Structural Modeling and Optimization of Aircraft Wings having Curvilinear Spars and Ribs (SpaRibs)

Shuvodeep De

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 Engineering Mechanics

Rakesh K. Kapania, Chair
Mayuresh J. Patil
Douglas P. Holmes
Mark Cramer
Muhammad R. Hajj

August 10, 2017

Blacksburg, Virginia

Keywords: Multidisciplinary Optimization, Finite Element Methods, Aeroelastic Analysis, Buckling, Parallel Computing

Copyright 2017, Shuvodeep De
Structural Modeling and Optimization of Aircraft Wings having Curvilinear Spars and Ribs (SpaRibs)

Shuvodeep De

ABSTRACT

The design and optimization of an aircraft structure is a very challenging task which involves many disciplines, each characterized by its own set of requirements, constraints and objectives. The development of additive manufacturing techniques like Electron Beam Free Form Fabrication (EBF3) has made it feasible to manufacture aircraft wings with curvilinear spars and ribs (SpaRibs). The SpaRibs topology, for aerospace structures such as wings, has a significant effect on structural and aeroelastic behavior of an aircraft wing. The original EBF3GLWingOpt developed by Liu et al. at Virginia Tech is a multi-scale optimization framework intended to perform multi-disciplinary optimization of the internal structure of subsonic transport aircraft wing with curvilinear spars and ribs (SpaRibs) with constraints on stress and buckling. The optimization framework consists of two sub-systems: the global optimizer (known as EBF3WingOpt) for the number and topology of the SpaRibs and the local optimizer (known as EBF3PanelOpt) for the thicknesses of local panels extracted from the global model. In structural analysis of the local panel displacement boundary condition is implemented and the displacement values are interpolated from the displacement field obtained by static aeroelastic analysis of the global structure. The global-local framework is written in Python environment and it incorporates MSC.PATRAN for modelling the geometry and MSC.NASTRAN for finite element analysis and has capabilities for parallel computation. The following work is an endeavor to develop a much more generalized version of EBF3GLWingOpt framework that can be used to perform multi-objective optimization for the internal structure of a wide range of aircraft with SpaRibs. The original EBF3GLWingOpt framework has a number of limitations. Firstly, it was developed just for the NASA CRM and cannot be used for other types of wings without major modifications. Secondly, the SpaRibs parameterization was such that the ribs can only start at the leading edge spar and end on the trailing edge spar. In addition, although the optimization framework takes into
account the buckling constraint of each skin panel, it does not ensure that the buckling constraint
of the global wing will be satisfied for the final design even if the weight converges within the
specified limit. In this work, the approach is modified to overcome the aforementioned limitations.
Using this new version of the optimization framework, it is possible to create SpaRibs that start at
any edge of the wing-box and ends at any other edge. This greatly increased the range of possible
SpaRibs topologies that can be considered in the optimization problem. The framework also finds
the weak panels in the final design after weight convergence is achieved and systematically
increases their thicknesses to ensure that all buckling constraints are satisfied. SpaRibs are
represented using a lesser number of parameters compared to the original EBF3GLWingOpt
framework. A reduction in the number of design variables will significantly reduce the
computational cost. In addition to these advantages, the framework also possesses a capability to
create an array of cut-outs in the SpaRibs for passage of fuel pipes and wirings. It was found that
sometimes MSC-PATRAN fails to generate the complete mesh for structures with complicated
SpaRibs geometries. The code reduces the chance of mesh failure by allowing small and
insignificant imperfections to the geometry. The multi-objective problem considered to be solved
using this framework is to reduce weight and increase the flutter velocity of the NASA CRM wing
while satisfying the SpaRibs with constraints on stress and buckling.
Shuvodeep De

**Structural Modeling and Optimization of Aircraft Wings having Curvilinear Spars and Ribs (SpaRibs)**

Shuvodeep De

**GENERAL AUDIENCE ABSTRACT**

The aviation industry is growing at a steady rate but presently, the industry is highly dependent on fossil fuel. As the world is running out of fossil fuels and the wide-spread acceptance of climate change due to carbon emissions, both the governments and industry are spending a significant amount of resources on research to reduce the weight and hence the fuel consumption of commercial aircraft. A commercial fixed-wing aircraft wing consists of spars which are beams running in span-wise direction, carrying the flight loads and ribs which are panels with holes attached to the spars to preserve the outer airfoil shape of the wing. Kapania et al. at Virginia Tech proposed the concept of reducing the weight of aircraft wing using unconventional design of the internal structure consisting of curvilinear spars and ribs (known as SpaRibs) for enhanced performance. A research code, EBF3GLWingOpt, was developed by the Kapania Group at Virginia Tech to find the best configuration of SpaRibs in terms of weight saving for given flight conditions. However, this software had a number of limitations and it can only create and analyze limited number of SpaRibs configurations. In this work, the limitations of the EBF3GLWingOpt code has been identified and new algorithms have been developed to make is robust and analyze larger number of SpaRibs configurations. The code also has the capability to create cut-outs in the SpaRibs for passage of fuel pipes and wirings. This new version of the code can be used to find best SpaRibs configuration for multiple objectives such as reduction of weight and increase flutter velocity. The code is developed in Python language and it has parallel computational capabilities. The wing is modeled using commercial FEA software, MSC.PATRAN and analyzed using MSC.NASTRAN which are from within EBF3GLWingOpt. Using this code a significant weight reduction for a transport aircraft wing has been achieved.
Dedication

I dedicate this dissertation to my parents, all artists and innovators who inspires me to dream of a better tomorrow.
Acknowledgement

Firstly, I would like to express my deep gratitude to my adviser Dr. Rakesh Kapania for his guidance throughout my graduate research and helping immensely to develop my background in structural engineering. I am really thankful to my coworkers Wrik Mallik, Qiang Liu, Mohamed Jrad, Wei Zhao, Davide Locatelli, Karanpreet Singh, Rikin Gupta and Nathan Love for several technical discussions. I would also like to express my sincere thanks to all my committee members: Dr. Mayuresh Patil, Dr. Muhammad R. Hajj, Dr. Douglas Holmes and Dr. Mark Cramer for their valuable feedbacks on my work.

This research was funded by the NASA SBIR/STTR program, under contracts NNX14CD16P and NNX15CD08C. I am thankful to Dr. Myles Baker, Joe Robinson, Steven Doyle, Grant Ogawa and Vananh Ho of M4 Engineering, Inc. and Dr. Chan-gi Pak of NASA Armstrong Flight Research Center for their sincere collaboration.

The original development of the EBF3PanelOpt was funded by NASA Langley Research Center through a grant administered by the National Institute of Aerospace. I am thankful for that support and the guidance from Karen Taminger, Marcia Domack, Carol Wieseman, Christine Jutte and Brett Stanford.

Finally, I must say that I really grateful to all my friends and all my favorite artists who help me to stay positive in life and get through tough times.
# Table of Contents

Table of Contents ........................................................................................................................................ vii

Table of Figures .......................................................................................................................................... xi

Table of Tables ............................................................................................................................................ xv

Nomenclature .............................................................................................................................................. xvi

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

1.1 Background ........................................................................................................................................ 1

1.2 Multidisciplinary Structural Optimization ..................................................................................... 2

1.3 Optimization involving Curvilinear Stiffening Members .............................................................. 4

1.3.1 Constraints in Aerospace Structural Optimization ................................................................ 5

1.3.2 Load Conditions using Static Aero-elastic Analysis ................................................................ 9

1.4 Structural Optimization Algorithms ............................................................................................... 9

1.4.1 Mathematical Statement of Optimization .................................................................................... 10

1.4.2 Particle Swarm Optimization ...................................................................................................... 11

1.4.3 The Weighted Sum Method ....................................................................................................... 13

1.4.4 Solving Multi-Objective Problems using PSO .......................................................................... 14

1.5 Research Objectives ........................................................................................................................ 17

2 Original EBF3WingOpt Framework .................................................................................................... 18
2.1 SpaRibs Parameterization.......................................................................................... 19
2.2 Demonstration using CRM Wing................................................................................ 20
2.3 Boeing HSCT N+2 Rear Wingbox Construction by Linked-shape Method............... 24
2.4 Local Panel Optimization (EBF3PanelOpt module).................................................. 25
2.5 Local Panel Generation by the In-out Algorithm....................................................... 28
2.6 Creation of Stiffeners ................................................................................................ 31
2.7 Integration of EBF3WingOpt and EBF3PanelOpt.................................................... 32
2.8 Parallel Computation................................................................................................ 32
2.9 Chapter Summary...................................................................................................... 34

3 Geometry Parameterization and Mesh Generation...................................................... 36

3.1 Enhance SpaRibs Topology Optimization Tools ...................................................... 36
  3.1.1 Method of Lines................................................................................................... 38
  3.1.2 Extended Design Space Method........................................................................ 41
  3.1.3 Determining Point Locations on Curves............................................................ 43
  3.1.4 Examples of SpaRibs Topology.......................................................................... 44
  3.1.5 Ribs Cut-out........................................................................................................ 46
  3.1.6 Creation of Stiffeners.......................................................................................... 48
  3.1.7 Mesh Modification and Removal of Bad Elements............................................ 49
3.2 Geometry and Mesh Failure in PATRAN................................................................. 50
  3.2.1 Possible Causes of Geometry Failure................................................................. 50
  3.2.2 Possible Causes of Mesh Failure......................................................................... 50
  3.2.3 Checking for Mesh Failure.................................................................................. 52
3.3 Chapter Summary...................................................................................................... 53

4 Generalized Local Panel Extraction ......................................................................... 54

4.1 Mesh Continuity Algorithm....................................................................................... 54
5.8 Baseline Model and Mesh Convergence ................................................................. 90
5.9 *SpaRibs* Topology Optimization using PSO .......................................................... 92
5.10 *SpaRibs* Optimization Examples ........................................................................ 94
   5.10.1 Case 1: Minimize Weight ................................................................................. 95
   5.10.2 Case 2: Minimize Weight and Maximize Flutter Velocity ............................... 97
5.11 : Final Results after Mesh Refinement .................................................................... 98
5.12 Discussions ............................................................................................................. 102
5.13 Chapter Summary ................................................................................................... 103
6 Conclusion and Future Work ...................................................................................... 104
7 References .................................................................................................................. 106
# Table of Figures

Figure 1-1: Upper Skin of Wing Showing Global Buckling ............................................. 6  
Figure 1-2: Upper Skin of Wing Showing Local Buckling ............................................ 7  
Figure 1-3: Velocity-damping ($V_g$) and Velocity-frequency ($V_n$) Diagram .................. 8  
Figure 1-4: Particle Position Updating in PSO Problem ................................................ 12  
Figure 1-5: MOPSO Pareto-front from (a) MATLAB code developed by Yarpiz Team (b)  
Python Script developed in VT .................................................................................. 15  
Figure 1-6: Iteration History for the given Objective ..................................................... 16  
Figure 2-1: Linked-shape Method ................................................................................. 19  
Figure 2-2: Outer Mold Line (OML) of the NASA CRM Wing ...................................... 20  
Figure 2-3: (a) SpaRibs Profile (b) Aerodynamic Mesh (c) Spline nodes on Upper Skin .... 23  
Figure 2-4: Internal Structure of Boeing HSCT Wing constructed using Linked-shape Method 24  
Figure 2-5: (a) Geometry of Rear Wing-box of Boeing HSCT N+2 Wing using Linked-shape  
Method (b) Mesh using Triangular Elements .................................................................. 25  
Figure 2-6: Cantilever Wing with Stiffened Panels [35] .................................................. 26  
Figure 2-7: Global-Local Process in EBF3PanelOpt (Qiang Liu [64]) ............................... 27  
Figure 2-8: (a) SpaRibs Profile (b) Iterative History in EBF3PanelOpt ............................ 27  
Figure 2-9: Extraction Method ...................................................................................... 29  
Figure 2-10: SpaRibs Configuration. (a) Local Panel Algorithm works (b) Does not work .... 30  
Figure 2-11: Stiffeners Shape Parameterization using Linked-shape method .................... 31  
Figure 2-12: Simple Task Running 5 Jobs using License Cycle-check Method ............... 33  
Figure 2-13: Simple Task Running 5 Nastran Jobs Using Memory Self-adjustment Method .... 33
Figure 3-1: The SpaRibs Profile. CRM Wing. Baseline Model (Left) Curvilinear (Right) .......... 37
Figure 3-2: Discontinuous SpaRibs at Trailing-edge Break ........................................... 38
Figure 3-3: Control-line for Construction of Curvilinear SpaRibs .................................... 38
Figure 3-4: The Four Possible Configurations of a Rib inside the Wing-box ....................... 39
Figure 3-5: The Eight Possible Configuration of a Curvilinear Ribs in the CRM Wing ........ 40
Figure 3-6: Ensuring C0 continuity at Trailing-edge between Projection Curves ................. 41
Figure 3-7: Extended Space Method to create SpaRibs ................................................. 42
Figure 3-8: Extended Space Method to create curvilinear SpaRibs ................................ 42
Figure 3-9: SpaRib Topology using the Extended Space Method ........................................ 44
Figure 3-10: Curvilinear SpaRibs Configurations with the Extended Space Method ........ 45
Figure 3-11: Mesh-contiuity at the Trailing-edge break ..................................................... 46
Figure 3-12: Process of creating Cut-outs in MSC.PATRAN .......................................... 46
Figure 3-13: Boeing N+2 HSCT Wingbox with Oval cut-outs ........................................ 47
Figure 3-14: Process of creating Cut-outs using Surface Manifold in MSC.PATRAN .......... 47
Figure 3-15: SpaRibs with Cut-outs .................................................................................. 48
Figure 3-16: Process of Creating Stiffeners ........................................................................ 48
Figure 3-17: Stiffeners (black) between Spars ................................................................. 49
Figure 3-18: Removal of Highly Skewed Elements ......................................................... 49
Figure 3-19: Geometry Failure in MSC.PATRAN ............................................................. 50
Figure 3-20: Possible Causes of Mesh Failure ..................................................................... 51
Figure 3-21: Avoiding associating Stiffeners smaller than the Maximum Element Size ...... 51
Figure 3-22: Stiffeners on Lower Skin near the Trailing-edge Break ............................... 51
Figure 3-23: Mesh-verification Algorithm .......................................................................... 52
Figure 4-1: Determining Panel Elements by Boundary Elements (red) and an Interior Node (black) ..................................................................................................................... 55
Figure 8-2: Mesh Continuity Algorithm to find Elements inside Local Panel .................. 55
Figure 4-3: Numbering the Local Panels ........................................................................ 56
Figure 4-4: Plate are Split by Three Non-Intersecting Curves ......................................... 57
Figure 8-5: Algorithm to determine Elements along the Outer (free) Boundary ............. 58
Figure 8-6: Middle Element Algorithm .......................................................................... 59
Figure 8-7: Middle Element Algorithm .......................................................................... 59
Figure 8-8: Algorithm to find elements on one side of Curve 2 ................................................. 60
Figure 4-9: Using Middle Element Algorithm to find Elements in Local Panel ............................. 60
Figure 4-10: Local Panel from Upper Skin of Boeing N+2 HSCT Wingbox ................................. 61
Figure 4-11: Inner and Outer Section of Wing FEM ..................................................................... 62
Figure 4-12: Sectioning the FEM using the Intersection of the Second Rib .............................. 62
Figure 4-13: Common Elements between two Sections ............................................................. 63
Figure 4-14: Local Panel creation near Wing-root using Mesh-continuity Algorithm .................. 64
Figure 4-15: Local Panel with Stiffeners (formed by Quad Elements) ......................................... 64
Figure 4-16: Part of the Skin with Stiffeners as created by the Original EBF3GLWingOpt ........... 65
Figure 4-17: Rectangular Plate with Stiffeners and Buckling Eigen Vector ............................... 65
Figure 4-18: (a) Mesh profile with Triangular Elements of Size 0.6 (b) First Buckling Eigenvalue
(c) Comparison of first 10 Buckling Eigenvalues ..................................................................... 66
Figure 4-19: Buckling Results Comparison between Mesh of Triangular Elements of Size 1.315
and Quadrilateral Elements of Size 0.6 ...................................................................................... 67
Figure 4-20: Buckling Results Comparison between Mesh of Triangular Elements of Size 2.0
and Quadrilateral Elements of Size 0.6 ...................................................................................... 67
Figure 4-21: Buckling Results Comparison between Mesh with Triangular Elements of Size 2.88
and Quadrilateral Elements of Size 0.6 ...................................................................................... 68
Figure 4-22: Buckling Results Comparison between Mesh with Size 5 (where Quadrilateral
Elements used for Stiffeners and Triangular Elements used for Plate) and Mesh with
Quadrilateral Elements of Size 0.6 ............................................................................................. 68
Figure 5-1: Complete Optimization Framework ........................................................................... 73
Figure 5-2: Boundary Conditions: (a) Skin panel, (b) Rib, (c) Spar ............................................. 74
Figure 5-3: Global-local Optimization of Aircraft Wing ............................................................... 75
Figure 5-4: Local Panel created from the Upper Skin of Boeing N+2 ........................................ 75
Figure 5-5 Iteration History of Panel Weight and Panel Thickness ............................................. 76
Figure 5-6: The Critical Bucking Mode Shape of Original and Optimal Panel Configurations .. 77
Figure 5-7: Sorting out nodes from Global model with Displacement above Threshold .......... 78
Figure 5-8: Selective Stiffening Algorithm (Step 1) ................................................................. 78
Figure 5-9: Selective Stiffening Algorithm (Step 2) ................................................................. 79
# Table of Tables

Table 1-1: PSO Parameters........................................................................................................... 12
Table 2-1: Linked-shape Parameters .......................................................................................... 20
Table 2-2: Design Variable Descriptions....................................................................................... 21
Table 2-3: Material Properties ...................................................................................................... 22
Table 2-4: Design Variables for Stiffeners on the Local Panel ......................................................... 31
Table 3-1: Offset parameters for Curvilinear SpaRibs Configurations ......................................... 45
Table 5-1: Global-Local Optimization of Several Wing Configurations ........................................ 87
Table 5-2: Global-Local + Selective Stiffening ............................................................................. 87
Table 5-3: Difference in Weight and Flutter Speed......................................................................... 87
Table 5-4: Weight Estimation and Computational Time for Different number of Elements/Nodes for CRM Baseline Model ........................................................................................................... 91
Table 5-5: Variation of Weight and Flutter Velocity with Number of Stiffeners ......................... 94
Table 5-6: Offset Parameters and Weight of Particles Analyzed .................................................. 95
Table 5-7: Structural Properties of Model 1 .................................................................................. 99
Table 5-8: Structural Properties of Model 2 ................................................................................. 100
Table 5-9: Structural Properties of Model 3 ................................................................................. 101
## Nomenclature

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDO</td>
<td>Multi-disciplinary Optimization</td>
</tr>
<tr>
<td>EBF³</td>
<td>Electron Beam Freeform Fabrication</td>
</tr>
<tr>
<td>CRM</td>
<td>Common Research Model</td>
</tr>
<tr>
<td>EAS</td>
<td>Equivalent Air Speed</td>
</tr>
<tr>
<td>V-g</td>
<td>Velocity Damping</td>
</tr>
<tr>
<td>V-n</td>
<td>Velocity Frequency</td>
</tr>
<tr>
<td>BF</td>
<td>Buckling Factor</td>
</tr>
<tr>
<td>OML</td>
<td>Outer Mold Line</td>
</tr>
<tr>
<td>AR</td>
<td>Aspect Ratio</td>
</tr>
<tr>
<td>O₁₁, O₁₂, O₂₁, O₂₂, O₃₁, O₃₂</td>
<td>Offset parameter</td>
</tr>
<tr>
<td>TIF</td>
<td>Thickness Increment Factor</td>
</tr>
<tr>
<td>AOA</td>
<td>Angle of Attack</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>MOPSO</td>
<td>Multi-objective Particle Swarm Optimization</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>σ₀m</td>
<td>von-Mises Stress</td>
</tr>
<tr>
<td>σ₀y</td>
<td>Yield Stress</td>
</tr>
<tr>
<td>λₚ</td>
<td>Buckling eigenvalue</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Background

It is expected that the aviation industry will see an annual growth of 5% over the next twenty years [1]. As of now, the industry is highly dependent on fossil fuel. As the world is running out of fossil fuels and the widespread acceptance of climate change due to carbon emissions, both the governments and industry are spending a significant amount of resources on research to reduce the weight and hence the fuel consumption of commercial aircraft. There has been a substantial research recently to use methods of topology optimization to reduce structural weight and enhance the aeroelastic performance of aircraft [2] [3] [4] [5] [6] [7] [8] [9] [10] [11]. However, the obvious drawback of these methods is that the optimized design is often too complex to study and to manufacture commercially. However, Kapania et al. at Virginia Tech pioneered a more feasible solution to reduce weight of aircraft using curvilinear structural members i.e. spars, ribs and stiffeners [12]. It is possible to manufacture these structures using new additive manufacturing techniques like the Electron Beam Free Form Fabrication approach [13].

The structural design optimization of an aircraft is a very challenging task where different disciplines, such as aerodynamics, structural analysis, propulsion, control and dynamics, and the manufacturing and operating costs needed to be considered. The process consists of the optimum internal structural layout and sizing of individual structural components based on the aerodynamic
loads and the aeroelastic response of the structure. The interaction between the structure and the aerodynamics is extremely important and must be considered inside the optimization framework. Although the weight reduction is the most important objective, it often comes at the price of worsening of the performance in some other discipline. Hence the optimization process often involves multiple optimization. Over nearly a decade, Kapania et al. developed a multidisciplinary, multi-objective optimization framework [14] for the internal structure of subsonic commercial aircraft wing with curvilinear stiffening elements. This framework is known as EBF3GLWingOpt. The ‘EBF3’ part of this term is the short form of Electron Beam Free Form Fabrication. This is because the main motivation behind development of this optimization process was the fact that it is feasible to manufacture wing with curvilinear structural components using such additive manufacturing techniques. The ‘GL’ part stands for Global-local. This is because the optimization involves multi-scale (often called global-local) analysis.

1.2 Multidisciplinary Structural Optimization

Multidisciplinary Design Optimization (MDO) is a branch of engineering where the goal is to solve design problems using optimization techniques while incorporating more than one discipline. The purpose for incorporating several disciplines in the optimization problems is because the dependence of optimum performance of a system on each of the relevant discipline is often not independent of each other, in other words the disciplines interact with each other. MDO find applications in various fields like electronics, power systems, navel architectures, automobile engineering etc. but the field where it is used the most appears to be aerospace.

The multi-disciplinary optimization of structures can be traced back to the work of Schmit [15] [16] [17] where numerical optimization techniques were integrated with finite element analysis. Since 1970, multi-disciplinary design optimization is increasingly gaining popularity in aircraft industry, especially due to the exponential increase in available computational capacity. Haftka [18] [19] [20], Fulton [21] and their colleagues pioneered the optimization of aircraft wing considering multiple constraints, such as strength, flutter velocity and buckling. Multidisciplinary design optimization techniques have also been applied for problems involving complete aircraft models [22] [23] [24] [25] [26]
Multidisciplinary Optimization (MDO) procedures in general can be classified as monolithic architecture and distributed architectures according to the survey by Martines [10]. The monolithic architecture employs the All-at-Once optimization [27], which is where all design variables used to describe the system and constraints of various disciplines are considered in a single optimization process. This All-At-Once optimization procedure is simple to implement because all it requires is to parameterize the system. A recent use of such a framework can be found in the work of Singh et al. [28]. However, if too many design variables and constraints are needed to be considered in the optimization problem it becomes very computationally expensive. Users often are limited by computational resources and it becomes infeasible to solve the problem. Such problems with a large number of design variables or constraints, are often solved by a multiple-level optimization approach [29]. In each level of optimization only a fraction of the total design variables or constraints are considered in each step of the optimization process. In the work of Locatelli et al. [29] such a two-step optimization process is used for supersonic wing structure (HSBT Boeing N+2 Wing), in which all the design variables are classified as shape design variables or size design variables. In each of these approaches, the first step deals with optimization of the shape design variables. The second step is then to refine the size design variables for each combination of the shape design variables.

The distributed MDO architectures are mostly used to solve problems involving highly complex systems where the system is decomposed into multiple smaller components to make it feasible to use parallel computation to reduce the overall time of optimization. Numerous algorithms have been proposed to decompose the system so that reliable results can be obtained. Kroo et al. studied the so called collaborative optimization [30] [31] which allows each discipline to solve its sub-system problem in parallel. In the work of Sobieszczanski-Sobieski et al. [32] [33] the optimization problem consists of two levels: system level and subsystem level. The design variables describing the structure at global level are optimized in the system level. In the subsystem level, the system is decomposed into multiple components and independent optimization problems each involving variables describing a single component are solved. The design variables and constraints for subsystem optimization are evaluated at the system level. This process of decomposition is often known as global/local design optimization.
Shuvodeep De

The aircraft wing is a complex system for which this kind of global/local optimization have been used by Cimpa et al. [34] and Liu et al. [35]. In their research then the global finite element model of the wing is divided into several local panels. The thicknesses of local panels were refined by optimizing the stiffened local panels to minimize the structural mass.

1.3 : Optimization involving Curvilinear Stiffening Members

Over the last several years, Taminger and Hafley [13] at NASA Langley Research Center developed a novel additive manufacturing technique known as electron-beam-free-form-fabrication (EBF$^3$), which enables the manufacture of metallic structures of arbitrary shapes. This new technology inspired Kapania and his group at Virginia Tech [29] [29] towards developing a family of computational frameworks to optimize the shape of the internal structure of an aircraft wing with curvilinear stiffening members i.e. curvilinear spars and ribs (SpaRibs) as well as blade stiffeners.

Following the work done by Kapania et al. other groups too have also started to use innovative and unconventional designs for the internal structure of aircraft wings. Robinson et al. [37] at M4-Engineering Inc. also developed a set of tools and techniques for modeling and optimizing the structure of a wing with curvilinear SpaRibs. It has already been proven [38] [39] [40] that curvilinear stiffeners can improve the buckling resistance of arbitrary shaped panels. Locatelli et al. [29] optimized the internal structure of an aircraft wing structure with SpaRibs and showed their potential to minimize the weight of a supersonic civil transport vehicle. Jutte et al. [41] performed extensive parametric studies using the NASA Common Research Model (CRM) varying the number, location and curvature of the stiffening members, i.e. spars, ribs and stiffeners, and demonstrated that configurations characterized by weight lower than the baseline aircraft and higher flutter speed exist. Doyle et al. [42] at M4-Engineering Inc. also showed the potential of SpaRibs to increase flutter speed. Francois et al. showed that the planform geometry of the spars and stringer has significant effect on the flutter velocity of a wind-tunnel wing model [43]. In the work of Stanford et al. [44] [45], a comparison between tow steered composite laminate skins and curvilinear stiffeners and showed how both the technology can reduce the weight of high-aspect ratio transport aircraft wing.
Curvilinear stiffeners were also found to influence the vibrational and acoustic performance of plates and shell. This has been studied extensively by Shi et al. [46] [47] as well as Singh et al. [28]. Zhao et al. [48] [49] studied the same for composite panels. Joshi et al. [50] performed multi-objective optimization of stiffened panels to reduce weight for minimum radiated acoustic power.

1.3.1 Constraints in Aerospace Structural Optimization

The three most important constraints that must be satisfied in wing design problem are (a) Stress (b) Buckling (c) Flutter

1.3.1.1: Stress

Stress is the force per unit area at a point in a continuum. It is a tensor with nine components and each component is dependent on the direction of force and the orientation of the surface on which it is acting. For a ductile material, the stress level at which the material starts to deform plastically is known as yield stress. According to von Mises yield criterion, a ductile material yields when square root of the second deviatoric stress invariant reaches the maximum shear stress.

Aerodynamic forces acting on a wing causes deformation and hence stresses in the structure of a flexible aircraft wing. The von-Mises Stress at every point in the aircraft structure must be less than the yield stress of the material considered. The yield stress of 7000 series Aluminum, commonly used for aerospace applications is around 48000 psi. During optimization of the aircraft wing the thickness of the skin, SpaRibs and stiffeners should not be reduced to the extent that stress constraint is violated. In this work MSC.NASTRAN Sol 101 is used to determine the stress distribution.

1.3.1.2: Buckling

Buckling is static instability of structure under compressive and shear forces. The top skin (especially during 2.5 g pull up maneuver) and the bottom skin (during -1.0 g push over maneuver)
of a transport aircraft wing can be subjected is very high compressive stress and is the highly susceptible to buckling.

For complex structures such as an aircraft wing, buckling may occur locally, in some of the components instead of globally. Local buckling is not as catastrophic as global buckling but it too should be avoided because they distort the skin and thus significantly affect the aerodynamic pressure profile and hence the performance of the aircraft wing. In this work, MSC.NASTRAN Sol 105 is used for buckling analysis. If the Buckling Factor is greater than 1 then the structure is safe.

In a design problem, local buckling needed to be avoided, in other words buckling constraints needed to be satisfied both in local and global level. Figure 1-1 shows examples of global buckling of a wing under aerodynamic loads. Figure 1-2 shows example of local buckling. It is obvious that if this kind of local buckling occurs the wing will not collapse. In this work, the structure is considered safe only if Buckling Factor (BF) >1 which will be reported by MSC.NASTRAN only if buckling is satisfied in both local and global level.

Figure 1-1: Upper Skin of Wing Showing Global Buckling
1.3.1.3 : Flutter

Flutter is a dynamic instability of an elastic structure when immersed in a fluid in motion. It is characterized by the oscillations of a structure of increasing amplitude and which can eventually lead to a structural failure. The velocity of the structure relative to the fluid at which the onset of flutter occurs is known as flutter velocity and while designing an aircraft it is very important to know the Equivalent Air Speed (EAS) at which the onset of flutter occurs. Computing flutter velocity is an eigenvalue problem and the theoretical background can be found in the work of Mallik et al. [51]. In this work MSC.NASTRAN Sol 145 is used for the flutter analysis of the wing. MSC.NASTRAN implements several flutter analysis methods, such as the K-method, the PK-method and the PKNL method [52]. The K-method computes eigenvalues for user specified reduced frequencies whereas the PK-method does so for user specified velocities. The PKNL method is essentially PK-method without looping (PK Non-Loop) on all combinations of velocity, density and Mach number. The equation is solved only for specific points and thus is computationally inexpensive compared to PK method.

In this work, the PKNL method is selected in MSC.NASTRAN Sol 145 for flutter analysis of the wing. The range of the altitude is 0-35000ft, the Mach number 0.85 and the first 20 vibration modes are used in the flutter analysis. The wing is designed at cruising altitude of 35000 ft. and Mach number 0.85, where the Equivalent Air Speed (EAS) is 273 Knots. Considering 1.2 safety factor, the minimum allowable EAS is 328 knots.

Figure 1-2: Upper Skin of Wing Showing Local Buckling
The flutter velocity is the determined by a function written in Python from the V-g diagram (Velocity-damping diagram) as the Equivalent Air Speed (EAS) at which a flutter mode crosses the 0-damping line and become positive and increases monotonically. The point is marked with a red star in the V-g diagram in the example shown in Figure 1-3.

A reduction in weight of the wing means less fuel consumption and the increase in flutter velocity ensures safe cruising at a higher Mach number at a given altitude. Thus, it is desired to simultaneously minimize the weight and maximize the flutter velocity of an aircraft wing while preventing it to fail in strength and stability. However, a design with minimum weight does not ensure maximum flutter velocity and vice versa i.e. there is a trade-off between the two [41]. Therefore, both weight and flutter velocity are considered as objectives during the optimization in order to quantify this trade-off.

Figure 1-3: Velocity-damping (V-g) and Velocity-frequency (V-n) Diagram

1.3.1.4 : Other Constraints

Several other constraints might need to be considered in the design of an aircraft wing. Some other important constraints for a designing high aspect ratio subsonic aircraft wing as listed by Liu et al. [53] are maximum tip displacement, minimum first natural frequency and maximum twist angle of the wingtip.
1.3.2 Load Conditions using Static Aero-elastic Analysis

One of the most important steps in the aircraft design problem is to determine the aerodynamic loads and to accurately apply those forces to the structure. The lift distribution is dependent on airspeed, air density and outer shape of the airfoil. By methods like Computational Fluid Dynamics (CFD), the aerodynamic forces can be determined with great accuracy. Keye et al. [54] used the Reynolds-averaged Navier-Stokes CFD solver in aero-structural optimization of the CRM Wing. However, in MDO problems when a large number of design variables needed to be considered, this approach becomes computationally very expensive. In this work, the flight loads and the deformation profile of the wing-structure are obtained using MSC.NASTRAN Sol 144. In the sub-sonic regime MSC.NASTRAN uses the Doublet Lattice Method [55] [56] for computing the aerodynamic loads. This method is based on the linearized aerodynamic potential theory where both the thickness of the wing and the viscous effects are neglected. The lifting surfaces are represented by panels. In MSC.NASTRAN Sol 144, the aero-elements are quadrilaterals which are regularly arranged with sides parallel to airflow. The aerodynamic pressure is computed at a discrete set of points within these aero-elements. The structural grids are connected to the aerodynamic grids by surface splining method. The structural grids are the nodes corresponding to SpaRibs caps i.e. the common nodes between the upper skin and the SpaRibs. The flight loads are computed by static aero elastic trim analysis with given angle of attack.

1.4 Structural Optimization Algorithms

The goal in every optimization problem is to search through the design space (often multi-dimensional) and to reach the minima or maxima of the objective function. Maxima/minima of a function can be local and global. In complex/discontinuous/higher-dimension design space it often becomes impossible to decide whether an optimum is global or local and in optimization problems involving such design spaces this always becomes a challenge even if solution is obtained.

Optimization algorithms are of two types: gradient-based and non-gradient based. The gradient based optimization works with information on the gradient of the objective function. Considering the Karush–Kuhn–Tucker (KKT) necessary conditions for optimized designs [57],
the recursive relations used for gradient-based optimization are derived. For a nonlinear optimization problem with objective function and constraints twice-differentiable, Sequential Quadratic Programming (SQP) was developed. Sometimes the calculation of gradient at each iteration becomes computationally expensive and approximation methods are used. Such approximation methods were constructed by Schmit and Chang [58] and applied in structural optimization problems. Also, Canfield [59] developed the Rayleigh Quotient approximation which improve the accuracy of eigenvalue approximations.

One of the limitations of the gradient-based optimization is that the gradient needs to be determined throughout the design space. This means that for design spaces involving discontinuity and discrete variables, this method cannot be used. Also, for complex design spaces, under the assumption that multiple local minima might exist, it is important to provide an appropriate starting point for the search if one intends to find the global optima. As the convergence criteria is based on the change in gradient in successive iteration in the recursive process, there is a chance that the search might end in a local minimum.

The non-gradient based methods do not work with information but, for the most part, rather use nature-inspired evolutionary methods to reach the optimum. Two popular non-gradient based algorithm are Genetic Algorithm (GA) [60] and Particle Swarm Algorithm (PSO) [61] [62], and both have been used extensively in the optimization problems in the aerospace industry.

1.4.1 Mathematical Statement of Optimization

The general mathematical statement for optimization problem can be stated as:

\[ \min \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_r(\mathbf{x})] \]

Subjected to: \( g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \ldots, p \)

\( h_j(\mathbf{x}) = 0 \quad j = 1, 2, \ldots, q \)

where, \( \mathbf{x} = [x_1, x_2, \ldots, x_n]^T \) is the design variable vector

\( (\mathbf{x} \in \mathbb{R} \text{ and } x_i \leq x_{il} \leq x_{iu} \quad i \in \{1, 2, \ldots, n\}) \)
\[ f_k : \mathbb{R}^n \rightarrow \mathbb{R}, \ k = 1, 2, \ldots, r \] are the objective functions,

\[ g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}, \ i = 1, 2, \ldots, p; \ j = 1, 2, \ldots, q \] are the constraint functions

If the objective vector \( F(x) \) contains only one element i.e. \( F(x) = [f(x)] \), the optimization problem is known as single-objective optimization problem whose solution is a unique value, \( x_{\text{opt}} \).

In multi-objective optimization, a feasible solution, which simultaneously minimizes all functions in the objective vector usually does not exist. Hence, the pareto-optimal solutions are usually reported. Pareto-optimal solution are those that cannot be improved in any of the objectives without degrading one or more of the other objectives functions.

Mathematically, for \( x_i, x_j \in \mathbb{R} \), \( x_i \) is said to dominate \( x_j \) if,

\[
\begin{align*}
& f_k(x_i) > f_k(x_j) \quad \text{for } k = 1, 2, \ldots, r \\
& f_l(x_i) > f_k(x_j) \quad \text{for at least one index } k \in \{1, 2, \ldots, r\}
\end{align*}
\]

A solution \( x^* \) is known as pareto-optimal if it is not dominated by other solutions and the set of pareto-optimum solution is known as the pareto-front.

### 1.4.2 Particle Swarm Optimization

For non-gradient based optimization method, Particle Swarm Optimization is used in this work to optimize the number and the shape design variables of the SpaRibs and the algorithm for single objective optimization described briefly in this section. In every iteration, random particles (points in the design space) are distributed and evaluated. The positions of the particles are updated using individual and social corrections as shown in Figure 1-4. Social corrections are obtained based on the direction in which the entire swarm of particles is moving whereas an individual correction for a given particle is obtained from the direction in which that particle is going. Particle’s direction and position during the optimization process are updated (after the \( k^{\text{th}} \) iteration) using Equation 1-3 and Equation 1-4.
Where $x_k^i$ are the design variables and are called the positions of the particles, $v_k^i$ is the velocity of the particle which is used to update the position; $r_1$ and $r_2$ are the uniform random numbers between 0 and 1; $c_1$ and $c_2$ are known as thrust parameters; $w$ is the inertia weighting parameter of velocity; $p^i$ and $p^g_k$ are the best particle position (throughout iterative history) and the best swarm position, respectively. The inertia weight parameter, $w$, decides the influence of the particle’s velocity compared to the personal and social influences and it decides the optimization convergence rate. The parameter $t$ is called time step and is often taken as 1. The values of the parameters as considered in this work are listed in Table 1-1.

<table>
<thead>
<tr>
<th>PSO Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w$</td>
<td>0.78</td>
</tr>
<tr>
<td>$c_1$</td>
<td>2</td>
</tr>
<tr>
<td>$c_2$</td>
<td>2</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>1</td>
</tr>
</tbody>
</table>

Convergence is said to have been achieved when the difference in the objective value for the particles in the swarm falls within a specified limit or the maximum allowable number of iterations is reached. It is always recommended that for Particle Swarm Optimization if the convergence rate
is too high there is higher chance that the search will end in a local optimum. Thus, it is always recommended not to use a value too high for $w$.

*The pseudocode for PSO Algorithm is given below:*

```plaintext
For each particle of swarm
 Initialize particle position and velocity (randomly in Design Space)
 End For
Do until maximum iterations or minimum error criteria
 For each particle of swarm
 Evaluate Objective value
 If the Objective value is better than pBest
 Set pBest = current Objective value
 End If
 If current Objective value better than gBest
 Set gBest = current Objective value
 End If
End For
For each particle of swarm
 Update velocity (using Equation 1-3)
 Update position using Equation 1-4
End For
```

1.4.3 *The Weighted Sum Method*

The PSO Algorithm described above works only for the single-objective optimization problems. However, multi-objective problems can be converted to a single-objective function by constructing a function as a combination of the objective functions and optimizing that function. This method is known as the weighted sum method. Usually the function optimized is taken as a linear combination of the objective function (or their reciprocals). The final solution will definitely depend on how this master objective function is constructed. For example, if one wants to reduce weight, increase the flutter velocity and reduce the total drag of an aircraft, the master objective function (that is minimized) can be constructed as:
Shuvodeep De

\[ \text{Obj} = \text{Aw} + B \frac{1}{V_{\text{flutter}}} + CF_{\text{Drag}} \]

where, \( w, V_{\text{flutter}} \) and \( F_{\text{Drag}} \) are weight, flutter velocity and total drag force respectively.

1.4.4 Solving Multi-Objective Problems using PSO

As part of the work and for future research, The Experience Repository based Multi-Objective Particle Swarm Optimization (MOPSO) [63] algorithm is coded in the Python language.

The pseudocode for MOPSO Algorithm is given below:

```
For each particle of swarm
    Initialize particle position and velocity (randomly in Design Space)
End For

Initialized external archive (initially empty)

Quality (leader)

Do until maximum iterations or minimum error criteria
    For each particle of the swarm
        Select a particle (leader) from external archive
        Evaluate the objective function
        Mutation on positon and re-evaluate objective function
        If objective function after position mutation better than original
            Update objective value with value after mutation
        End If
        Select random number (rand) between 0 and 1
        If rand < 0.5
            Update objective value with value after mutation
        End If
        If the value of the objective is the better than the pbest
            Current value of the objective function is set as the new pbest
        End If
        Update the velocity of the particle
        Update the position of the particle
    End For
```

Update leader in external archive
Quality (leader)
Report the results of external archive

The code is verified with the MOPSO code written by S. Mostapha Kalami Heris (part of Yarpiz team) which is available online by solving the following benchmark example:

\[ \text{Minimize}, F(x) = [f_1, f_2] \]

where, \( f_1 = x(1) \)

\[ f = g(1 - \sqrt{\frac{x(1)}{g}}) \]

\[ g = 1 + \frac{9}{n - 1} \sum_{2}^{n} x(n) \]

Here given the list of design variables \( x \) is a vector of \( n \) real numbers each of which has a limit \([0, 1]\).

The analysis is run with 200 particles and the repository size is chosen to be 100. The maximum number of iteration is 200. The value of \( n \) is chosen to be 5 and each variable has a limit \([0,1]\).

![Figure 1-5: MOPSO Pareto-front from (a) MATLAB code developed by Yarpiz Team (b) Python Script developed in VT](image)
Figure 1-5 shows that the pareto-front after the 200th iteration as computed by the two codes looks similar. Using a computer with 3.2 GHz clock speed and 32 GB RAM, the Yarpiz MATLAB code took 60.8 s to reach the 200th iteration whereas the Python code as developed took 27.85 s on the same computer.

In MDO problems, evaluating the objective is often very expensive. MOPSO algorithms usually require quite a large number of iterations before the pareto front is traced accurately. Thus, a very high computational resource is required to solve such a problem using MOPSO. Figure 1-6 shows the iteration history of the optimization process with 20 particles. As evident from the results, the algorithm couldn’t trace the pareto-front until the 60th iteration.

Figure 1-6: Iteration History for the given Objective
1.5 Research Objectives

The main purpose of this research is to develop a generalized framework for the multidisciplinary multi-objective structural optimization of the internal structure of aircraft wing. The original EBF3GLWingOpt framework developed by Liu et al. was determined to be not robust enough as it works only for very simple SpaRibs parameterization for the NASA CRM Wing. The research objectives can be summarized as:

- Identify limitations of the original EBF3GLWingOpt framework.
- Develop new parameterization methods to increase the variety of SpaRibs profile that can be created
- Develop an algorithm to extract local panels from a structure this is independent of the geometry of the structure hence can be used for different structures
- Develop a method to create stiffeners for the local panels which is independent of shape of the panels
- Ensure highly skewed elements removed before the FEM of structure used for analysis
- Develop algorithm to optimize thickness of local panels that can be used as substitute for the gradient based optimizer, MSC.NASTRAN Sol 200 or the EBF3PanelOpt (for which efficiency dependent on available number of MSC licenses)

The implementation of the algorithms is demonstrated using the NASA CRM Wing but it can be implemented for other structures as well
The development of the original EBF3WingOpt framework has been done in collaboration with Dr. Qiang Liu and Dr. Mohamed Jrad since 2013. As mentioned before, the key contributions of this dissertation are to identify the limitations of the original EBF3WingIOpt framework and to develop algorithms for making the framework more robust and flexible. The details related to the optimization framework can be found in Dr. Liu’s dissertation and the methods to reduce computation time using parallel processing can be found in Dr. Jrad’s dissertation. In this chapter, an overview of the framework is given for completeness. The aircraft wing is described by size and shape design variables. The objective of multidisciplinary optimization is to minimize the structural weight of the wing while satisfying a number of constraints (including stress, buckling and flutter).

The parametrization of the SpaRibs using limited design variables, aero-elastic analysis, buckling analysis, and topology/sizing optimization of the SpaRibs for the global wing are all integrated into a framework, known as EBF3WingOpt. A framework EBF3PanelOpt is developed for the optimization of local panels that consider buckling and stress constraints. A global-local multidisciplinary optimization framework, known as EBF3GLWingOpt, has been developed to incorporate EBF3WingOpt and EBF3PanelOpt for the optimization of an aircraft wing. The commercial software MSC.PATRAN is used to generate geometry and the finite element model.
of the wing structure, while MSC.NASTRAN is used for static, aeroelastic, buckling, and dynamic analysis.

2.1 *SpaRibs* Parameterization

One can define a set of parameters for specifying the topology of each of the ribs and the spars. Although this makes the topology definition very flexible, the main disadvantages are that, the number of design variables becomes very large which leads to high computation cost for the optimization process. Also, the number of spars and the number of ribs are needed to be kept as fixed. However, by the linked-shape method [36] developed at Virginia Tech, *SpaRibs* system can be defined using just a few number of parameters. A maximum of six parameters are needed to define a set of curvilinear spars/ribs in each of the wing-boxes and their shape are not independent of each other. As shown in Figure 2-1 (Locatelli *et al.* [36]), the curves defining shape of the *SpaRibs* are first created in the normalized space and then transformed to the physical space using the MSC.PATRAN geometry generation module. Each of the curves is created using third order B-spline. The control points of the set of B-splines in the wing-box lines lies on the same line segment (known as the control-line). As listed in Table 2-1, number of Spars/Ribs is specified by design variable $P_1$ while the variables $P_2$ and $P_3$ define position of the control point line. The starting and ending points of the control-line are on two specified space edges in the wing-box. Spacing between starting points, control points and ending points are considered to be in geometric series, respectively. The three variables: $r_1$, $r_2$ and $r_3$ are the ratios of the first spacing and last spacing between the adjacent points.

![Figure 2-1: Linked-shape Method](image)
Table 2-1: Linked-shape Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td>Number of SpaRibs</td>
</tr>
<tr>
<td>$P_2$</td>
<td>$\eta_1$</td>
</tr>
<tr>
<td>$P_3$</td>
<td>$\eta_2$</td>
</tr>
<tr>
<td>$r_1$</td>
<td>$\xi_{11}/\xi_{12}$</td>
</tr>
<tr>
<td>$r_2$</td>
<td>$\xi_{21}/\xi_{22}$</td>
</tr>
<tr>
<td>$r_3$</td>
<td>$\xi_{31}/\xi_{32}$</td>
</tr>
</tbody>
</table>

### 2.2 Demonstration using CRM Wing

The capabilities of the optimization framework have been demonstrated by performing optimization for a subsonic fixed wing called NASA common research model (CRM). CRM is a fixed cantilever wing with a full span of 2,313 inch, aspect ratio of $AR = 9.0$, and a taper-ratio of 0.275. The wing is composed of two wing sections: the inner wing and the outer wing, which are connected at the break at about 37% semi-span. The front spar and rear spar are at 9% chord and 70% chord respectively. The outer mold line (OML) if the CRM wing is shown in Figure 2-2. Such a cantilever wing has been used by Stanford et al. to study aerodynamic tailoring [45] The SpaRibs are created between the front and rear spars.

![Outer Mold Line (OML) of the NASA CRM Wing](image)

**Figure 2-2: Outer Mold Line (OML) of the NASA CRM Wing**
In the current optimization problem, the topology and thicknesses of the SpaRibs and the thicknesses of the panels on the upper and the lower skin are optimized. However, the outer mold line (OML) is not changed. As a result, for each of the flight conditions the lift total lift do not change during the optimization process. The internal structure of the CRM wing i.e. the SpaRibs topology governed by the linked shape method (Figure 2-1) using 24 parameters. The line on which starting point of the B-splines are extracted and on which the end points created gets mapped into the leading-edge spar and trailing-edge spar respectively.

The description of those variables is given in Table 2-2. If the front and the rear spar considered fixed, then the first two parameters will need to be fixed and up to 22 parameters can be considered as design parameters in the optimization problem. The CRM wing is modeled using aluminum alloy 2024-T3 and its material properties are given in Table 2-3.

<table>
<thead>
<tr>
<th>Table 2-2: Design Variable Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dvar1</strong></td>
</tr>
<tr>
<td><strong>Dvar2</strong></td>
</tr>
<tr>
<td><strong>Dvar3 (p₁)</strong></td>
</tr>
<tr>
<td><strong>Dvar4 (p₂)</strong></td>
</tr>
<tr>
<td><strong>Dvar5 (p₃)</strong></td>
</tr>
<tr>
<td><strong>Dvar6 (p₄)</strong></td>
</tr>
<tr>
<td><strong>Dvar7 (p₅)</strong></td>
</tr>
<tr>
<td><strong>Dvar8 (p₆)</strong></td>
</tr>
<tr>
<td><strong>Dvar9 (p₁)</strong></td>
</tr>
<tr>
<td><strong>Dvar10 (p₂)</strong></td>
</tr>
</tbody>
</table>
In the optimization problem, the structural deformations are calculated using MSC.NASTRAN Solution 144. MSC.NASTRAN provides the doublet lattice method for the computation of the aerodynamic loads at subsonic regime. Two flight conditions are considered in
Shuvodeep De

this research: an angle of attack of -2 degrees and 6 degrees. The aerodynamic model of wing is created using MSC.PATRAN and it comprises of three aerodynamic lifting surfaces and 500 aerodynamic boxes. The spline nodes are the common nodes between the top wing skin and the SpaRibs. In order to avoid involving too many nodes, a minimum distance of 20 inch between adjacent spline nodes is specified. The flutter velocity is determined using MSC.NASTRAN Solution 145 where the PKNL method is used. An example of the SpaRibs profile created by the Linked-shape method, the aerodynamic mesh and the spline nodes are shown in Figure 2-3 (Liu et al.). It should be mentioned that the curvilinear ribs need to start at the trailing-edge spar and end in the leading-edge spar. Similarly, the curvilinear spars need to run from wing-root to the wing-tip. It is not possible to create any other SpaRibs configuration using the linked-shape method. A new method of parameterization to overcome this limitation has been developed and it will be described in details the next chapter.

Figure 2-3: (a) SpaRibs Profile (b) Aerodynamic Mesh (c) Spline nodes on Upper Skin
2.3 Boeing HSCT N+2 Rear Wingbox Construction by Linked-shape Method

Locatelli et al. [29] used the link-shaped method to parameterize the internal geometry of the Boeing HSCT N+2 Wing. As shown in Figure 2-4 (Locatelli et al. [29]), this complex wing comprises of 14 wing boxes. SpaRibs are constructed in quadrilateral or triangular wing boxes and C₀ continuity is enforced between them at each cell boundaries.

During the NASA SBIR/STTR program under the contract NNX14CD16P the construction of SpaRibs and Global-local Optimization of skin panels for the rear wing-box (numbered 10 in Figure 2-4) was demonstrated using the Linked-shape Method. The top skin and bottom skin of the wing-box and the end points of the SpaRibs (on the outer boundary of the skin) was provided. There are 7 SpaRibs in the span-wise direction and 8 SpaRibs in the chord-wise direction. Only the control points of each of the SpaRibs can be considered as the design variables here. Figure 2-5 shows the geometry and the mesh as constructed in MSC.PATRAN.
For a more detailed design of the wing structure, it is necessary to size each panel of the skin independently and eventually add stiffeners to improve the buckling performance. Moreover, each of the SpaRibs may need to be sized separately and the holes and cut outs may be required to decrease the weight of the internal structure to allow the passage of hydraulic and electrical lines. These types of design features can be implemented by extracting the local panels or ribs and spars sections from the global finite element model, create a more refined mesh of the component and optimize it independently from the rest of the structure.
In the original EBF3GLWingOpt, the edges of the local panel edges are determined by finding the intersections of the spars, ribs, and wing skins. Interior panel nodes are then sorted out using the edge information by the in-out algorithm described in the next section. The panel mesh is used to build a surface using MSC.PATRAN. The stiffeners are constructed as evenly spaced surfaces normal to the panel surface and are ‘associated’ with the panels (which means after the panel and the stiffener will share common nodes after the mesh is generated. The stiffened panel as created is then refined using fine mesh, as shown in Figure 2-6. This process of construction of the panel’s surface is performed only in the first iteration of the Global-Local process. After that, the panel model is updated based on its design variables. MSC.NASTRAN Sol 144 is run for the global wing model and displacement boundary condition are imposed on the outer boundary of the panels by linear interpolation from the global deformation field.

The maximum von Mises stress \( (\sigma_{vm})_{max} \) and the first buckling eigenvalue \( \lambda_p \) are computed using MSC.NASTRAN Sol 101 and MSC.NASTRAN Sol 105 respectively. During the local panel optimization process, if \( t_0 \) is the initial panel thickness, then the new panel thickness is computed as follows,

\[
\begin{align*}
    t_{opt1} &= t_0 \frac{(\sigma_{vm})_{max}}{\sigma_y} \\
    t_{opt2} &= t_0 \left( \frac{1}{\lambda_p} \right)^{\frac{1}{2}} \\
    t_{opt} &= \max(t_{opt1}, t_{opt2})
\end{align*}
\]

Where \( (\sigma_{vm})_{max} \) is the maximum von Mises stress, \( \sigma_y \) is the yield stress of the material, and \( \lambda_p \) is the buckling eigenvalue. The optimal thickness of the panel \( t_{opt} \) is the maximum of the optimal
thicknesses calculated by strength and buckling constraints. The Global-Local process is illustrated in Figure 2-7. In each iteration, the global wing model is constructed by assembling the optimized local panels and new global deformation field is computed by running MSC.NASTRAN Sol 144. From the deformation field boundary conditions for the panels are computed for the next iteration. After every iteration, the percentage difference between weight of new design and the design from previous iteration is calculated. Convergence is said to have been achieved when the difference falls within a certain user-specified limit.

**Figure 2-7: Global-Local Process in EBF3PanelOpt (Qiang Liu [64])**

Figure 2-7 shows the optimization history of wing with a certain SpaRibs configuration by EBF3PanelOpt. The optimization converged after the 4th iteration.

**Figure 2-8: (a) SpaRibs Profile (b) Iterative History in EBF3PanelOpt**
Although, in this global-local optimization framework, the satisfaction of buckling and stress constraints is ensured for the local panel, it does not automatically lead to satisfaction of stress and buckling constraints for the entire wing. It was found that even though weight convergence can be achieved, the global buckling constraint is often not satisfied. This is one of the major drawbacks of the original EBF3GLWingOpt framework.

### 2.5 Local Panel Generation by the In-out Algorithm

In the local panel generation, the first step is to determine the outer edge of the panels using the intersection of the curvilinear spars and ribs with the upper skin and the lower skin. The mesh from the global model is considered and comparisons of coordinates are made in order to extract the nodes of each panel edge. This list of nodes is then sorted using information from the connectivity matrix, such that successive nodes in the list are adjacent to each other in the mesh. This applied method is applicable to arbitrary shaped curvilinear spars, ribs that form 4-edged panels. The next step consists of finding all nodes that lie inside the panel surface. This is solved as a Point in Polygon Problem [65]. The edges of the panels are given as a polygon using the sorted nodes from the previous step, and all nodes that are inside that panel are extracted using the Python function “path” from the “matplotlib” library (which essentially solves the Point in Polygon Problem). This function works only with 2D coordinates and not 3D. As a solution, XY, YZ, and XZ projections can be used if panel surface is straight or slightly curvilinear and lies almost in one of these planes. For highly curvilinear SpaRibs, the projections become non-representative of the panel surface.

This limitation has been overcome using the following process shown in Figure 2-9 in the following page.
Select Adjacent Sparibs

Define a 2D plane using three Corner Points of the Panel

Define a 2D Orthonormal Base vector (u,v) in the Panel

Project all Nodes on the Plane.
(Original Nodes are in Black and Projected Nodes are in Red)

Apply “mathplotlib.path” function to extract Panel Nodes
(the resulted set is, \{1,2,3\}, in the Picture)

Ignore Nodes that are not connected to the Panel Edges
(Node Number 3 in the Picture)

Figure 2-9: Extraction Method
Shuvodeep De

It should be noted that in the case of a panel bounded by SpaRibs of very high curvature, some nodes that lie outside the panel might have a projection inside the panel surface. They are picked up using the connectivity matrix (they are not connected to the edge nodes) and are discarded. (*Modification for highly curvilinear SpaRibs made by Dr. Jrad*)

This method of creating local panel worked well in the original EBF3GLWingOpt but it has some limitations which causes it to fail in many other SpaRibs configuration (to be discussed later). Firstly, the boundary of the local panels is determined by the intersection of two adjacent curvilinear ribs and two adjacent curvilinear spars on the skin. Thus, it is **limited to creation of only regular four-edged panels with two of the edges along adjacent curvilinear spars**. It cannot create local panels for SpaRibs configuration shown in Figure 2-10 (b) where some of the SpaRibs are ending at the wing-root.

Secondly, the boundary nodes are sorted by comparison of coordinates with respect to the wing-root i.e. the process is limited to the NASA CRM Wing. A generalized algorithm, independent of coordinates to extract local panels of any size and shape has been developed as part of this work and it will be described in details in the eighth chapter.

![Figure 2-10: SpaRibs Configuration. (a) Local Panel Algorithm works (b) Does not work](image-url)
2.6 Creation of Stiffeners

Stiffeners having a rectangular cross section are attached to the panels along the span-wise direction to resist the in-plane compression within the CRM wing skins. The buckling factor of the panel increases by adding more stiffeners. However, the stiffened panel weight is summation of weight of the plate and the weight of stiffeners. As found in the study of Liu [64], adding stiffeners beyond a certain number may increase the weight of the stiffened panel.

The shape of the stiffeners is defined using the linked-shape method developed by Locatelli et al. [12] in a normalized space and their geometry is constructed using third-order B-splines and projected into the physical space, similarly to the SpaRibs. The starting point and the ending point of the stiffeners lie on the edges of the local panel which are the one closer to the wing-root and the one closer to the wing-tip, respectively. This method is restricted to local-panels having only four edges but the advantage is that it reduces the number of design variables. Figure 2-11 shows a rectangular local panel with stiffeners where the control point lies on the dotted straight line.

![Figure 2-11: Stiffeners Shape Parameterization using Linked-shape method](image)

### Table 2-4: Design Variables for Stiffeners on the Local Panel

<table>
<thead>
<tr>
<th>DVp1</th>
<th>Stiffener Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVp2</td>
<td>Stiffener Thickness</td>
</tr>
<tr>
<td>DVp3</td>
<td>Start Point Ratio</td>
</tr>
<tr>
<td>DVp4</td>
<td>Control Point Ratio</td>
</tr>
<tr>
<td>DVp5</td>
<td>End Point Ratio</td>
</tr>
</tbody>
</table>
As the topology optimization of each of the local panels would be computationally very expensive, straight uniformly spaced stiffeners i.e. the value of the design variables DVp3, DVp4 and DVp5 are fixed as 1.

2.7 Integration of EBF3WingOpt and EBF3PanelOpt

As described in Section 2.4, the module EBF3PanelOpt optimizes the thickness of the local panels and recreate the global wing by assembling the optimized panels. However, the parameters describing the shape of the SpaRibs are not optimized by this module. In order to find the best combination of SpaRibs number, shape, and panel thickness, a multilevel optimization framework is developed by integrating the EBF3panelOpt framework with the Particle Swarm Optimization (PSO) Algorithm. In this framework, the shape design variables and the number of the SpaRibs in the outer and the inner wing-box are optimized by Particle Swarm Optimization (PSO). The framework to optimize the number and shape design variables of the SpaRibs is called EBF3WingOpt. For each particle of EBF3WingOpt, the local panels are optimized using EBF3PanelOpt. This integrated framework has been termed as EBF3GLWingOpt.

2.8 Parallel Computation

In order to reduce computational cost, a parallel computing framework has been developed mainly by Dr. Jrad for the original EBF3GLWingOpt. The parallel computational capabilities are implemented in both PSO level as well as in the optimization of the local panels. A user using EBF3GLWingOpt framework is limited by the memory that can be used for computation as well as the available number of MSC licenses.

The license cycle-check method and the memory self-adjustment method have been employed to overcome the limitations of the number of MSC licenses and memory saturation, respectively. A detailed description of these two methods can be found in [35]. The main idea of the two methods consists of creating a communication language between the running processing and a set of signs that they use to share information about the current status of the simulation in order to control themselves and use the available resources efficiently, even if these available
resources are not sufficient. The challenge in these two methods is that they all control themselves collectively, unlike the traditional way of thinking where the main process controls the rest of the processes. The capacity of the machine is computed using a self-learning procedure where the running processes collect information about the amount of memory they have used for each job in the whole simulation. This dynamically growing database gives, therefore, an accurate estimate of the amount of memory that these new jobs require. Therefore, the simulation runs smoothly and MSC.NASTRAN only allocates the amount of memory that it really needs. Furthermore, the estimated capacity of the machine is more accurate and the memory saturation is avoided. This Global-Local parallel computational framework ran on Virginia Tech’s Advanced Research Computing System “NewRiver” which has 134 nodes. Each node has 24 core @CPU Speed 2.50 GHz and 128 Gb of RAM. While the simulation time depends heavily on the wing design variables and input parameters (parallel processes, mesh size, etc.), an example of a wing with 4 inner ribs, 25 outer ribs, 18 longitudinal stiffeners and a mesh of 160k elements and 100k nodes converges in about 1 hour and 41 minutes using 60 processes.

Figure 2-12: Simple Task Running 5 Jobs using License Cycle-check Method

Figure 2-13: Simple Task Running 5 Nastran Jobs Using Memory Self-adjustment Method
2.9 Chapter Summary

This chapter gives an overview of the EBF3GLWingOpt framework that has been developed in collaboration with Dr. Qiang Liu and Dr. Mohamed Jrad. EBF3GLWingOpt is a multidisciplinary multiscale structural optimization framework for the NASA Common Research Model (CRM) with curvilinear spars and ribs (known as SpaRibs) and stiffeners.

The optimization framework consists of two sub-systems: EBF3WingOpt optimizes the SpaRibs shape using Particle Swarm Optimization (PSO) Algorithm and the local optimization which optimizes the thickness of local panels extracted from the global model considering static and buckling constraints. The software written in Python environment incorporates MSC.PATRAN for generating the geometry and the mesh and MSC.NASTRAN for finite element analysis and has capabilities for parallel computation.

The SpaRibs and the stiffeners for the wing are parameterized using limited number of design variables by the Linked-shape method originally developed by Dr. Davide Locatelli. Linked-shape method parameterized of the rear wing-box of NASA HSBT N+2 Wing-box has also been demonstrated. The local panels are extracted from the CRM Wing using an algorithm which basically solves the “Point in Polygon Problem” to pick up the nodes inside the panel.

The optimization framework has certain limitations. They are:

- The curvilinear ribs need to start at the trailing-edge spar and end in the leading-edge spar. Similarly, the curvilinear spars need to run from wing-root to the wing-tip.
- The algorithm to extract local panel is dependent on the geometry of the NASA CRM Wing and is capable of extracting panels with only four edges. In other words, it cannot be used for any other applications.
- Satisfaction of stress and buckling constraints for local panels does not ensure satisfaction of

The main objective of this work was to identify and overcome the limitations of this framework and to make it more robust. In the next chapter, a new method for parameterization of
the geometry of *SpaRibs* is described. This method allows the creation of a much wider range of SpaRibs configurations than what was possible using the original EBF3GLWingOpt framework. In the fourth chapter an algorithm to divide the wing-skin into local panels with the help of *SpaRibs* intersection is described. The original EBF3GLWingOpt framework has the capability of creating local panels but it was limited only to creating four edged panels from the NASA CRM wing geometry. The new algorithm is completely based on set operations on the connectivity data of the finite element model and can be used to create panels of any number of edges from any geometry. In the fifth chapter the complete framework used for shape and size optimization is described along with examples of its application. The sixth and last chapter provides a summary of the work along with mention of possible future work.
3 Geometry Parameterization and Mesh Generation

3.1 Enhance SpaRibs Topology Optimization Tools

The original EBF3GLWingOpt framework provides the option of using independent design spaces for creating ribs in the inner wing box and the outer wing box, separately. However, this approach limits the number of configurations of the internal structure that can be analyzed as explained in the previous chapter. A new approach for creating the ribs continuously from the inner to the outer wing box was implemented as part of this effort.

The EBF3GLWingOpt framework is first used to generate the finite element model of the baseline wing. The model comprises of 3 spars and 37 ribs. Figure 3-1 provides a side by side comparison of the baseline model and a model with curvilinear SpaRibs. In the model with curvilinear SpaRibs, 10 ribs are placed in the inner wing-box and 25 in the outer wing-box (For the ribs $r_1$, $r_3=1.00$, $r_2=1.67$; for the middle spar $r_1$, $r_2$, $r_3=2.00$).
If the SpaRibs in the wing are created using a single design space, the rib-spacing in the inner and in the outer wing-box cannot be controlled independently. However, this new approach has an advantage over the previous one since it allows creation of new ribs configurations. The entire set of ribs can be defined using 6 design variables for curvilinear SpaRibs topology or 4 design variables for straight SpaRibs topology while preserving mesh continuity in all parts of the model. It should be mentioned that straight SpaRibs are nothing but SpaRibs with infinite curvature. Since the start point, end point and the control point should be collinear, the control point can be dropped.

The CRM wing has a slope discontinuity in at the trailing edge break and it was found that the method of creating SpaRibs as used in the original EBF3GLWingOpt framework fails to create a continuous surface for the SpaRibs as shown in Figure 3-2. This is because in MSC.PATRAN when a curve is projected on a surface that has a slope discontinuity, it creates two curves which are discontinuous where there is slope change. This section describes the method developed to overcome this limitation.
Discontinuous SpaRibs through the trailing-edge break

Figure 3-2: Discontinuous SpaRibs at Trailing-edge Break

3.1.1 Method of Lines

The dimensions for the current CRM Wing are as follows (Figure 3-3):

\[ L_1 = 379.41 \text{ in}, \quad L_2 = 911.16 \text{ in} \]
\[ L_3 = 850.22 \text{ in}, \quad L_4 = 328.58 \text{ in} \]

Figure 3-3: Control-line for Construction of Curvilinear SpaRibs.

In this method, the lines \( L_1L_2 \) and \( L_3L_4 \) are ‘chained’ which mean they were combined to form a single curve with slope discontinuity at the trailing edge break. Points are created on them at specified parametric positions, determined using Linked-shaped method.
We define:

\[
L_{fc} = \frac{L_1}{L_1 + L_2}, \quad L_{rc} = \frac{L_4}{L_3 + L_4}
\]

For straight SpaRibs topology only four configurations are possible as shown in Figure 3-4 depending on the location of the starting point on \(L_1L_2\) and ending point on \(L_4L_3\). If \(P_f\) and \(P_r\) are the parameters indicating the position of the start and end point respectively, the four configurations are defined as follows:

**Configuration 1**: \(P_f < L_{fc}\) and \(P_r < L_{rc}\)

**Configuration 2**: \(P_f < L_{fc}\) and \(P_r > L_{rc}\)

**Configuration 3**: \(P_f > L_{fc}\) and \(P_r < L_{rc}\)

**Configuration 4**: \(P_f < L_{fc}\) and \(P_r < L_{rc}\)

![Figure 3-4: The Four Possible Configurations of a Rib inside the Wing-box.](image)

For curvilinear SpaRibs, there are eight possible configurations of the ribs are possible depending upon the locations of the starting point, ending point and the point on the control line.

Figure 3-3 shows the control line used for the construction of the curvilinear ribs. It consists of the segments \(L_5\) and \(L_6\) chained together. In addition to \(L_{fc}\) and \(L_{rc}\), we define another parameter \(L_{cc}\) for the control-line:
Based upon the positions of $P_f$ and $P_r$, the eight possible configurations are shown in Figure 3-5. Unlike $L_{fc}$ and $L_{rc}$, the value of $L_{cc}$ is not fixed because the position of the control-line can change.

Configuration 1: $P_f < L_{fc}, P_r < L_{rc}, P_{cc} < L_{cc}$
Configuration 2: $P_f < L_{fc}, P_r > L_{rc}, P_{cc} > L_{cc}$
Configuration 3: $P_f < L_{fc}, P_r > L_{rc}, P_{cc} < L_{cc}$
Configuration 4: $P_f < L_{fc}, P_r < L_{rc}, P_{cc} > L_{cc}$
Configuration 5: $P_f > L_{fc}, P_r > L_{rc}, P_{cc} > L_{cc}$
Configuration 6: $P_f > L_{fc}, P_r < L_{rc}, P_{cc} > L_{cc}$
Configuration 7: $P_f > L_{fc}, P_r < L_{rc}, P_{cc} < L_{cc}$
Configuration 8: $P_f > L_{fc}, P_r > L_{rc}, P_{cc} < L_{cc}$

Figure 3-5: The Eight Possible Configuration of a Curvilinear Ribs in the CRM Wing.

The Blue Line in the middle represents the Control Line
In each of these cases except #4 and #8 two curves will be created due to projection. For Case #4 and Case #8 three curves will be created. The process of creating continuous curve is described using Case #4. First, a point is created where the two curves created in the inner wing-box due to the projection intersect the trailing edge break (P1 and P9). On the curve created in the outer wing box a set of closely spaced points are created (P2,P3…P8). After that a curve is fit through points P1, P2….P9. Once this curve is created, the previous curve in the outer wing-box is deleted. The three existing curves will have C0 continuity at the trailing-edge break and it can create surface for SpaRibs without any gap at the trailing-edge break.

Figure 3-6: Ensuring C0 continuity at Trailing-edge between Projection Curves

3.1.2 Extended Design Space Method

In this approach, the SpaRibs are defined by B-splines in a normalized design space and then transformed to the physical space, similar to the Linked-shape Method. The starting point, the ending point and the control point of each of these B-splines are constrained to move along certain curves shown by the dotted red lines in Figure 3-7 and Figure 3-8. Parts of the curve on which the starting and ending points of the B-splines located gets transformed to two non-intersecting edges of the wing (for example, the leading-edge spar and the trailing edge spar). As the curves extends outside the reference design space (aquamarine region in Figure 3-7 and Figure 3-8) by a certain parametric length (represented as Le1 and Le2), this method is called Extended Design Space Method. This approach allows the creation of SpaRibs profiles which are not possible to be created using the original version of the EBF3WingOpt code. The SpaRibs topology is governed by a set of design parameters known as the “Offset Parameters”. These offset
parameters define the part of the curves on which the starting, ending and control points for the B-splines are created.

In Figure 3-7, four Offset Parameters: $O_{11}$, $O_{12}$, $O_{21}$, and $O_{22}$ are shown and these can be used to define straight SpaRibs (technically SpaRibs with infinite curvature as mentioned before). Straight SpaRibs can be defined using line-segments and only start and end points need to be specified. The Offset parameters are used to extract two points on the curves at a specified parametric distance from either end of the curve. In this figure, the start and end points for SpaRibs are uniformly spaced between these two points. As mentioned, the aquamarine area represents the normalized design space for the wing and the violet line in between represents the trailing edge break. Only the parts of the line which lie inside the normalized design space are mapped to the physical space to create the rib geometry.

![Figure 3-7: Extended Space Method to create SpaRibs](image)

In the case of curvilinear SpaRibs, the curve for the control points need to be considered and a total of six offset parameters can be used as shown in Figure 3-8 as $O_{11}$, $O_{12}$, $O_{21}$, $O_{22}$, $O_{31}$, and $O_{32}$. The spacing between the points on each of the curves is considered uniform in this study, but they can be made non-uniform by introducing additional design variables. In this code, there is also the option for creating bifurcation of the trailing edge spar in the inner wing-box.

![Figure 3-8: Extended Space Method to create curvilinear SpaRibs](image)
The Extended Space method can be used to define various SpaRibs configurations for the CRM wing. The number and curvatures of the SpaRibs can be varied. In addition, the code also has the option for creating bifurcation of the trailing-edge spar in the inner wing-box.

### 3.1.3 Determining Point Locations on Curves

The determination of spacing between the points for the topmost curve in Figure 3-7 and Figure 3-8 is discussed in this section. It is determined similarly for the other two curves. The curves here are normalized, which means that the total length of each of them is 1. If $L_{\text{Total}}$ is the total length of the curve then from the figure,

\[
L_{\text{Total}} = L_{e1} + L_1 + L_2 + L_{e1}
\]

\[
\Rightarrow L_{e1} + L_1 + L_2 + L_{e1} = 1
\]

Length of the curve where the points will be created, $L_P$,

\[
L_P = L_{\text{Total}} - O_{11} - O_{12}
\]

\[
L_P = 1 - O_{11} - O_{12}
\]

3-1

Supposed, the points are in geometric progression with common ratio $R$ and length of the first interval $A$. Equation 3-1 can be written as:

\[
L_P = A(1 + R + R^2 + \cdots R^N)
\]

\[
1 - O_{11} - O_{12} = A(1 + R + R^2 + \cdots R^N)
\]

If the points are considered equally spaced, $R = 1$ and

\[
1 - O_{11} - O_{12} = AN
\]

\[
\Rightarrow A = \frac{1 - O_{11} - O_{12}}{N}
\]

3-2
Otherwise the ratio of the spacing between the first two points and the last two points may be considered as design variable i.e.

\[ DVs = \frac{AR^N}{A} \]

\[ \Rightarrow R = DVs^{\frac{1}{N}} \]

3.1.4 Examples of SpaRibs Topology

The Extended Space method was used to define various SpaRib configurations for the CRM wing.

3.1.4.1 Straight SpaRibs

For simplicity, only the chain L1L2 was extended (by 20% of original length on either side). Also, the offset parameters were set to different values to define different topologies. Figure 3-9 shows some of these topologies. Each of the models consists of 25 ribs.

![Figure 3-9: SpaRib Topology using the Extended Space Method](image)

3.1.4.2 Curvilinear SpaRibs

Depending on the parameters, a broad range of SpaRibs topologies with different number of SpaRibs and curvatures can be created. Figure 3-10 shows some of the possible configurations (37 ribs). Table 3-1 gives the corresponding values of the offset parameters.
Figure 3-10: Curvilinear *SpaRibs* Configurations with the Extended Space Method

<table>
<thead>
<tr>
<th>Model</th>
<th>Offset_1_1</th>
<th>Offset_1_2</th>
<th>Offset_2_1</th>
<th>Offset_2_2</th>
<th>Offset_3_1</th>
<th>Offset_3_2</th>
<th>Bif.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>0.000</td>
<td>0.025</td>
<td>0.050</td>
<td>0.050</td>
<td>0.000</td>
<td>0.025</td>
<td>yes</td>
</tr>
<tr>
<td>Model 2</td>
<td>0.050</td>
<td>0.000</td>
<td>0.000</td>
<td>0.030</td>
<td>0.050</td>
<td>0.025</td>
<td>yes</td>
</tr>
<tr>
<td>Model 3</td>
<td>0.050</td>
<td>0.050</td>
<td>0.050</td>
<td>0.030</td>
<td>0.075</td>
<td>0.000</td>
<td>yes</td>
</tr>
<tr>
<td>Model 4</td>
<td>0.025</td>
<td>0.050</td>
<td>0.050</td>
<td>0.050</td>
<td>0.025</td>
<td>0.050</td>
<td>yes</td>
</tr>
<tr>
<td>Model 5</td>
<td>0.1</td>
<td>0.1</td>
<td>0.025</td>
<td>0.025</td>
<td>0.1</td>
<td>0.1</td>
<td>no</td>
</tr>
<tr>
<td>Model 6</td>
<td>0.145</td>
<td>0.1</td>
<td>0.109</td>
<td>0.175</td>
<td>0.015</td>
<td>0.159</td>
<td>no</td>
</tr>
</tbody>
</table>

Figure 3-11 shows the mesh around the trailing edge break of Model 4. Clearly both C0 and C1 continuity has been satisfied between the part of the rib which lies inside the outer wing-box and
that which lies inside the inner wing-box. There is mesh continuity between all the curvilinear ribs and other structural components.

![Figure 3-11: Mesh-continuity at the Trailing-edge break](image)

### 3.1.5 Ribs Cut-out

The ribs of a commercial aircraft usually contain cutouts which first of all make the ribs lighter and secondly allow for the passage of fuel pipes and wires. The **EBF3GLWingOpt** software uses a simple process to govern the size and shape of the cut-outs on each of the rib local panels using as less as a single design variable. The steps executed in MSC.PATRAN are given below in the flowchart in Figure 3-12.

![Figure 3-12: Process of creating Cut-outs in MSC.PATRAN](image)
Shuvodeep De

The algorithm is first used to create holes in the rear wing-box of the Boeing N+2 HSCT wing-box. The value for \(d_1, d_2, d_3\) and \(d_4\) is as 0.2. Fig shows the resulting structure with holes.

![Figure 3-13: Boeing N+2 HSCT Wingbox with Oval cut-outs](image)

This wing-box is a simple regular structure where each panel of the *SpaRibs* are almost of the same size and hence this process works well for it. However, if the *SpaRibs* panel are too long this process often fails. This happens when the part of the closed B-spline that is projected onto the surface lies outside the surface. For such situations, an alternate method is developed with creates quadrilateral cut-outs instead of ovals but the projection of the curve always falls entirely inside the *SpaRibs* panel. The process is given in the flowchart in Figure 3-14.

![Figure 3-14: Process of creating Cut-outs using Surface Manifold in MSC.PATRAN](image)
Shuvodeep De

The cut-outs created by this method contain sharp edge and to get rid of them a filet must be added. The fillet radius varies directly with the area of the local panel. Figure 3-15 shows an example of the SpaRib system with cut-outs created using the above method. There are 14 ribs in inner wing-box and 29 in outer wing-box.

![Figure 3-15: SpaRibs with Cut-outs](image)

### 3.1.6 Creation of Stiffeners

Stiffeners are added on the upper and the lower skin to increase the rigidity and improve the buckling performance. The stiffener parameterization is independent of the SpaRibs parameterization.

A set of curves (termed as guiding curves) is created on the upper and the lower skin of the wing which run from the wing-root to the wing-tip using MSC.PATRAN. The points of intersection between these lines and the ribs are determined and a stiffener is created on the skin between every two consecutive points. The advantage of this method is that the profile of a stiffener can be represented using few number of parameters that define the shape of these curves. In Figure 3-16 the process of creating stiffeners, represented by a flowchart.

![Figure 3-16: Process of Creating Stiffeners](image)
Figure 3-17 show an example where 14 uniformly spaced guiding curves created in each of the two wing-boxes. The curves are B-splines with start, end and control points uniformly spaced.

**Figure 3-17: Stiffeners (black) between Spars**

### 3.1.7 Mesh Modification and Removal of Bad Elements

Meshing the wing geometry in MSC.PATRAN after adding the stiffeners might generate highly skewed triangular elements. As these elements can lead to highly inaccurate results, it is important that the mesh be modified and all such elements be removed. An algorithm was developed to find nodes which are located within a certain distance (user specified) from each other. For each set of these nodes, the coordinates of any one of them are modified to bring both the nodes to the same location. This causes the collapse of some elements which are subsequently deleted. Figure 3-18 shows an example of highly skewed triangular elements created by MSC.PATRAN that are removed by the algorithm, which resulted in an improved mesh.

**Figure 3-18: Removal of Highly Skewed Elements**
3.2 Geometry and Mesh Failure in PATRAN

It has been found that for some SpaRibs configurations, MSC.PATAN fails to create the entire geometry. Sometimes even though the entire geometry is created, it fails to create the mesh for all parts. In either case, the model cannot be used for analysis.

3.2.1 Possible Causes of Geometry Failure

Figure 3-19 shows a case where one of the ribs pass very near to the trailing edge break and MSC.PATAN fails to create the surface. These cases result in errors during the analysis and the design is automatically discarded.

![Figure 3-19: Geometry Failure in MSC.PATRAN](image)

3.2.2 Possible Causes of Mesh Failure

As mentioned before, MSC.PATAN fails to mesh the structure for some configurations and these designs need to be discarded. The simplest solution to avoid mesh-failure is to use a very refined mesh, but this inherently leads to increased computational costs. The most common cause of mesh failure is the association of stiffeners or part of SpaRibs which are much smaller than the specified maximum element size with the skins. Figure 3-20 shows some of the cases for which mesh failure occurs. In Case 1 a stiffener and a SpaRib are starting almost from the same point while in Case 2 part of SpaRib associated with the skin in the outer wing-box is too small compared to the element size.
The mesh failure due to association of the stiffeners can be avoided by skipping association and mesh of the stiffeners which are too small and which start very closely to the starting of a SpaRib at the wing-root.

It is assumed that the absence of a few small stiffeners would not have a significant effect on the final result but it will prevent mesh-failure for quite a large number of designs. Figure 3-22 shows the mesh in the region near the trailing edge break. The stiffeners are meshed with quad elements. It is clearly visible that stiffeners which are too small compared to the element size have not been meshed.
3.2.3 Checking for Mesh Failure

A function was developed to ensure if MSC.PATRAN successfully meshed all the surfaces of the geometry complete. Whenever a surface is meshed in MSC.PATRAN it writes a message in the ‘.ses_rec.01’ file: “N nodes and M elements created for Surface P”. If the total number of surfaces is not equal to the total number of such messages then it can be concluded that the mesh failed.

Figure 3-23: Mesh-verification Algorithm.
3.3 Chapter Summary

This chapter describes a new method to create SpaRibs for wings with multiple sections. The parameterization of SpaRibs in the original EBF3GLWingOpt was limited to creation of curvilinear ribs that start at the leading-edge spar and end at the trailing-edge spars and curvilinear spars that start at the wing-root and end at the wing-tip. With the new method of SpaRibs parameterization known as the ‘Extended Design-space Method’ it is possible to create a wider range of SpaRibs profile. A method to create lightening cut-outs on the SpaRibs using limited number of design variables is also described. Further, stiffeners are added to the upper and the lower skin of the wing to improve buckling resistance and the stiffener parameterization is independent of the SpaRibs parameterization. For certain configurations of SpaRibs and stiffeners, MSC.PATRAN fails to generate the mesh for the entire structure. A method to avoid mesh failure by avoiding creation of stiffeners of too short a length is also discussed. Once the mesh of the wing is generated, the next step is to divide it into local panels for global-local analysis. The algorithm to create local panel will be discussed in details in the next chapter.
4 Generalized Local Panel Extraction

As mentioned previously, the algorithm used in the original EBF3GLWingOpt framework to extract the local panels has certain limitations. It is dependent upon the coordinates of the nodes and can extract only panels with four edges from the NASA CRM Wing. It was necessary to develop a generalized code to break a 2D mesh into local panels. Such a code can be used for any SpaRibs configuration for the CRM wing as well as applied to other structures like the fuselage or vehicle frame with minor modifications. This algorithm is purely based on set operations on the element connectivity matrix (completely independent of the nodal coordinates). The only limitation is that it works only with triangular elements. This algorithm will be described in details in this chapter.

4.1 Mesh Continuity Algorithm

The basic concept is that in the finite element model of a structure, if there is no mesh discontinuity then given all boundary nodes on a closed curve on a surface and at least one interior node, it is possible to determine all interior nodes and elements using the connectivity matrix information following the steps given in Figure 8.2 (refer Figure 4-1).
Figure 4-1: Determining Panel Elements by Boundary Elements (red) and an Interior Node (black).

Find Elements except Boundary Elements which share given Interior Node and add to ELEMENT_LIST

Find Elements not in ELEMENT_LIST but shares at least one Node with Elements in ELEMENT_LIST

Initialize with Null Vectors ELEMENT_LIST and NODE_LIST

Find Boundary Nodes and all the Elements contained inside the Panel that share at least one of these Boundary Nodes (Elements marked in red in Figure 8 1)

Add Boundary Elements to ELEMENT_LIST
Add Boundary Nodes to NODE_LIST

Add Elements to ELEMENT_LIST and nodes to NODE_LIST

Find Elements not in ELEMENT_LIST but shares at least one Node with Elements in ELEMENT_LIST

No

ELEMENT_LIST updated?

Yes

Stop

Figure 4-2: Mesh Continuity Algorithm to find Elements inside Local Panel
4.2 Numbering of the Local Panel

In this method, the finite element model of the wing skin is first sliced using the set of curvilinear ribs and then each slice is again divided using the curvilinear spar. The numbering process of the local panels is shown in Figure 4-3.

![Figure 4-3: Numbering the Local Panels](image)

4.3 Algorithm to Determine the Boundary Elements

The Mesh-continuity algorithm is easy to apply once the elements along the outer boundary of the panel is determined. The general process to find this boundary elements is a bit complicated and will be discussed in detail in this section with a simple example where the objective is to split a rectangular plate (with holes) meshed with triangular elements into two panels. The process consists of two steps. The first step is to determine all elements along the free boundary that is part of one of the panels. The second step is to determine the elements with edges along the dividing curve, which are interior to the panel.
To determine elements along free boundary:

The plate shown in Figure 4-4 is needed to be split along Curve 2 (Curve 3 is the next curve in the family, Curve 1 is the previous one). First elements along the free edge of the panels will be found. In the algorithms described in this section, CHECKPOINT_NODE refers to the node with respect to which the positions of other nodes are determined. ELMENT_LIST_1 and ELMENT_LIST_2 are vector containing [Element ID, Connectivity Nodes] while NODE_LIST_1 and NODE_LIST_2 are list of Nodes. All of them are initialized as null sets.

Figure 4-4: Plate are Split by Three Non-Intersecting Curves
Figure 4-5: Algorithm to determine Elements along the Outer (free) Boundary
The nodes along Curve_1 is already known. Thus, finding the elements along this curve is very simple. All it needs is to find the elements that have at least two nodes along Curve_1. The elements in ELEMENT_LIST (O) and those along Curve_1 are marked in red.

The algorithm described above is not applicable to finding elements interior to the local panel with edges along Curve_2 as will include elements outside local panel as well (elements shown in Yellow in). This is because all those elements have edges along Curve_2. To find the elements only interior to the panels another algorithm known as Middle element algorithm has been developed.

**Middle Element Algorithm (MEA):** Here the objective is to determine elements on left/right side of a curve C (red) in Figure 8-6.

1. Initialize, CHECK_ELEM= E1, CHECK_NODE= N1, i=1
2. Add E1 to ELEMENT_LIST
3. Find Element with Node N1, share Edge with CHECK_ELEM, do not have Edge along curve C and not in ELEMENT_LIST (here Mi)
4. Add Mi to ELEMENT_LIST
5. CHECK_ELEM=Mi
6. i=i+1
7. Find Element with Node N1, share Edge with CHECK_ELEM, have Edge along curve C and not in ELEMENT_LIST
8. CHECK_NODE= N2

**Figure 4-6: Middle Element Algorithm**

**Figure 4-7: Middle Element Algorithm**
Applying the middle element algorithm to the flat plate example:

**Figure 4-8: Algorithm to find elements on one side of Curve 2**

**Figure 4-9: Using Middle Element Algorithm to find Elements in Local Panel**
Shuvodeep De

The elements contained in the vector ELEMENT_LIST consists of elements only above Curve 2 (marked by yellow and blue in Figure 4-9. The elements contained in ELEMENT_LIST (O) and ELEMENT_LIST forms the outer boundary of the panel above the curve 2. Once these elements along the outer boundary are determined, the mesh Continuity Algorithm can be used to find all the elements belonging to the panel.

4.4 Implementation on N+2 HSCT Wing-Box

The algorithm discussed above is first implemented on the simple Boeing N+2 HSCT Wing-box to create local panels from the upper and lower skin. Figure 4-10 shows a local panel from the top skin. This local panel will be later used to demonstrate thickness optimization using MSC.NASTRAN Solution 200.

![Figure 4-10: Local Panel from Upper Skin of Boeing N+2 HSCT Wingbox](image)

4.5 Implementation on CRM-Wing

The mesh continuity algorithm is implemented to create local panels from the upper and the lower skin of the wing. The FEM of the skin is divided first using its intersections with the ribs and then using its intersections with the spars. As shown in Figure 4-11, the FEM of the upper-skin of the wing near the wing-tip is divided into the inner section (green) and outer section (red) using the intersection of the rib that is nearest to the wing-tip (black line).

Starting from the internal rib adjacent to the wingtip, the first step is to find the common nodes between the skin and the rib. The second step is to find all the elements sharing these nodes.
which are contained in the outer section. After that, elements along the free boundary of the outer section are determined. It is not necessary to determine all the elements which share the nodes along the free boundary but knowing only elements having an edge along the free boundary it is sufficient. These boundary elements are used to find all elements of the outer section of the FEM using the mesh continuity algorithm.

![Figure 4-11: Inner and Outer Section of Wing FEM](image1)

The elements of the outer section are removed from the FEM and it is then sectioned using the intersection of the next rib as shown in. This process is repeated till the skin is divided by all the ribs.

![Figure 4-12: Sectioning the FEM using the Intersection of the Second Rib](image2)

The entire process is carried out using the family of spars. The set of nodes and set of elements common to each of the segments formed using the curvilinear ribs’ intersection and the segments formed using the curvilinear spars’ intersection, if not an empty set, is used to form the FEM of a local panels as shown in Figure 4-13.
The set of boundary nodes for each of the segment has already been determined in the process. Thus, the nodes in the union of the set of boundary nodes of the segment formed by the ribs and the set of boundary nodes of the segment formed by the spars that are shared by elements of the local panel are the panel’s boundary nodes.

![Figure 4-13: Common Elements between two Sections](image)

The advantage of this method is that it is completely based on set operations on the connectivity data of the FEM and independent of the coordinates of the nodes of the finite element model. It is also independent of the order in which the elements are distributed in the FEM. Thus, this method can be used to create local panels; regardless of where the SpaRibs start and where they end inside the wing. The local panels can have three, four or more edges.

Another advantage of this method of creating local panels is that the boundary nodes of each of the local panels is already determined and the information is stored to be used to impose boundary conditions during the optimization process.

The advantage of the Mesh-Continuity algorithm is that it is completely based on set operation and independent of the coordinates of the nodes of the finite element model. It can be used to create local panels of any shape and out of finite element model of any part like fuselage, tail etc. Figure 4-14 shows an example where a five-edged local panel is created out of a wing FEM.
4.6 Adding Stiffeners

In case of irregular panels, it is often difficult to determine the order in which the edges are numbered in MSC.PATRAN. Further a panel can have any number of edges. That is why adding stiffeners to the panels using the linked-shape method as described in Section 6.6 is somewhat challenging. In the alternate method, the stiffeners are created on the global model using guiding lines and are meshed with the quad elements. The element size should be larger than the height of the stiffeners to ensure that the stiffeners are represented by a chain of quad elements with each element sharing two common nodes with the skin. The nodes belonging to each of the local panels is already known by the method of creating local panels using the Mesh Continuity Algorithm. The .bdf file for the stiffeners are exported and read to find the quad elements with nodes common with the nodes of each of the local panels. These elements form the stiffeners for respective local panels. An example of such a stiffened panel is shown in Figure 4-15.
4.7 Proper Choice of Elements

While solving any problem using Finite Element Analysis, it is very important to choose the right type of element. MSC.PATRAN has the option to use different types of triangular and quadrilateral elements for meshing surfaces. In the original EBF3GLWingOpt framework constant strain triangular elements (CTRIA3) was used to mesh skin, SpaRibs as well as the stiffeners. On the stiffeners, there is always a chance of creating high aspect-ratio triangular element as shown in the example in Figure 4-16.

![Figure 4-16: Part of the Skin with Stiffeners as created by the Original EBF3GLWingOpt](image)

While solving a problem using finite element method it is extremely important to choose the appropriate element else the method might give inaccurate results. This section describes a study to explore the effect of the stiffener mesh on buckling factor. A rectangular plate with 100x100 units dimension is created in MSC.PATRAN environment. Four uniformly spaced stiffeners are created in the vertical direction and associated with the plate. Compressive displacement boundary (uniform magnitude of 0.01 units) conditions are imposed on the four edges of the panel as shown in Figure 4-17. The panel is first meshed with very fine Quad elements (element size 0.6 units) MSC.NASTRAN Sol 105 is used for buckling analysis and the eigenvector is shown in the figure.

![Figure 4-17: Rectangular Plate with Stiffeners and Buckling Eigen Vector](image)
The panel is re-meshed with triangular elements of the same size and the values of the first 10 eigenvalues are compared with those obtained with Quad elements. The results seem consistent as shown in Figure 4-18 (c). Figure 4-18 (a) and Figure 4-18 (b) show the stiffener mesh profile and the buckling eigenvector respectively.

**Figure 4-18:** (a) Mesh profile with Triangular Elements of Size 0.6 (b) First Buckling Eigenvalue (c) Comparison of first 10 Buckling Eigenvalues

The element size is increased progressively and compared with results given by mesh with quad elements if size 0.6. Results for element size 1.315, 2.88 and 2.0 are shown in Figure 4-19, Figure 4-20 and Figure 4-21 respectively. At this element size the stiffener mesh profile changes and that’s why results with these element sizes are shown. It is evident that the results become fairly inconsistent for coarse triangular mesh (element size 2.188 and beyond). However, the results are surprisingly consistent when quadrilateral elements are used for the stiffeners even for very coarse mesh (element size 5). This is another reason why quad elements are chosen for the stiffeners.
Figure 4-19: Buckling Results Comparison between Mesh of Triangular Elements of Size 1.315 and Quadrilateral Elements of Size 0.6

Figure 4-20: Buckling Results Comparison between Mesh of Triangular Elements of Size 2.0 and Quadrilateral Elements of Size 0.6
Figure 4-21: Buckling Results Comparison between Mesh with Triangular Elements of Size 2.88 and Quadrilateral Elements of Size 0.6

Figure 4-22: Buckling Results Comparison between Mesh with Size 5 (where Quadrilateral Elements used for Stiffeners and Triangular Elements used for Plate) and Mesh with Quadrilateral Elements of Size 0.6


4.8 Chapter Summary

This chapter gives a detailed description of a new algorithm to create stiffened local panels from the finite element model of the wing. The algorithm used in the original EBF3GLWingOpt framework to create local panels fails for many of the new SpaRibs configuration as it was limited to creation of four-edged panels with two edges along adjacent spars. The new algorithm is based on applying set operations on the connectivity data of the wing finite element model and is independent of nodal coordinates. It can be used to create local panels of any shape and size and can be used for structures other than the NASA CRM wing. Elements of the stiffeners attached with each of the local panels are found and stiffened panels are created. The chapter also presents a study to find proper elements on the stiffeners. Once the local panels are determined, the structure is ready for global-local optimization. The optimization framework is discussed in the next chapter.
Size and Shape Optimization Framework

The purpose of development of the EBF3GLWingOpt framework by Kapania et al. at Virginia Tech is that when fully completed it will have the multi-disciplinary, multi-objective optimization capability as well as a provision for creating SpaRibs geometry for a wide range of aircraft wings. The first section of the chapter gives a brief description of the progressive development of the EBF3GLWingOpt framework and the addition of capabilities

5.1 EBF3GLWingOptVersions

5.1.1 EBF3GLWingOpt 5.3.1 (Version delivered to NASA on Nov 2014)
- Defining local panels is based on comparisons using xyz coordinates -> Failure occurs for high SpaRibs curvatures.
- PSO function works with 2 design variables (Nbr_Inner_Ribs, Nbr_Outer_Ribs)

5.1.2 EBF3GLWingOpt 5.3.2:
- Defining local panels is made possible using the connectivity matrix and some coordinate transformation Works for high SpaRibs curvatures)
- Slightly faster and sometimes converges in less number of iterations.
- Comparison with V5.0.1: less than 1% difference in final weight
Shuvodeep De

- PSO function works for arbitrary number of design variables (Generalized)
- GBO function works for arbitrary number of design variables (Generalized)
- Flutter analysis added to the framework

5.1.3  *EBF3GLWingOpt 5.3.3*

- Enhanced function Classes.py to read input parameters using dictionary => Easier way for defining new input parameters in the code.
- The option of using the format “parameter = value” is added in the input file and the *config* file in order to follow a standard format.

5.1.4  *EBF3GLWingOpt 5.3.4*

- The global local analysis computes the flutter speed only for the optimized design, if it is requested in the input file.
- Multi-objective Optimization capability using Weighted-sum method. Added Flutter Speed to the evaluation function (PSO). \( f = \beta w^* w + \beta v^* \frac{1}{\sqrt{V_f}} \)

5.1.5  *EBF3GLWingOpt 5.3.5*

- The number of stiffeners has been added as a design variable and this new option can be ignored from the input file.
- The use of the input files has been modified in all the functions. Files are first copied to the temporary directory then to the working directory, before being used. This allows to run manually multiple analyses without any interference.

5.1.6  *EBF3GLWingOpt 5.3.6*

- Adapted to ARC super computers at Virginia Tech. The nodes are assigned at each job submission request => a new python function SetUpHosts.py creates the new hosts file.
- **MPI send/recv** commands has been updated in all parallel functions due to the different versions installed in the ARC cluster and our linux machines.

5.1.7  *EBF3GLWingOpt 5.3.7*
The function run MSC.Nastran.py is updated with more advanced techniques for a dynamic allocation of MSC.Nastran job while running in parallel. A better communication between the unrelated jobs is employed using the principle of lock file and the available memory is used at each step time to determine the decisions be made.

5.1.8  EBF3GLWingOpt 5.3.8
- More features have been added to control the running processes during PSO and avoid problems due to failure of non-feasible designs or memory saturation.
- The feature of creating lateral stiffeners is added.
- Satisfaction of the global buckling factor for the final design is ensured.
- This version is more robust

5.1.9  EBF3GLWingOpt 6.0.1
- New method of SpaRibs parametrization using Offset Parameters
- Create local panels independently from the geometry of the Sparibs
- Optimize the local panel using “Selective-stiffening” algorithm without isolating them from structure
- Minimize mesh failure after addition of stiffeners
- Remove elements with bad aspect ratio

5.2  Global-local Optimization using Nastran Sol 200

MSC.PATRAN Sol 200 is a gradient based optimization too. Attempt was made to move thickness optimization to MSC.PATRAN Sol 200. The goal is to minimize the weight of a wing while satisfying buckling constraint (BF ≥ 1) for each of the local panels. In addition, the maximum stress needs to be less than the yield stress of the material and the global BF ≥ 1. The optimization framework as shown in Figure 5-1 is developed in Python and it consists of five main steps: create geometry, mesh the geometry and verify the mesh, global-local optimization, selective stiffening of the panels to ensure buckling factor of the wing ≥ 1 and compute the flutter velocity.
Both the geometry and the mesh are generated in MSC.PATRAN. After that the mesh is verified for completeness. If the mesh is correctly extracted, the FEM is fed into the global-local optimization loop. Otherwise, the code expects the user to change the values of the design variables. As it is difficult to determine all the combinations of design variables for which the mesh might fail beforehand, this step is carried out manually.

The FEM on the wing is divided into skin panels and SpaRibs and a thickness is assigned to each of them. The initial thickness for all the panels and SpaRibs is assigned with the same value. The aerodynamic pressure field and the structural deformations are determined using MSC.NASTRAN Sol 144. More than one flight condition can be considered in the analysis. The displacement field output of MSC.NASTRAN Sol 144 is used to create boundary conditions for each of the skin panels and each of the SpaRibs. For the skin panels, displacement boundary conditions are imposed on the peripheral nodes which have already been determined during the process of creating local panels. For the SpaRibs, the boundary condition is imposed on the peripheral nodes as well as the nodes along its intersection with other SpaRibs as shown in the

Figure 5-1: Complete Optimization Framework
example in Figure 5-2. The blue dots represent the nodes on which boundary conditions are imposed.

![Boundary Conditions](image)

**Figure 5-2: Boundary Conditions: (a) Skin panel, (b) Rib, (c) Spar**

The thickness of each of the local component can be optimized thereafter by running NASTRAN Sol 200 and the thickness is updated in the respective .bdf file. A new .bdf file of the global wing is created by merging .bdf files of all the local components and NASTRAN Sol 144 is run using the new .bdf file. As the thicknesses of the elements has been changed, this will output a different displacement field than the one obtained previously. New boundary conditions are imposed on the local components (SpaRibs and skin panels) and the panels they are re-optimized followed by updating the thickness in the .bdf files. This process continues in an iterative way. The global-local optimization framework is shown in Figure 5-3.
Figure 5-3: Global-local Optimization of Aircraft Wing

Figure 5-4: Local Panel created from the Upper Skin of Boeing N+2

5.3 Boeing HSCT N+2 Panel Optimization

The local panel as created from the upper skin of the Boeing HSCT N+2 rear wingbox enforced with displacement constraints at its four edges is considered as shown in Figure 5-4. The objective of using NASTRAN sol200 for local panel optimization is to minimize the panel weight. Stiffeners are not added at this stage and panel thickness is the only design variable for an unstiffened panel. Stress constraints are not considered in the local panel optimization, it’s the
reason that the reaction force calculated from the enforced displacement constraints varying with panel thickness, so the obtained stress values vary with panel thickness. So, the buckling eigenvalue is the only one constraint considered in the local panel optimization. Constraints on von Mises stress will be considered in the global wing optimization. So, the optimization parameters for the local panel optimization in MSC.NASTRAN Sol 200 are,

**Objective:** minimize weight

**Constraint:** buckling eigenvalue $\lambda > 1.05$

**Design variable:** $0.02 \leq t \leq 10\text{inch, } t_0 = 2\text{inch}$

The gradient-based optimization algorithm was utilized in this problem, and the direct method was utilized for sensitivity matrix calculation. The iteration history of the structural total weight and the panel thickness is depicted in the Figure 5-5. The structural weight reduced by 85.5% from 106.74lbf to 15.51 lbf of optimal panel, and the total panel weight is 126.3 lbf including 110.8 lbf nonstructural weight. The critical buckling mode of the optimal panel was shown in Figure 5-6, the buckling eigenvalue of the optimal panel is 1.007, which satisfies the buckling load factor constraint.

![Figure 5-5 Iteration History of Panel Weight and Panel Thickness](image)
5.4 Selective Stiffening Algorithm

In global-local optimization, the buckling constraint for individual isolated panels and SpaRibs will be satisfied after each iteration. However, this does not ensure that if these panels are assembled to recreate the wing model, the buckling constraint and stress constraint of that model will automatically be satisfied. The so called Selective Stiffening Algorithm has been developed to ensure that for the final wing model after the optimization always has a buckling factor of 1 and stress is less than the yield stress of the material. The algorithm is applied once it is found that further convergence in the objective function i.e. weight cannot be achieved after a certain number of iterations.

The Selective-stiffening Algorithm consists of two steps. The first step ensures that the buckling factor (BF) of the structure is greater than 1. The second step is to check if the stress everywhere in the structure is less than the yield stress. Appropriate adjustments to the thickness variables are made till the stress constraint is satisfied for the entire structure.

**Step 1 (To satisfy buckling constraint):** It involves running NASTRAN SOL 144 and then NASTRAN SOL 105 (buckling analysis) and finding out the eigenvectors corresponding to the flight conditions with buckling factor less than 1. For each of these eigenvectors, the nodal
displacement magnitudes are calculated. After that the nodes with displacement bigger than a specified value are sorted out and the panels or stiffeners are determined where these nodes belong as shown in Figure 5-7. The properties (thicknesses) of the sorted panels or stiffeners are increased by a factor known as the Thickness Increment Factor (TIF). TIF is an empirical function of the buckling factor. SOL 105 is run for the modified global model and rest follows. This process is repeated until the buckling factor of the global model goes above 1. The complete algorithm is represented in the flowchart in Figure 5-8.

Step 2 (To satisfy buckling constraint): It involves running NASTRAN SOL 144. For each of the load conditions the elements violating the stress constraint are determined. The
thickness of panels to which these elements belong to are increased by a factor which is an empirical function of the maximum stress in the panel or stiffener and the yield stress of the material.

A very simple function for TIF is:

\[ TIF = (BF)^{-1/2} \]

This is based on the fact that for rectangular plates the critical thickness is inversely proportional to square-root of the Buckling Factor for in-plane buckling local. This function is used in the problem described in the next section.

### 5.5 Examples of Optimization (unstiffened CRM Wing) using MSC.NASTRAN Solution 200

The framework is verified by optimizing the CRM baseline model. This model is the same as the one with straight spars and ribs but without bifurcation of the rear spar. A second model with ribs perpendicular to the leading edge spar is also considered. This configuration is similar to
Shuvodeep De

that of Boeing 747-200. Figure 5-10 shows a side by side comparison of the two configurations analyzed.

![Model 1: Ribs parallel to Wing-root (a) Model 2: Ribs perpendicular to Leading-edge Spar.](image)

**Figure 5-10:** (a) Model 1: Ribs parallel to Wing-root (b) Model 2: Ribs perpendicular to Leading-edge Spar.

**Model 1:** The model is characterized by a total of 53,106 elements and 24,992 nodes. The initial thickness for all components is considered to be 0.2in. The material considered is Aluminum alloy 2024-T.3 and the initial weight is 11,321 lb. The leading edge and the trailing edge are modelled as a series of point masses attached to the leading edge spar and the trailing edge spar using Multiple-point Constraints (MPC). The total weight of the leading edge and the trailing edge is 8100 lb. A cruising Mach number of 0.85, with angle of attack -2°, 0°, 2°, 4°, and 6° are the five flight conditions considered in this study. Although the stress constraint has not been violated (maximum von Mises stress is 4.04E4 psi for AOA 6°) for this model, the buckling constraint is violated (BF is 0.0404 for AOA 6°). The weight converges within 1% after the 11th iteration of the global-local algorithm. Figure 5-11 shows the iteration history of the total weight of the wing, the difference between weights in consecutive iterations and the constraint function: minimum buckling factor.
After weight convergence is achieved, the buckling factor for AOA -2°, 0°, 2°, 4° and 6° is 0.79, 2.045, 1.248, 0.854 and 0.647 respectively. Thus, for AOA -2°, 4° and 6° the buckling constraint is yet to be satisfied. Buckling constraint satisfaction will be achieved by applying the selective stiffening algorithm.
Shuvodeep De

**Local convergence:** It is observed that different panels and *SpaRibs* converge at different rates. It is assumed that if the weight of a panel remains unchanged in consecutive iterations, it is unaffected by any change in the rest of the structure. Under that assumption, optimization of that panel can be skipped in subsequent iterations. Once the weight of a panel converges, it is not required to run NASTRAN SOL 200 for it in the subsequent iterations.

**Selective-stiffening of Local Panels:** The global-buckling factor after the final iteration is less than 1 for AOA -2°, 4° and 6°. Figure 5-12 shows the history of the buckling factors corresponding to different AOAs. The total weight of the wing is increased from 20,614 lb to 21,001 lb (1.87%) after the Selective Stiffening algorithm is applied. The total computational time for the analysis is 17 hours 58 minutes on the computer with 12 gigabytes of RAM and 6 cores Intel(R) Xeon(R) CPU W3680@ 3.33GHz.

![Figure 5-12: Iteration History of Buckling Factor and Weight in Selective Stiffening of Local Panels](image-url)
The thickness distribution of the wing-skin for the final design is given in Figure 5-13. The thickness values are in inches.

Figure 5-13: Thickness Distribution in the Final Design: (a) Upper Skin and (b) Lower Skin

Figure 5-14 shows the stress distribution for AOA 6 degree (for which maximum stress occurs). The stress constraint in the final design is satisfied as maximum von Mises stress = 2.66E4 psi.

Figure 5-14: Stress distribution for AOA 6 degree

Flutter Analysis: The flutter analysis using MSC.NASTRAN Sol 145 at Mach 0.85 and altitude range from 0 to 35000 ft. has been performed on the final design. The wing didn’t show any flutter inside the flight envelope.
Model 2: The geometry is characterized by a total of 49,374 elements and 23,330 nodes. The initial thickness, material and flight conditions are the same as in the previous analysis. The minimum buckling factor is 0.04473 and occurs for AOA 6°. In this case the stress constraint is also violated as the maximum von Mises stress is 5.95E4 psi. The history of the analysis is given in Figure 5-15 In this case also after the 11\textsuperscript{th} iteration the weight converges within 1 % limit. The weight after the 11\textsuperscript{th} iteration is 20,818 lb.

![Graphs showing history of total weight, weight difference, and minimum buckling factor](image-url)

**Figure 5-15: History of (a) Total Weight (b) Weight Difference between Consecutive Iterations (c) Minimum Buckling Factor**
Figure 5-16 shows the Selective-stiffening iteration history. It took 6 iterations to satisfy the buckling constraint. The final weight is 20,818 lb to 21,124 lb (1.47% increase).

**Figure 5-16: History of Buckling Factor and Weight in Selective Stiffening of Local Panels**

Figure 5-16 shows the thickness distribution for the upper and the lower skin for the final design.

**Figure 5-17: Thickness Distribution in the Final Design: (a) Upper Skin and (b) Lower Skin**
The maximum stress for this configuration is 39,200 psi and occurs for AOA 6 degree. The stress distribution is given in Figure 5-18.

![Stress Map for AOA 6 degree](image)

Figure 5-18: Stress Map for AOA 6 degree

Model 2 is around 0.58% heavier than Model 1 after optimization. Furthermore, in each of these example problems, it is found that the stress constraint and flutter constraints are satisfied in the final model.

### 5.6 Integrating Selective Stiffening Algorithm with EBF3WingOpt v5.8.1

The Global-Local optimization in the original EBF3GLWingOpt Framework ensures that all the local panels satisfy the aforementioned buckling, KS, and crippling constraints are satisfied. However, it does not guarantee that the global buckling factor of the final wing is satisfied. In order to ensure this condition, the Selective Stiffening algorithm (SS), is applied to find the panels vulnerable to buckling, and increase their thickness in an iterative process using a Thickness Increment Factor (TIF) equal to the square root of the inverse of the buckling factor. To investigate the effect of the SS process on the optimized wing, several cases of the same wing configuration, but with different number of longitudinal stiffeners, have been optimized with the Global-Local optimization process (Table 5-1). The global buckling factor of all these resulted structures is close to 0.7. The Selective Stiffening algorithm is, therefore, applied to each one of them (Table 5-2) and the global buckling factor became 1. They all converged in few iterations (less than 4 iterations) and took about 10 minutes each. The effect of this SS process on the wing weight and flutter speed
Shuvodeep De

is shown to be negligible (see Table 5-3). Therefore, the SS does not affect the decision of the Particle Swarm Optimization (described below) and will only be applied to the best obtained design.

**Table 5-1: Global-Local Optimization of Several Wing Configurations**

<table>
<thead>
<tr>
<th>Nbr of Stiffeners</th>
<th>Weight (lbs)</th>
<th>Flutter Speed (knots)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>12033.99</td>
<td>608.71</td>
</tr>
<tr>
<td>14</td>
<td>11838.08</td>
<td>595.72</td>
</tr>
<tr>
<td>15</td>
<td>11774.46</td>
<td>579.36</td>
</tr>
<tr>
<td>16</td>
<td>11692</td>
<td>567.71</td>
</tr>
<tr>
<td>17</td>
<td>11700.3</td>
<td>552.02</td>
</tr>
<tr>
<td>18</td>
<td>11748.35</td>
<td>546.08</td>
</tr>
<tr>
<td>19</td>
<td>11640.07</td>
<td>526.67</td>
</tr>
<tr>
<td>20</td>
<td>11656.08</td>
<td>475.44</td>
</tr>
</tbody>
</table>

**Table 5-2: Global-Local + Selective Stiffening**

<table>
<thead>
<tr>
<th>Weight (lbs)</th>
<th>Flutter Speed (knots)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12088.13</td>
<td>609.14</td>
</tr>
<tr>
<td>11888.67</td>
<td>596.09</td>
</tr>
<tr>
<td>11833.59</td>
<td>580.3</td>
</tr>
<tr>
<td>11746.59</td>
<td>568.34</td>
</tr>
<tr>
<td>11732.77</td>
<td>552.09</td>
</tr>
<tr>
<td>11782.28</td>
<td>546.33</td>
</tr>
<tr>
<td>11692.91</td>
<td>527.16</td>
</tr>
<tr>
<td>11688.63</td>
<td>475.94</td>
</tr>
</tbody>
</table>

**Table 5-3: Difference in Weight and Flutter Speed**

<table>
<thead>
<tr>
<th>Diff Weight (%)</th>
<th>Diff Flutter Speed (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.45</td>
<td>0.07</td>
</tr>
<tr>
<td>0.43</td>
<td>0.06</td>
</tr>
<tr>
<td>0.5</td>
<td>0.16</td>
</tr>
<tr>
<td>0.47</td>
<td>0.11</td>
</tr>
<tr>
<td>0.28</td>
<td>0.013</td>
</tr>
<tr>
<td>0.29</td>
<td>0.046</td>
</tr>
<tr>
<td>0.45</td>
<td>0.09</td>
</tr>
<tr>
<td>0.28</td>
<td>0.1</td>
</tr>
</tbody>
</table>
5.7 Optimization by Selective Stiffening

The major drawback of using MSC.NASTRAN Solution 200 is that results can be dependant on the initial guess and this is obvious because it is a gradient-based optimizer. It was also found that for certain initial guesses it gave infeasible solutions. In an effort to increase the accuracy and the efficiency of the analysis we introduced a different approach based solely on the application of the selective-stiffening algorithm, thus bypassing the global-local optimization of each panel. The initial thicknesses of the skins, spars and ribs are set to minimum gauge thickness (0.1 in). As the process doesn’t involve isolation and optimization of each of the local panels, a much coarse mesh can be used. As mentioned before the Thickness Increment Factor (TIF) is an empirical function. The weight for a certain configuration is depended on the TIF as well as the element size. The FEM becomes stiffer with an increase in the element size and overestimates the buckling factor. As a result, it is likely to underestimate the final weight. So, for comparing two different methods of size optimization, the methods should be tested for the same mesh and the same TIF.

When the expression shown in equation 5-1 for TIF was used, it was found to over-design many of the panels and give a total weight much higher than that obtained using MSC.NASTRAN Solution 200. It happens because:

- As evident from Figure 5-7, all the panels considered to be stiffened do not show the same magnitude of displacement i.e. they are not equally vulnerable to compressive force
- The SpaRibs unlike the skin panels are subjected more to shear force then in-plane compressive forces. Thus, it is not justified to use the same formulation of TIF.

Thus, the TIF is made dependent not only on buckling factor (BF) but also on the ratio of the maximum displacement of a panel and the global maximum displacement (here represented by m). A simple expression of TIF for the skin panels is formulated considering the following conditions:
The panel with maximum displacement (i.e. \( m = 1 \)) is stiffened by a factor equal to square-root of inverse of buckling factor (TIF_{max}).

When, \( BF = 1 \), no panel is stiffened i.e. \( TIF = 1 \) for every panel.

For any BF between 0 and 1, panels are stiffened by a factor which varies linear with m.

Considering these conditions the expression for TIF for the skin panels is:

\[
\begin{align*}
TIF_{\text{max}} &= (BF)^{-1/2} \\
TIF &= m(TIF_{\text{max}} - 1) + 1
\end{align*}
\]

where, \( D_{\text{max}}^G \) = Maximum displacement in eigenvector

\( D_{\text{max}}^P \) = Maximum displacement in a panel

\[
m = \frac{D_{\text{max}}^P}{D_{\text{max}}^G}
\]

When the same formulation is used for the SpaRibs it overdesigned them because unlike the skin panels, they are not subjected to in-plane compressive loads. In a conventional aircraft consisting of straight ribs parallel to wing-root and straight spars running from wing-root to wing-tip, the ribs and the web of the spars are mainly subjected to shear loads while the spar-caps carry bending loads. When they are replaced by SpaRibs, the loading becomes more complicated and dependent of the SpaRibs configuration.

The expression of TIF for the SpaRibs is obtained considering the same conditions as for the skin panels only except that the TIF_{max} is considered the sixth root of inverse of buckling factor. This is obtained entirely by trial and error process as it gave the least estimate of weight. The expression of TIF for the SpaRibs is thus,

\[
\begin{align*}
TIF_{\text{max}} &= (BF)^{-1/6} \\
TIF &= m(TIF_{\text{max}} - 1) + 1
\end{align*}
\]

Although with this function, the weight computed for the given set of constraints and load conditions was found to be significantly lower than that obtained by MSC.NASTRAN Sol 200 as
discussed before, it is found that the optimization process is very slow i.e. it takes a large number of iterations in the Selective-stiffening process before the final design is obtained. However, if the panels are slightly overdesigned to buckling factor $1.05$, it gives fairly reliable output for the weight with a lower number of iterations. The expression for TIF gets modified to:

$$\text{TIF} = m \left( \frac{BF}{1.05} \right)^{-1/2} - 1 \quad \text{for Skin Panels}$$

$$= m \left( \frac{BF}{1.05} \right)^{-1/6} - 1 \quad \text{for SpaRibs}$$

where,

- $D_{max_G}$ = Maximum displacement in eigenvector
- $D_{max_P}$ = Maximum displacement in a panel

$$m = \frac{D_{max_P}}{D_{max_G}}$$

In all the results discussed in the next section, this value of TIF has been used.

### 5.8 Baseline Model and Mesh Convergence

Size optimization by Selective Stiffening is a computationally expensive process. The algorithm is first implemented to optimize the unstiffened wing with SpaRibs configuration as the CRM baseline model. The wing geometry (upper skin, lower skin, SpaRibs) is meshed using the “Paver” option in MSC.PATRAN. The initial configuration has 0.1in thickness for all SpaRibs and for every skin panel. This model violates both buckling and stress constraints. Table 5-4 summarizes the number of elements, number of nodes, the structural weight estimation and corresponding CPU time for various mesh sizes after optimizing the thickness of the local panels and SpaRibs using the Selective Stiffening Algorithm. The difference in structural weight estimation and computational time is calculated with respect to the model with element size (represented by maximum allowed element edge length) equal to 3 in (extremely refined). It is evident that mesh convergence within 0.5% has been achieved at element size 4 in. On the other hand, increasing the mesh size beyond 8 in does not decrease the computation time further. The analysis was run on a computer with Intel Linux 2.6.32-696.el6.x86_64 platform with clock frequency 3.33 GHz and 12 GB RAM. Since coarse mesh means stiffer elements, it results in
higher buckling factor for the same load condition and this results in an under-estimation of the structural weight. Furthermore, these models have a flutter speed that is larger than 680 knots i.e. they satisfy the flutter constraint. It should be mentioned that the ‘Total Weight’ given in Table 5-4 is sum of structural weight and total weight of leading and trailing edges (8100 lb.)

Table 5-4: Weight Estimation and Computational Time for Different number of Elements/Nodes for CRM Baseline Model

<table>
<thead>
<tr>
<th>Element Size (in)</th>
<th>Number of Elements</th>
<th>Number of Nodes</th>
<th>Structural Weight (lb)</th>
<th>Total Weight (lb)</th>
<th>CPU Time (min)</th>
<th>Change in Structural Weight (%)</th>
<th>Change in CPU Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>20739</td>
<td>9438</td>
<td>18610</td>
<td>26710</td>
<td>281</td>
<td>-2.37</td>
<td>-60.74</td>
</tr>
<tr>
<td>7</td>
<td>28416</td>
<td>13118</td>
<td>18701</td>
<td>26801</td>
<td>291</td>
<td>-1.89</td>
<td>-59.31</td>
</tr>
<tr>
<td>6</td>
<td>36407</td>
<td>16925</td>
<td>18718</td>
<td>26818</td>
<td>299</td>
<td>-1.80</td>
<td>-58.17</td>
</tr>
<tr>
<td>5</td>
<td>52486</td>
<td>24687</td>
<td>18908</td>
<td>27008</td>
<td>327</td>
<td>-0.81</td>
<td>-54.15</td>
</tr>
<tr>
<td>4</td>
<td>82029</td>
<td>39052</td>
<td>19009</td>
<td>27109</td>
<td>421</td>
<td>-0.28</td>
<td>-40.69</td>
</tr>
<tr>
<td>3</td>
<td>144388</td>
<td>69570</td>
<td>19062</td>
<td>27162</td>
<td>705</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 5-19: Change in CPU Time (a) and Weight Estimation (b) with Mesh Size for CRM Baseline Model
The thickness distribution as obtained with element size 4 is given in Figure 5-20.

![Thickness Distribution](image)

Figure 5-20: Thickness Distribution of Unstiffened Baseline Model after Local Panel Thickness Optimization

### 5.9 *SpaRibs* Topology Optimization using PSO

As mentioned before, the *SpaRibs* profile is defined by a set of design variables i.e. the Offset Parameters. In addition, the number of spars, ribs and stiffeners can also be considered as design variables. To consider both continuous and discrete design variables, the Selective-stiffening Algorithm is integrated with Particle Swarm Optimization (PSO) Algorithm is used. It is implemented using parallel processing. License cycle-check method and memory self-adjustment method, developed by Jrad et al. [14] have been used to overcome limitations on number of MSC licenses available and memory saturation, respectively. Since Selective-stiffening with small element size is computationally expensive, a coarse mesh (element size 8 in) is used inside the optimizer. The best design once obtained is re-meshed with element size 4 in to obtain the final result.

The optimization framework has been developed such that it can be used to optimize multiple often contradictory objectives (e.g. reducing weight and maximizing the flutter velocity) by the weighted sum method.
If the goal is to reduce the weight and increase flutter velocity of a wing at the same time, the objective (for weighted-sum method) can be considered as:

\[ f = \frac{W}{W_0} + \sqrt{\frac{V_0}{V_f}} \]  

(3)

Where \(W\) is the wing weight (lbs), \(V_f\) is the flutter speed (knots), \(W_0\) is the effective weight (lb), and \(V_0\) is the effective flutter velocity (knots).

For some of the SpaRibs configurations, the analysis fails. There are several possible reasons for analysis failure including failure to generate complete geometry or mesh in MSC.PATRAN due to non-feasible design geometry. To verify if the complete mesh has been created, it is checked if the number of surface created in MSC.PATRAN is equal to the number of surfaces on which elements have been generated. If the geometry of all the SpaRibs is not generated, the creation of local panels will fail and will eventually cause failure of the analysis. To prevent the analysis from stopping, a very high value (penalty) is assigned to the objective function for the particles for which analysis fails. As a result, in the next generation the particle is automatically discarded. The solution is said to converge when the change in the global best falls within specified limit (here k %) The complete optimization framework is shown in Figure 5-21.

![Figure 5-21: Framework for SpaRibs Optimization using Particle Swarm Optimization (PSO) Algorithm for N particles](image-url)
In this section, an example of SpaRibs optimization is described. At first a parametric study is conducted using element size 8 in to explore the influence of the number of stiffeners on weight. The SpaRibs configuration here is same as the baseline model and the stiffener height is 3 in. in the inner wing-box while 2 in. in the outer wing-box. Each of these models satisfy the flutter constraints and it is found that increasing the number of stiffeners to up to 14 helps in reducing the total weight. Increasing the number of stiffeners beyond 14 results in small reduction in weight of skins but since weight of the stiffeners is going up, the total structural weight increases. The weight of the structure for different number of stiffeners is given in Figure 5-22.

<table>
<thead>
<tr>
<th>Number of Stiffener</th>
<th>Weight of Skins and SpaRibs (lb)</th>
<th>Weight of Stiffeners (lb)</th>
<th>Total Weight (lb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>12323</td>
<td>898.35</td>
<td>13221.35</td>
</tr>
<tr>
<td>9</td>
<td>11998</td>
<td>961.44</td>
<td>12959.44</td>
</tr>
<tr>
<td>10</td>
<td>11155</td>
<td>1018.1</td>
<td>12173.1</td>
</tr>
<tr>
<td>11</td>
<td>10744</td>
<td>1110</td>
<td>11854</td>
</tr>
<tr>
<td>12</td>
<td>10191</td>
<td>1136</td>
<td>11327</td>
</tr>
<tr>
<td>13</td>
<td>9989.2</td>
<td>1196.3</td>
<td>11185.5</td>
</tr>
<tr>
<td>14 (Model 1)</td>
<td>9661.2</td>
<td>1207.6</td>
<td>10868.8</td>
</tr>
<tr>
<td>15</td>
<td>9600.7</td>
<td>1314.7</td>
<td>10915.4</td>
</tr>
</tbody>
</table>

Table 5-5: Variation of Weight and Flutter Velocity with Number of Stiffeners

Figure 5-22: Structural Weight for Different number of Stiffeners (Mesh Size: 8 in)
5.10.1 Case 1: Minimize Weight

The objective is to optimize Weight (W). The configuration of spar is same as the baseline model. The curvilinear ribs are uniformly spaced and their shape is defined by six Offset Parameters, as described before. The constraining curves of start-point of the B-splines gets mapped in the physical space as intersecting curve between the leading-edge spar and upper/lower skin with 10% extension (user specified) on either side. Similarly, the constraining curve for end-point of B-splines for the SpaRibs is mapped in the physical space as intersecting curve between the leading-edge spar and upper/lower skin with 10% extension on either side. The number of internal SpaRibs is same as that of the baseline model (37).

Figure 5-23: PSO Iteration History to Minimize Weight and SpaRibs profile for Best Design (Model 2)

Table 5-6: Offset Parameters and Weight of Particles Analyzed

<table>
<thead>
<tr>
<th>O₁₁</th>
<th>O₁₂</th>
<th>O₂₁</th>
<th>O₂₂</th>
<th>O₃₁</th>
<th>O₃₂</th>
<th>Weight (lb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.150</td>
<td>0.156</td>
<td>0.062</td>
<td>0.102</td>
<td>0.136</td>
<td>0.172</td>
<td>10601.500</td>
</tr>
<tr>
<td>0.106</td>
<td>0.160</td>
<td>0.118</td>
<td>0.134</td>
<td>0.040</td>
<td>0.164</td>
<td>10593.200</td>
</tr>
<tr>
<td>0.160</td>
<td>0.113</td>
<td>0.076</td>
<td>0.107</td>
<td>0.015</td>
<td>0.142</td>
<td>10602.500</td>
</tr>
<tr>
<td>0.116</td>
<td>0.169</td>
<td>0.019</td>
<td>0.169</td>
<td>0.046</td>
<td>0.155</td>
<td>10630.700</td>
</tr>
<tr>
<td>0.118</td>
<td>0.167</td>
<td>0.103</td>
<td>0.171</td>
<td>0.103</td>
<td>0.169</td>
<td>10615.800</td>
</tr>
<tr>
<td>0.114</td>
<td>0.107</td>
<td>0.103</td>
<td>0.170</td>
<td>0.105</td>
<td>0.134</td>
<td>10492.000</td>
</tr>
<tr>
<td>0.077</td>
<td>0.146</td>
<td>0.164</td>
<td>0.145</td>
<td>0.103</td>
<td>0.138</td>
<td>12639.100</td>
</tr>
<tr>
<td>0.071</td>
<td>0.105</td>
<td>0.129</td>
<td>0.159</td>
<td>0.093</td>
<td>0.146</td>
<td>10672.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>0.163</td>
<td>0.104</td>
<td>0.016</td>
<td>0.168</td>
<td>0.164</td>
<td>0.157</td>
<td>12008.400</td>
</tr>
<tr>
<td>0.036</td>
<td>0.169</td>
<td>0.155</td>
<td>0.166</td>
<td>0.169</td>
<td>0.170</td>
<td>13488.000</td>
</tr>
<tr>
<td>0.054</td>
<td>0.173</td>
<td>0.074</td>
<td>0.109</td>
<td>0.144</td>
<td>0.165</td>
<td>11426.700</td>
</tr>
<tr>
<td>0.128</td>
<td>0.100</td>
<td>0.139</td>
<td>0.175</td>
<td>0.093</td>
<td>0.138</td>
<td>10569.900</td>
</tr>
<tr>
<td>0.117</td>
<td>0.112</td>
<td>0.125</td>
<td>0.120</td>
<td>0.112</td>
<td>0.125</td>
<td>10204.000</td>
</tr>
<tr>
<td>0.088</td>
<td>0.127</td>
<td>0.094</td>
<td>0.154</td>
<td>0.107</td>
<td>0.132</td>
<td>10587.800</td>
</tr>
<tr>
<td>0.143</td>
<td>0.114</td>
<td>0.111</td>
<td>0.175</td>
<td>0.114</td>
<td>0.139</td>
<td>10288.800</td>
</tr>
<tr>
<td>0.159</td>
<td>0.100</td>
<td>0.110</td>
<td>0.175</td>
<td>0.161</td>
<td>0.125</td>
<td>10890.700</td>
</tr>
<tr>
<td>0.149</td>
<td>0.110</td>
<td>0.124</td>
<td>0.115</td>
<td>0.126</td>
<td>0.153</td>
<td>10316.600</td>
</tr>
<tr>
<td>0.175</td>
<td>0.121</td>
<td>0.072</td>
<td>0.167</td>
<td>0.144</td>
<td>0.125</td>
<td>10977.400</td>
</tr>
<tr>
<td>0.161</td>
<td>0.132</td>
<td>0.100</td>
<td>0.175</td>
<td>0.166</td>
<td>0.125</td>
<td>10896.400</td>
</tr>
<tr>
<td>0.147</td>
<td>0.171</td>
<td>0.059</td>
<td>0.175</td>
<td>0.157</td>
<td>0.167</td>
<td>11178.900</td>
</tr>
<tr>
<td>0.123</td>
<td>0.101</td>
<td>0.076</td>
<td>0.175</td>
<td>0.107</td>
<td>0.133</td>
<td>10663.100</td>
</tr>
<tr>
<td>0.117</td>
<td>0.122</td>
<td>0.054</td>
<td>0.172</td>
<td>0.126</td>
<td>0.154</td>
<td>10744.600</td>
</tr>
<tr>
<td>0.113</td>
<td>0.147</td>
<td>0.106</td>
<td>0.172</td>
<td>0.111</td>
<td>0.125</td>
<td>10739.200</td>
</tr>
<tr>
<td>0.175</td>
<td>0.100</td>
<td>0.092</td>
<td>0.175</td>
<td>0.175</td>
<td>0.129</td>
<td>11058.200</td>
</tr>
<tr>
<td>0.114</td>
<td>0.114</td>
<td>0.106</td>
<td>0.174</td>
<td>0.110</td>
<td>0.134</td>
<td>10507.200</td>
</tr>
<tr>
<td>0.114</td>
<td>0.100</td>
<td>0.097</td>
<td>0.138</td>
<td>0.057</td>
<td>0.125</td>
<td>10512.300</td>
</tr>
<tr>
<td>0.160</td>
<td>0.108</td>
<td>0.065</td>
<td>0.169</td>
<td>0.152</td>
<td>0.151</td>
<td>11101.300</td>
</tr>
<tr>
<td>0.098</td>
<td>0.160</td>
<td>0.163</td>
<td>0.149</td>
<td>0.113</td>
<td>0.130</td>
<td>12139.000</td>
</tr>
<tr>
<td>0.117</td>
<td>0.126</td>
<td>0.137</td>
<td>0.114</td>
<td>0.042</td>
<td>0.138</td>
<td>10953.700</td>
</tr>
<tr>
<td>0.114</td>
<td>0.110</td>
<td>0.144</td>
<td>0.107</td>
<td>0.007</td>
<td>0.125</td>
<td>11729.800</td>
</tr>
<tr>
<td>0.126</td>
<td>0.100</td>
<td>0.100</td>
<td>0.165</td>
<td>0.147</td>
<td>0.129</td>
<td>10546.900</td>
</tr>
<tr>
<td>0.175</td>
<td>0.116</td>
<td>0.144</td>
<td>0.126</td>
<td>0.154</td>
<td>0.125</td>
<td>11006.400</td>
</tr>
<tr>
<td>0.117</td>
<td>0.108</td>
<td>0.133</td>
<td>0.112</td>
<td>0.115</td>
<td>0.125</td>
<td>10810.900</td>
</tr>
</tbody>
</table>
5.10.2: Case 2: Minimize Weight and Maximize Flutter Velocity

In the optimization problem, 14 stiffeners with same height, as mentioned before, are considered. An objective function as mentioned in equation (3) is constructed. The values of $W_0$ and $V_0$ are considered 10 lb. and 158.11 knots following studies by Jrad et al. [14].

The 6 Offset Parameters are considered as design variables and the PSO is run with 24 particles per generation. To avoid configurations with undesirable highly curvilinear SpaRibs appropriate limits are specified for the Offset Parameters. Convergence is checked every two generations. As shown in Figure 5-24 the solution converges after the 4th generation.

![Figure 5-24: Particle Swarm Optimization History to Minimize Weight and Maximize Flutter Velocity and SpaRibs Profile of Best Design (Model 2)](image)

<table>
<thead>
<tr>
<th>Weight (lb)</th>
<th>Flutter Velocity (knots)</th>
<th>O11</th>
<th>O12</th>
<th>O21</th>
<th>O22</th>
<th>O31</th>
<th>O32</th>
<th>Objective Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>10670.2</td>
<td>0.000</td>
<td>0.125</td>
<td>0.110</td>
<td>0.140</td>
<td>0.160</td>
<td>0.085</td>
<td>0.131</td>
<td>35097368</td>
</tr>
<tr>
<td>11127.2</td>
<td>573.791</td>
<td>0.038</td>
<td>0.124</td>
<td>0.055</td>
<td>0.120</td>
<td>0.079</td>
<td>0.136</td>
<td>2156.4</td>
</tr>
<tr>
<td>10453.6</td>
<td>541.412</td>
<td>0.144</td>
<td>0.113</td>
<td>0.076</td>
<td>0.175</td>
<td>0.071</td>
<td>0.163</td>
<td>2119.8</td>
</tr>
<tr>
<td>10599.2</td>
<td>529.783</td>
<td>0.119</td>
<td>0.134</td>
<td>0.043</td>
<td>0.155</td>
<td>0.043</td>
<td>0.173</td>
<td>2146.1</td>
</tr>
<tr>
<td>10562.0</td>
<td>542.244</td>
<td>0.124</td>
<td>0.119</td>
<td>0.102</td>
<td>0.124</td>
<td>0.096</td>
<td>0.137</td>
<td>2129.8</td>
</tr>
<tr>
<td>10636.7</td>
<td>555.696</td>
<td>0.074</td>
<td>0.128</td>
<td>0.098</td>
<td>0.112</td>
<td>0.113</td>
<td>0.143</td>
<td>2124.2</td>
</tr>
<tr>
<td>12085.9</td>
<td>560.319</td>
<td>0.103</td>
<td>0.108</td>
<td>0.158</td>
<td>0.111</td>
<td>0.148</td>
<td>0.156</td>
<td>2264.7</td>
</tr>
<tr>
<td>10873.3</td>
<td>572.666</td>
<td>0.078</td>
<td>0.122</td>
<td>0.117</td>
<td>0.158</td>
<td>0.136</td>
<td>0.143</td>
<td>2132.0</td>
</tr>
</tbody>
</table>
### Final Results after Mesh Refinement

The model with spars and ribs as the baseline model and 14 stiffeners (named Model 1) and the best solution obtained by PSO (named Model 2) are re-meshed with mesh size 4 and local

<table>
<thead>
<tr>
<th>Model</th>
<th>Weight (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>544.602</td>
</tr>
<tr>
<td>Model 2</td>
<td>556.305</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Model 1</td>
<td>555.492</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Weight (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>2096.8</td>
</tr>
<tr>
<td>Model 2</td>
<td>2213.4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Model 1</td>
<td>2232.1</td>
</tr>
</tbody>
</table>
Shuvodeep De

panel thicknesses are optimized using Selective-stiffening. Both models satisfy stress and buckling constraints for both flight conditions. The flutter speeds are also significantly higher than 328 knots.

Table 5-7 summarizes the structural properties while Figure 5-25 shows the thickness distribution after local panel optimization for Model 1. Figure 5-26 shows the first ending and the first torsional modes.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Weight</td>
<td>11568 lb.</td>
</tr>
<tr>
<td>Total Weight</td>
<td>19669 lb.</td>
</tr>
<tr>
<td>Reduction of Structural Weight</td>
<td>39.14%</td>
</tr>
<tr>
<td>Buckling Factor for AOA 6 degree</td>
<td>1.001</td>
</tr>
<tr>
<td>Buckling Factor for AOA -2 degree</td>
<td>1.006</td>
</tr>
<tr>
<td>Freq. for First Bending Mode</td>
<td>1.29 Hz</td>
</tr>
<tr>
<td>Freq. for First Torsional Mode</td>
<td>9.36 Hz</td>
</tr>
<tr>
<td>Maximum von Mises Stress</td>
<td>39200 psi</td>
</tr>
<tr>
<td>Flutter Velocity</td>
<td>602 knots</td>
</tr>
</tbody>
</table>

Figure 5-25: Thickness Distribution after Local Panel Optimization
Figure 5-26: (a) First Bending and (b) First Torsional (Right) Mode

Table 5-8 summarizes the structural properties while shows the thickness distribution after local panel optimization for Model 2. Figure 5-26 shows the first bending and the first torsional modes.

Table 5-8: Structural Properties of Model 2

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Weight</td>
<td>11002 lb.</td>
</tr>
<tr>
<td>Total Weight</td>
<td>19102 lb.</td>
</tr>
<tr>
<td>Reduction of Structural Weight</td>
<td>42.1%</td>
</tr>
<tr>
<td>Buckling Factor for AOA 6 degree</td>
<td>1.001</td>
</tr>
<tr>
<td>Buckling Factor for AOA -2 degree</td>
<td>1.05</td>
</tr>
<tr>
<td>Freq. for First Bending Mode</td>
<td>1.29 Hz</td>
</tr>
<tr>
<td>Freq. for First Torsional Mode</td>
<td>9.32 Hz</td>
</tr>
<tr>
<td>Maximum von Mises Stress</td>
<td>47200 psi</td>
</tr>
<tr>
<td>Flutter Velocity (knots)</td>
<td>612 knots</td>
</tr>
</tbody>
</table>

Figure 5-27: Thickness Distribution after Local Panel Optimization
Table 5-8 summarizes the structural properties while shows the thickness distribution after local panel optimization for Model 2. Figure 5-26 shows the first bending and the first torsional modes.

Table 5-9: Structural Properties of Model 3

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Weight</td>
<td>11082 lb.</td>
</tr>
<tr>
<td>Total Weight</td>
<td>19182 lb.</td>
</tr>
<tr>
<td>Reduction of Structural Weight</td>
<td>41.9%</td>
</tr>
<tr>
<td>Buckling Factor for AOA 6 degree</td>
<td>1.001</td>
</tr>
<tr>
<td>Buckling Factor for AOA -2 degree</td>
<td>1.05</td>
</tr>
<tr>
<td>Freq. for First Bending Mode</td>
<td>1.29 Hz</td>
</tr>
<tr>
<td>Freq. for First Torsional Mode</td>
<td>9.31 Hz</td>
</tr>
<tr>
<td>Maximum von Mises Stress</td>
<td>45500 psi</td>
</tr>
<tr>
<td>Flutter Velocity (knots)</td>
<td>608 knots</td>
</tr>
</tbody>
</table>
5.12 Discussions

The optimized wing contains skin panels of different thickness. It is extremely challenging to manufacture such a wing-skin using conventional manufacturing because each panel needs to be machined to the desirable thickness. It is assumed that it will be feasible to manufacture such a wing using additive manufacturing techniques. In either case, further studies need to be conducted to conclude whether production of such a structure will be cost effective.

Also, it needs to be emphasized that the solution is obtained by an empirical expression of the Thickness Increment Factor (TIF). Although the stress constraint and buckling constraint are always satisfied and almost a fully stressed design is obtained, it cannot be concluded that all no panel have been overdesigned. Further study need to be conducted to determine the sensitivity of the solution on TIF.
In this chapter two processes of local panel size optimization are described. One of them is a global-local optimization process similar to EBF3PanelOpt discussed in the second chapter only except the panel thickness is obtained by MSC.NASTRAN Sol 200 instead of an empirical formula (as in EBF3PanelOpt). It was found that in this process even if weight convergence is obtained the global buckling constraint is not satisfied. To satisfy that, the Selective Stiffening Algorithm has been developed which finds the panels vulnerable to buckling and locally stiffens them in an iterative process until the buckling factor for all flight conditions are at least greater than 1. As MSC.NASTRAN Sol 200 is a gradient based optimizer the solution is dependent on the initial guess. In an effort to increase the accuracy and the efficiency of the analysis a different approach is introduced based solely on the application of the selective-stiffening algorithm. All the local panels are set to minimum gauge thickness and Selective Stiffening process is applied to the structure repeated till global buckling and stress constraints are satisfied. For this process, mesh convergence is achieved for element size smaller than 4 in. The framework to size the local panels using Selective Stiffening Algorithm is integrated with Particle Swarm Optimization Algorithm to optimize the SpaRibs topology.
6 Conclusion and Future Work

The work is a step towards the development of the EBF3GLWingOpt software by Kapania et al. at Virginia Tech that exhibits multi-disciplinary multi-objective optimization capability as well as provision for creating Sparibs geometry for wings of different shapes. The creation of Sparibs using limited number of design variables in a wing with two sections along with the use of PSO to optimize weight and flutter velocity simultaneously was demonstrated. Parametric study that gives an idea about the effect of multiple design variables on the wing weight and flutter speed has been conducted in order to make a wise selection of the most influential variables in the particle swarm optimization and in order to give guidelines for designers on how to place the structure’s components in order to obtain the desired wing weight and flutter velocity.

It has always been an interest if better results can be obtained by not restricting the SpaRibs to a single to a wing-box. There has been significant progress towards the development of a more generalized parameterization method. A number of limitations in the original EBF3GLWingOpt framework have been identified and removed.

1. The SpaRibs parameterization is made more general by adding a set of design variables (known as offset parameters). The code takes care of C1 continuity of the SpaRibs at the trailing edge break.
2. A more generalized profile of SpaRibs implies that they will encompass irregularly shaped local panels and it was found that the algorithm used to generate local panels in the original
EBF3GLWingOpt framework can no longer be used (as it is dependent on coordinates and written for creating local panels with four edges). An algorithm has been developed which is purely based on set operation on connectivity data of the FEM (independent of coordinates) and can be used to create local panels of any shape.

3. It was found that for a large number of designs MSC.PATRAN fails to generate the mesh and the cause is often attachment of parts of stiffeners that create parts on the skins that are way smaller than specified element size. This problem has been mitigated by identifying these stiffeners and avoiding associating them with the skin.

4. It was found that the global-local optimization framework in the original EBF3GLWingOpt framework does not ensure the satisfaction of global buckling constraint (BF>1) even if the total weight converges within a certain limit. An algorithm (known as Selective-stiffening Algorithm) has been developed which identifies the panels susceptible to buckling and increases their thickness to increase the global-buckling factor. This goes in an iterative manner till the global buckling factor becomes greater than 1. This algorithm can be implemented after Global-local optimization so that the final design satisfies buckling constraint both at the local and global level.

The Selective-stiffening algorithm alone can be used to size optimize the wing. In this case after buckling constraint is satisfied, the structure should be checked for stress. If stress constraint is violated the elements violating it are increased in thickness

In the future, this optimization framework will be integrated with the Multi-objective, Particle Swarm Optimization (MOPSO) code that has been developed to optimize SpaRibs topology to reduce weight and maximize flutter velocity.
7 References


Shuvodeep De


Shuvodeep De


