latin Hyper-cube Sampling (LHS) of the geometric parameters. For each of the data point in the dataset, the following discrete values of $\alpha$ and $\beta$ are used to define a 3-D region of the structure: 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6. Then, the unconstrained objective function $h(x)$ is evaluated at $6\times6=36$ grid points ($\alpha$ and $\beta$) for any set of geometric and load input values. The optimum parameter values are chosen from the grid point having minimum value of the unconstrained objective function. These optimum values of $\alpha$ and $\beta$ are recorded for each data point in the dataset. Finally, an ANN is trained with the geometric parameters as the inputs and the optimal modeling recommendation as the output, as shown in Fig. 7.25. In
The current work, the optimal $\alpha$ and $\beta$ are recorded for 5% error limit.

The advantage of this approach is that the user gets direct modeling recommendations based on fixed error limit, e.g., 5% in current case without any further optimization. But, the disadvantage of this approach is that if the desired error limit has to be changed, let us say 10%, then a new ANN has to be trained. This disadvantage is overcome using the variable error level based modeling recommendation scheme.

![Geometric Inputs](image)

**Figure 7.25:** Machine learning modeling recommendations using fixed error level.

**Modeling Recommendations Using a Variable Error Level**

In this approach, ANNs are used such that modeling recommendations can be provided based on variable error level. Figure 7.26 shows the flowchart of the modeling recommendation scheme. In the first step, a dataset based on Latin Hyper-cube Sampling (LHS) of the geometric parameters is created. For each of the data point in the dataset, the following discrete values of $\alpha$ and $\beta$ are used to define a 3-D region of the structure: 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6. For each of these discrete values of $\alpha$ and $\beta$, errors in the five quantities
of interest are recorded. Finally, an ANN is trained with inputs to be geometric parameters along with $\alpha$ and $\beta$, and outputs to be multi-fidelity model’s error in five quantifies, as shown in Fig. 7.26.

Once the ANN is trained with high accuracy to predict the errors, then it is used in the reduced grid-based optimization framework to find the optimal 2-D and 3-D region. The benefit of using this approach is that during grid-based optimization of Eqn. 7.4, the user can change the value of error limit, $X_{\text{user}}$, in the constraint $g_i$. As the ANN is trained to predict the errors for different multi-fidelity models with different values of $\alpha$ and $\beta$, this approach gives freedom to the user to get modeling recommendations using an error level specified after the ANN has already been trained.

Figure 7.26: Machine learning modeling recommendations using variable error level.
7.4 Application and Results

The fixed and variable error level methodologies will be applied to several case studies. More specifically, machine learning techniques are used to provide direct modeling recommendations based on the fixed error level of 5% and a variable error limit for the T-joint and stepped plate geometries.

7.4.1 Case Study - T-joint

The T-Joint structure is considered with a 10m height. Both vertical $F_y$ and shear $F_z$ loading are considered and are applied as surface traction on the edge surface of the full 3-D model and as a distributed load on the shell edge of the multi-fidelity model. Four non-dimensional parameters consisting of three geometric variables and one load variables are defined as follows:

$$\gamma = \frac{L}{H}; \quad \delta = \frac{t}{H}; \quad \kappa = \frac{W}{H}; \quad \phi = \frac{F_y}{F_z}$$

where the geometric parameters are shown in Fig. 7.2 and Fig. 7.12.

Fixed Error Level

In this case study, 200 data points are generated using Latin Hyper-Cube Sampling (LHS) for four non-dimensionalized parameters within following range:

$$0.5 \leq \gamma, \kappa \leq 2.0; \quad 0.01 \leq \delta \leq 0.03; \quad 0.25 \leq \phi \leq 4.0$$

The sampling range is selected such that no T-Joint geometry becomes too large in one dimension in comparison to other dimensions. For each data point in the dataset, the
discrete values of $\alpha$ and $\beta$ are used to create multi-fidelity models and five quantities of interest are compared with their respective full 3-D model. The methodology explained in Section 7.3.4 is used to pick optimal values of $\alpha$ and $\beta$ based on 5% error limit.

The optimal values of modeling recommendations are used to train, validate and test an ANN which gives direct modeling recommendations. The ANN has 3 hidden layers of 5 neurons in each layer with the activation function of hyperbolic tangent sigmoid transfer function. Figure 7.27 shows the performance history of the ANN. As stated earlier, the training set is used to “train” the ANN by determining the optimal synaptic weights and a validation set is used to check when to stop the ANN training to prevent overfitting. It can be observed in the figure that after epoch 18, the mean squared error starts increasing on the validation set. Therefore, MATLAB Neural Network Toolbox stopped and chose the ANN corresponding to epoch 18 as the final fit. The ANN accuracy is checked over the test dataset. A prediction is considered accurate if the trained ANN predicted exact distribution of the 3-D and 2-D region for $\alpha$ and $\beta$. The accuracy of the trained ANN for training and testing set is 98% and 95% respectively. Table 7.12 shows prediction of modeling recommendations of some of the data points from the test set. It is apparent that the trained ANN is able to successfully predict the optimal values required of $\alpha$ and $\beta$.

<table>
<thead>
<tr>
<th>Input $\gamma = L/H$</th>
<th>Input $\kappa = W/H$</th>
<th>Input $\delta = t/H$</th>
<th>Input $\phi = F_y/F_z$</th>
<th>True Output $\alpha = D_1/H$</th>
<th>True Output $\beta = D_2/L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7475</td>
<td>0.8525</td>
<td>0.0233</td>
<td>0.5687</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>1.7975</td>
<td>0.7925</td>
<td>0.0265</td>
<td>1.9150</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>1.2575</td>
<td>1.5725</td>
<td>0.0125</td>
<td>2.4550</td>
<td>0.2</td>
<td>0.1</td>
</tr>
</tbody>
</table>
7.4. Application and Results

Figure 7.27: ANN performance improvement with training epochs for T-Joint using a prescribed 5% error limit.

Table 7.12: Modeling recommendation predictions using the trained ANN for the test data points (Table 7.11) in the T-joint problem.

<table>
<thead>
<tr>
<th>Input $\gamma = L/H$</th>
<th>Input $\kappa = W/H$</th>
<th>Input $\delta = t/H$</th>
<th>Input $\phi = F_y/F_z$</th>
<th>Predicted Output $\alpha = D_1/H$</th>
<th>Predicted Output $\beta = D_2/L$</th>
<th>Strain Energy %Error</th>
<th>Max. Disp. %Error</th>
<th>Max. von Mises %Error</th>
<th>Max. Princ. Stress %Error</th>
<th>Max. Princ. Strain %Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7475</td>
<td>0.8525</td>
<td>0.0233</td>
<td>0.5687</td>
<td>0.1</td>
<td>0.2</td>
<td>3.12</td>
<td>3.18</td>
<td>0.18</td>
<td>0.083</td>
<td>0.008</td>
</tr>
<tr>
<td>1.7975</td>
<td>0.7925</td>
<td>0.0265</td>
<td>1.9150</td>
<td>0.1</td>
<td>0.1</td>
<td>3.86</td>
<td>4.28</td>
<td>3.05</td>
<td>0.560</td>
<td>1.260</td>
</tr>
<tr>
<td>1.2575</td>
<td>1.5725</td>
<td>0.0125</td>
<td>2.4550</td>
<td>0.2</td>
<td>0.1</td>
<td>4.16</td>
<td>4.29</td>
<td>0.04</td>
<td>0.341</td>
<td>0.190</td>
</tr>
</tbody>
</table>

Variable Error Level

In this case study, 200 data points generated using for the four non-dimensionalized parameters in the Section 7.4.1 are used again. In this case study, the methodology explained in Section 7.3.4 is used. For each data point, a total of 36 discrete combinations of $\alpha$ and $\beta$ are used to create multi-fidelity models and the five quantities of interest are compared with their respective values from the full 3-D model. This resulted to 7,200 data points which are
used to train the ANNs.

The ANN will provide predictions of error levels for a given multi-fidelity representation of a particular geometry/load case. Hence, in order to train the ANNs, values of six parameters including four non-dimensionalized parameters and two parameters of $\alpha$ and $\beta$ are provided as inputs and five error quantities are provided as outputs for training, validation and testing. The ANN has 3 hidden layers of 30 neurons in each with activation function of hyperbolic tangent sigmoid transfer function. Figure 7.28 shows the epoch history of ANN performance during training. After epoch 14, the ANN starts over-fitting and the mean squared error starts increasing for validation and testing set. Thus, MATLAB ANN toolbox choses the best ANN model to be the one at epoch 14. Next, the accuracy of the ANN is evaluated on the test dataset. A prediction is considered to be accurate if ANN predicted error percentage within 2% error. For example, if the actual error level is 18%, then if ANN predicted any value within 16-20%, then that prediction is considered to be accurate. The accuracy of the ANN with this criteria over the training and testing set is 99% and 98% respectively. Table 7.13 shows error predictions for few data points from the test set where it can be noted that the trained ANN is able to predict errors very close to the actual error value.

Later, the trained ANN is used in the grid-based optimization framework to provide modeling recommendations for user-defined error levels. As an example, a data point was chosen from the testing set. The ANN was used to evaluate the error levels for five quantities of interest in Eqn. 7.4 for discrete values of $\alpha$ and $\beta$. Finally, the modeling recommendations are provided based on the minimum possible value of the objective function. Table 7.14 shows an example case for one of the geometry from testing set. Different modeling recommendations are provided if the user-input value is 5% or 10%. Thus, it shows that this approach has the benefit of providing modeling recommendations without having the error level specified during the actual machine learning training.
Figure 7.28: ANN performance improvement with training epochs for variable error level case study of T-Joint.

Table 7.13: Error prediction using ANN for data points from the test set for a T-joint.

<table>
<thead>
<tr>
<th>Data Point #</th>
<th>Strain Energy % Error</th>
<th>Max. Displ. % Error</th>
<th>Max. von Mises % Error</th>
<th>Max. Princ. Stress % Error</th>
<th>Max. Princ. Strain % Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.273</td>
<td>1.400</td>
<td>1.243</td>
<td>1.386</td>
<td>0.044</td>
</tr>
<tr>
<td>5</td>
<td>2.457</td>
<td>2.435</td>
<td>2.559</td>
<td>2.415</td>
<td>0.239</td>
</tr>
</tbody>
</table>

Table 7.14: Modeling recommendations for different error limits for a T-Joint.

<table>
<thead>
<tr>
<th>User Error Level</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>10%</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

7.4.2 Case Study - Stepped Plate

Next, a stepped plate structure is studied using both fixed error level and variable level based modeling recommendations. The stepped plate structure is considered with a $L_1 = 10$ m.
Both vertical $F_y$ and shear $F_z$ loading are considered and are applied as surface traction on
the edge surface of the full 3-D model and as a distributed load on the shell of the multi-
fidelity model. Five non-dimensional parameters consisting of four geometric variables and
one load variables are defined as follows:

$$
\begin{align*}
\gamma &= \frac{L_2}{L_1}; & \delta &= \frac{t_1}{t_2}; & \kappa &= \frac{t_2}{L_1}; & \eta &= \frac{W}{L_1}; & \phi &= \frac{F_y}{F_z}
\end{align*}
$$

(7.7)

where the geometric parameters are shown in Fig. 7.14 and Fig. 7.22.

**Fixed Error Level**

In this case study, 170 data points are generated using Latin Hyper-Cube Sampling (LHS)
for the five non-dimensionalized parameters within following range:

$$
0.5 \leq \gamma, \eta \leq 2.0; \quad 0.01 \leq \kappa \leq 0.03; \quad 2.0 \leq \delta \leq 4.0; \quad 0.25 \leq \phi \leq 4.0
$$

(7.8)

As conducted in the previous T-joint case study, for each data point in the dataset, discrete
values of $\alpha$ and $\beta$ are used to create multi-fidelity models and five quantities of interest
are compared with their respective full 3-D model values. The methodology explained in
Section 7.3.4 is used to pick optimal values of $\alpha$ and $\beta$ based on 5% error limit.

The optimal values of modeling recommendations are used to train, validate and test the
ANN. The ANN has 3 hidden layers of 5 neurons in each with activation function of hyper-
bolic tangent sigmoid transfer function. Figure 7.29 shows the performance history of ANN
training where epoch 19 is taken as the final ANN, as overfitting occurs in later training
epochs. The accuracy of trained ANN for training and testing datasets is 94% and 91%,
respectively. Table 7.16 shows prediction of modeling recommendations of some of the data points from the test set. This case study shows that the trained ANN is able to successfully predict the $\alpha$ and $\beta$ values defining the optimal multi-fidelity model representation.

![Best Validation Performance is 0.00050341 at epoch 19](image)

Figure 7.29: ANN performance improvement with training epochs for fixed error level case study of stepped plate.

Table 7.15: Few test data points in stepped plate problems.

<table>
<thead>
<tr>
<th>Input $\gamma = L_2/L_1$</th>
<th>Input $\eta = W/L_1$</th>
<th>Input $\kappa = t_2 = L_1$</th>
<th>Input $\phi = F_y/F_z$</th>
<th>Input $\delta = t_1/t_2$</th>
<th>True Output $\alpha = D_1/L_1$</th>
<th>True Output $\beta = D_2/L_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.512</td>
<td>1.977</td>
<td>0.016</td>
<td>0.756</td>
<td>2.242</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>1.887</td>
<td>1.107</td>
<td>0.017</td>
<td>0.463</td>
<td>2.261</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>0.882</td>
<td>1.902</td>
<td>0.025</td>
<td>1.915</td>
<td>3.819</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Table 7.16: Modeling recommendations using trained ANN for test data points (Table 7.15) in stepped plate problems.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma = \frac{L_2}{L_1}$</td>
<td>$\eta = \frac{W}{L_1}$</td>
<td>$\kappa = \frac{t_2}{L_1}$</td>
<td>$\phi = \frac{F_y}{F_z}$</td>
<td>$\delta = \frac{t_1}{t_2}$</td>
<td>$\alpha = \frac{D_1}{L_1}$</td>
<td>$\beta = \frac{D_2}{L_2}$</td>
<td>% Error</td>
<td>% Error</td>
<td>% Error</td>
<td>% Error</td>
</tr>
<tr>
<td>1.512</td>
<td>1.977</td>
<td>0.016</td>
<td>0.756</td>
<td>2.242</td>
<td>0.1</td>
<td>0.2</td>
<td>3.149</td>
<td>3.076</td>
<td>0.128</td>
<td>0.529</td>
</tr>
<tr>
<td>1.887</td>
<td>1.107</td>
<td>0.017</td>
<td>0.463</td>
<td>2.061</td>
<td>0.1</td>
<td>0.2</td>
<td>3.127</td>
<td>3.065</td>
<td>2.312</td>
<td>0.304</td>
</tr>
<tr>
<td>0.882</td>
<td>1.902</td>
<td>0.025</td>
<td>1.915</td>
<td>3.819</td>
<td>0.1</td>
<td>0.1</td>
<td>4.049</td>
<td>3.862</td>
<td>2.827</td>
<td>0.588</td>
</tr>
</tbody>
</table>

Variable Error Level

In this case study, 170 data points are generated for the five dimensionless parameters defined in Section 7.4.2. Now the variable error methodology explained in Section 7.3.4 is used to give modeling recommendations for error levels specified after the machine learning has already taken place. For each data point in the dataset, a total of 36 discrete combinations of $\alpha$ and $\beta$ are used to create multi-fidelity models and the five error quantities of interest are compared with their respective values from the full 3-D model. This resulted in 6,120 data points which are used to train the ANNs.

In order to train the ANN to provide predictions of error levels for any given multi-fidelity representation of a particular structure and loading ratio, seven parameters including five dimensionless parameters along with two multi-fidelity modeling parameters ($\alpha$ and $\beta$) are provided as input and five error quantities are provided as output for the training, validation and testing of the ANN. The ANN has 2 hidden layers of 20 neurons in each with activation function of hyperbolic tangent sigmoid transfer function. Figure 7.30 shows the epoch history of ANN performance during training. The best epoch of training is 91. Similar to before, a prediction will be considered to be accurate if the ANN predicted error percentage is within 2% of the actual value. The accuracy of the ANN with this criteria for both the training and testing set is 99%. Table 7.17 shows error predictions for a few selected data points from the test set. Finally, the trained ANN can be used to predict the % error in Eqn. 7.4 for
discrete values of $\alpha$ and $\beta$ for any geometry input. The optimal combination of modeling recommendations is selected to be the one corresponding to one with minimum objective function value. An example of such a case is shown in Table 7.18.

![Best Validation Performance is 0.10341 at epoch 91](image)

Figure 7.30: The ANN performance improvement with training epochs for variable error level case study of stepped plate.

<table>
<thead>
<tr>
<th>Data Point</th>
<th>Strain Energy % Error</th>
<th>Max. Disp % Error</th>
<th>Max. von Mises % Error</th>
<th>Max. Princi. Stress % Error</th>
<th>Max. Princi. Strain % Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3.041</td>
<td>2.929</td>
<td>2.963</td>
<td>2.938</td>
<td>0.104</td>
</tr>
<tr>
<td>4</td>
<td>3.803</td>
<td>3.814</td>
<td>3.512</td>
<td>3.683</td>
<td>11.000</td>
</tr>
<tr>
<td>5</td>
<td>3.186</td>
<td>3.507</td>
<td>3.021</td>
<td>3.426</td>
<td>4.514</td>
</tr>
</tbody>
</table>

Table 7.17: Error prediction using ANN for select data points from the test set for the stepped plate example.
Table 7.18: Modeling recommendations based on different error limits for a stepped plate problem.

<table>
<thead>
<tr>
<th>User Error Level</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>15%</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

7.5 Conclusions

Machine learning approaches have been developed and presented for finite element modeling recommendations. The decisions about the selection of appropriate modeling techniques, including type of elements, boundary conditions, etc., are usually based on an analyst’s judgement based on their knowledge and intuition built over years. In the present work, an attempt is made to remove the subjective nature of such decision making by a multi-level qualification scheme where high-fidelity models are used to judge the sufficiency of low and mid-fidelity models for a broad range of geometric and loading parameters. A design tool is developed with a motivation to provide a design agent with the specific modeling and fidelity requirements for modeling complex geometries.

Two simple geometries are considered to demonstrate the approach utility in a proof-of-concept manner. A multi-fidelity modeling approach is presented for these structures. The machine learning techniques are used to provide automated modeling recommendations for multi-fidelity models such that the computational cost could be minimized while maintaining desired accuracy in comparison to respective full 3-D models. Two different modeling recommendations schemes are presented where the recommendations can either be provided based on the fixed error limit specified during the machine learning or a variable error limit specified after the machine learning. The results show that ANNs can successfully be used to accurately provide modeling recommendations. Once the machine learning has been performed, this approach provides recommendations within seconds. Such a methodology is
best utilized for common geometries for which a “handbook” collection of trained ANNs can be utilized to guide multi-fidelity modeling decisions.
Chapter 8

Summary and Future Work

The present work takes a step towards improving structural design and optimization using machine learning (ML). It is shown that the finite element analysis (FEA) based standard optimization methods for aircraft panels, with bio-inspired curvilinear stiffeners, are very expensive and that even parallel processing could only be of limited value. The use of machine learning techniques, like deep learning and active learning, may be another avenue for us to pursue. The use of machine learning is thus proposed to accelerate the structural design and optimization by replacing the FEA evaluations using the machine learning-based surrogate models. It is shown that ML has a great potential to accelerate the structural optimization problems. The first three chapters of this thesis presented the use of different optimization schemes on complex non-convex optimization problems. These schemes include: 1) a two-step iterative use of a gradient-free optimization technique, and a gradient-based optimization technique; 2) a gradient-free particle swarm optimizer (PSO) based optimization; and 3) a bi-level optimization technique. These optimization approaches require a very large number of evaluations of the FEA. Parallel processing is shown to be an alternative to reduce the required CPU run time.

The DNNs are successfully used as surrogate models to replace expensive FEA evaluations. However, large datasets are needed for getting highly accurate surrogate model. An active learning-based GA is proposed to accelerate the evolutionary optimization of complex structures. This approach is quite general, and can be applied to any structural optimization
problem. An advantage of the proposed approach is that it does not need a trained surrogate model prior to optimization. The active learner adaptively learns about the structure during the optimization to improve the computational performance. The results show that the approach has the potential to reduce the total required constraint evaluations by 50% to 70%.

Lastly, machine learning software are developed to provide modeling recommendations for creating valid models for arbitrary geometric and load parameters. The approach showed that the software can rapidly provide modeling recommendations. Moreover, the machine learning software can be updated as new knowledge about the simulations of structure’s behavior becomes available by retraining the underlying ANN using updated training data.

In the future, following research objectives could be pursued:

- The research presented in Chapter 5 showed the use of DNNs for designing panel stiffened by only two stiffeners. The number of stiffeners can be used as an additional input to the DNN. Also, different shear loads can be considered. This would make the approach to be more general for designing panels with different number of stiffeners.

- The active learning-based approach showed in Chapter 6, can be used in the two-step iterative optimization framework shown in Chapter 2. The design variables can be divided into two categories, depending on the requirement if they need a gradient-based or a gradient-free optimization technique. As shown in Chapter 2, most of the optimization run-time is taken by the gradient-free approach (PSO). The PSO could be replaced with ALGA for accelerating the optimization. This would help with reducing the total required FEA evaluations for the optimization.

- Different active learners based on support vector machines (SVMs), $k$-nearest neighbours (kNNs) can be studied on different optimization problems.


Effects of Stiffness Variation on the Buckling and Failure Responses,” *7th EUROMECH Solid Mechanics Conference, Portugal*, 2009.


