



CONCEPTUAL LAYOUT OF WING STRUCTURE USING TOPOLOGY OPTIMIZATION  
FOR MORPHING MICRO AIR VEHICLES IN A PERCHING MANEUVER

THESIS

James R. Elgersma, Civ, USAF

AFIT/GAE/ENY/12-M11

DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY

**AIR FORCE INSTITUTE OF TECHNOLOGY**

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT/GAE/ENY/12-M11

CONCEPTUAL LAYOUT OF WING STRUCTURE USING TOPOLOGY  
OPTIMIZATION FOR MORPHING MICRO AIR VEHICLES IN A  
PERCHING MANEUVER

THESIS

Presented to the Faculty  
Department of Aeronautics and Astronautics  
Air Force Institute of Technology  
Air University  
Air Education and Training Command  
In Partial Fulfillment of the Requirements for the  
Degree of Master of Science in Aeronautical Engineering

James R. Elgersma, BSME  
Civ, USAF

March 2012

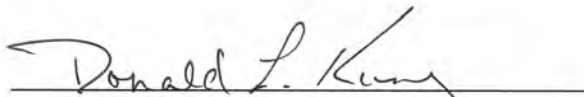
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GAE/ENY/12-M11

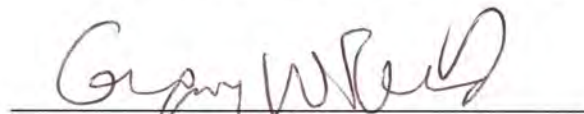
CONCEPTUAL LAYOUT OF WING STRUCTURE USING TOPOLOGY  
OPTIMIZATION FOR MORPHING MICRO AIR VEHICLES IN A  
PERCHING MANEUVER

James R. Elgersma, BSME  
Civ, USAF

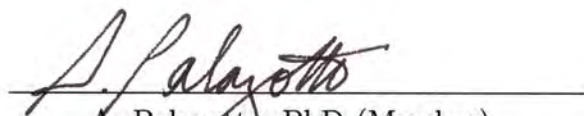
Approved:

  
\_\_\_\_\_  
D.L. Kunz, PhD, P.E. (Chairman)

9 March 2012  
date

  
\_\_\_\_\_  
G. Reich, PhD (Member)

9 March 2012  
date

  
\_\_\_\_\_  
A. Palazotto, PhD (Member)

2/29/12  
date

*Abstract*

A topology optimization model for conceptual wing structure layouts of morphing micro air vehicles (MAVs) has been developed and implemented in MATLAB. Specifically, a six degree-of-freedom finite element (FE) model with a general quadrilateral discretization scheme was created by superposition of a known simple linear plane membrane element and a Kirchhoff plate bending element derived herein. The purpose of the six degree-of-freedom model was to accommodate in-plane and out-of-plane aerodynamic loading combinations. The FE model was validated and the MATLAB implementation was verified with classical beam and plate solutions. A compliance minimization optimization objective was then formulated with the Solid Isotropic Material with Penalization (SIMP) method, subject to the equilibrium constraint computed by the FE model, and solved with the Optimality Criteria (OC) method. With the topology optimization model in place, four aerodynamic loading scenarios were extracted from points along a feasible MAV perching flight trajectory and used to determine wing thickness distributions for given planform shapes. The results suggest conceptual structural layouts in morphing MAVs, but equally important, the simple MATLAB implementation of the model can be adapted for a variety of objective statements for MAV morphing wing design.

*To Mom and Dad—  
my greatest teachers*

## *Acknowledgements*

First and foremost, I am thankful to the Lord for the grandeur of creation. Brilliant minds and captivated pupils will spend a long time rediscovering the complexity and elegance that God merely spoke into existence a few thousand years ago.

I sincerely thank Dr. Donald Kunz for the exceptional guidance and instruction throughout this thesis, as well as the many hours spent editing this document.

I am very grateful to Dr. Gregory Reich and Dr. James Joo for introducing me to adaptive structures and to the discipline of topology optimization. I can not think of a research topic I would have enjoyed more.

I am also appreciative of the support that Kris Aber provided for various portions of this thesis.

Lastly, I am indebted to my wife and son who continually encouraged me and were an oasis throughout this whole process.

James R. Elgersma

## Table of Contents

	Page
Abstract . . . . .	iv
Acknowledgements . . . . .	vi
List of Figures . . . . .	ix
List of Tables . . . . .	xii
List of Abbreviations . . . . .	xiii
I. Introduction . . . . .	1
1.1 Morphing . . . . .	1
1.2 Optimization . . . . .	6
1.3 Problem Description . . . . .	9
1.4 Thesis Outline . . . . .	9
II. Literature Review . . . . .	10
2.1 Early Morphing . . . . .	10
2.2 Multidisciplinary Optimization of an In-plane Morphing Wing . . . . .	12
2.2.1 Single Configuration . . . . .	13
2.2.2 Multiple Configurations . . . . .	14
2.3 Flexible Skin Concepts . . . . .	16
2.3.1 One-Dimensional Morphing . . . . .	16
2.3.2 Skin Design Using Topology Optimization . . . . .	18
2.3.3 Flexible Skin Design Process . . . . .	20
2.4 Perch Landing Maneuvers . . . . .	23
2.5 Problem Statement . . . . .	24
III. Methodology . . . . .	25
3.1 Geometry Setup . . . . .	25
3.2 Finite Element Models . . . . .	31
3.2.1 Discretization . . . . .	32
3.2.2 Element Stiffness Matrix . . . . .	32
3.2.3 Global Stiffness Matrix . . . . .	50
3.2.4 Displacement Solution . . . . .	51
3.2.5 Element Stresses . . . . .	53
3.2.6 Validation . . . . .	54



	Page	
3.3	Aerodynamics . . . . .	63
3.3.1	Perching Maneuvers . . . . .	63
3.3.2	Force Estimation . . . . .	66
3.4	Optimization . . . . .	69
3.4.1	Compliance Objective Using the SIMP Model . . . . .	69
3.4.2	Optimality Criteria Method . . . . .	77
3.4.3	Parameter and Convergence Study . . . . .	81
IV.	Results . . . . .	89
4.1	Aerodynamic Results . . . . .	89
4.2	Structural Layouts . . . . .	96
4.3	Grid Independence . . . . .	112
V.	Conclusions . . . . .	114
5.1	Summary . . . . .	114
5.2	Recommendations . . . . .	116
5.3	Future Work . . . . .	118
Appendix A.	Derivation of General Quadrilateral Plate Bending Element Stiffness Matrix . . . . .	119
A.1	Definition of Moment/Curvature Relationship . . . . .	120
A.2	Element Selection and Displacement Definitions . . . . .	122
A.3	Definition of Curvature/Displacement Relationship . . . . .	128
A.4	Stiffness Matrix Formulation . . . . .	130
A.5	Gaussian Quadrature Method of Integration . . . . .	131
A.6	Computation of Stiffness Matrix . . . . .	133
Appendix B.	Finite Element MATLAB Codes . . . . .	134
Appendix C.	SIMP MATLAB Codes . . . . .	152
Appendix D.	Miscellaneous MATLAB Codes . . . . .	167
Bibliography	. . . . .	171

## *List of Figures*

Figure		Page
1.1.	Eagle Owl Soaring [4] . . . . .	3
1.2.	Eagle Owl Flyby [2] . . . . .	3
1.3.	Eagle Owl Perching [3] . . . . .	3
1.4.	Comparison of Fixed-Wing and Morphing Performances [20] . .	4
1.5.	Lockheed-Martin's Z-Wing [22] . . . . .	5
1.6.	NextGen's Batwing [22] . . . . .	6
1.7.	Generalized Topology Design Problem . . . . .	8
2.1.	MAK-10 with Telescoping Wing [5] . . . . .	11
2.2.	Bakshaev LIG-7 with Extendable Chordwise Panels [1] . . . . .	11
2.3.	Sample In-Plane Morphing Problem Definition . . . . .	14
2.4.	Cellular Flexible Skin Design [33] . . . . .	19
2.5.	Two-Step Skin Design Process . . . . .	21
2.6.	Typical Perching Maneuver Trajectory . . . . .	23
3.1.	Geometry Layout for Three Section Wing . . . . .	26
3.2.	Wing Geometry GUI . . . . .	26
3.3.	Example Wing Geometry . . . . .	27
3.4.	Birdwing in Forward Swept Configuration [m] . . . . .	28
3.5.	Birdwing in Zero Sweep Configuration [m] . . . . .	28
3.6.	Birdwing in Back Swept Configuration [m] . . . . .	29
3.7.	Birdwing in Dive Configuration [m] . . . . .	29
3.8.	General Quadrilateral Element Mapping to Computational Do- main [23] . . . . .	33
3.9.	Mapping of General Quadrilateral Element to Computational Domain . . . . .	40
3.10.	Natural Coordinate Unit Vectors . . . . .	49
3.11.	Computational Domain Indexing Example . . . . .	51

Figure		Page
3.12.	Computational Domain Indexing Example . . . . .	52
3.13.	Cantilever Beam Mesh . . . . .	55
3.14.	Von Mises Stress [Pa] of Cantilever Beam for Four Meshes . . . . .	57
3.15.	Bending Stress [Pa] of Cantilever Beam for 4 x 40 Membrane Mesh . . . . .	57
3.16.	Plate Bending Mesh . . . . .	58
3.17.	Stress Contours for Simply-Supported Bending Plate with Central Deflection . . . . .	60
3.18.	Membrane-Bending Model with Only a Membrane Load . . . . .	62
3.19.	Membrane-Bending Model with Only a Bending Load . . . . .	62
3.20.	Membrane-Bending Model with Combined Loading . . . . .	62
3.21.	Perching Maneuver: Altitude vs. Range . . . . .	63
3.22.	Perching Maneuver: Range vs. Time . . . . .	64
3.23.	Perching Maneuver: Attitude vs. Range . . . . .	64
3.24.	Perching Maneuver: Velocity vs. Range . . . . .	65
3.25.	Main Menu of Tornado Software . . . . .	66
3.26.	Example of Tornado Geometry Output [28] . . . . .	67
3.27.	Example of Tornado Vortex Panels Output [28] . . . . .	67
3.28.	Viscous Drag Estimation Curve . . . . .	69
3.29.	SIMP Penalization of Element Thickness . . . . .	74
3.30.	Example “SIMP” Output . . . . .	80
3.31.	Example “SIMPm” Output with No Checkerboard Filter . . . . .	80
3.32.	Reference Problem with a 60 x 30 Mesh . . . . .	81
3.33.	Membrane Structures for Reference Problem with Varying Penalizations ( $Vol_f = 0.5$ , $r_{filter} = 1.5$ ) . . . . .	82
3.34.	Membrane Structures for Reference Problem with Varying Volume Fraction ( $q = 3$ , $r_{filter} = 1.5$ ) . . . . .	83
3.35.	Membrane Structures of Reference Problem with Varying Filter Radii . . . . .	84

Figure		Page
3.36.	Membrane Structure of Reference Problem at Progressing Iterations . . . . .	85
4.1.	$\Delta C_p$ for Birdwing Geometries Along Perching Trajectory . . . . .	91
4.2.	Force [ $N$ ] for Birdwing in Swept Back Configuration at Point 1 . . . . .	92
4.3.	Force [ $N$ ] for Birdwing in Dive Configuration at Point 2 . . . . .	93
4.4.	Force [ $N$ ] for Birdwing in Zero Sweep Configuration at Point 3 . . . . .	94
4.5.	Force [ $N$ ] for Birdwing in Forward Swept Configuration at Point 4 . . . . .	95
4.6.	Membrane Structure for Birdwing at Point 1 . . . . .	97
4.7.	Bending Structure for Birdwing at Point 1 . . . . .	98
4.8.	Combined Structure for Birdwing at Point 1 . . . . .	99
4.9.	Combined Structure for Birdwing at Point 1 without Viscous Drag . . . . .	100
4.10.	Membrane Structure for Birdwing at Point 2 . . . . .	101
4.11.	Bending Structure for Birdwing at Point 2 . . . . .	102
4.12.	Combined Structure for Birdwing at Point 2 . . . . .	104
4.13.	Combined Structure for Birdwing at Point 2 without Viscous Drag . . . . .	105
4.14.	Membrane Structure for Birdwing at Point 3 . . . . .	106
4.15.	Bending Structure for Birdwing at Point 3 . . . . .	107
4.16.	Combined Structure for Birdwing at Point 3 . . . . .	108
4.17.	Combined Structure for Birdwing at Point 3 without Viscous Drag . . . . .	108
4.18.	Membrane Structure for Birdwing at Point 4 . . . . .	109
4.19.	Bending Structure for Birdwing at Point 4 . . . . .	110
4.20.	Combined Structure for Birdwing at Point 4 without Viscous Drag . . . . .	111
4.21.	Combined Structure for Birdwing at Point 3 with Coarse Mesh . . . . .	113
4.22.	Convergence Data for Figure 4.21 . . . . .	113
A.1.	Mapping of General Quadrilateral Element to Computational Domain . . . . .	123
A.2.	2 x 2 Gaussian Quadrature Evaluation of Stiffness Integral . . . . .	132

*List of Tables*

Table		Page
3.1.	Sectioned Geometry for Multiple Configuration Birdwing . . .	30
3.2.	End Deflection [mm] of Cantilever Beam . . . . .	56
3.3.	Data for Selected Points along Perching Maneuver . . . . .	65
3.4.	Convergence Data for Figure 3.36 . . . . .	86
3.5.	Variation of Move Limit for Figure 3.36 . . . . .	87
3.6.	Variation of Damping for Figure 3.36 . . . . .	88
4.1.	Aerodynamic Data for Birdwing Along Perching Trajectory . .	89

*List of Abbreviations*

Abbreviation		Page
DARPA	Defense Advanced Research Projects Agency . . . . .	1
MAS	Morphing Aircraft Structures . . . . .	1
AFRL	Air Force Research Laboratory . . . . .	1
UAV	unmanned aerial vehicle . . . . .	2
SMP	Shape Memory Polymer . . . . .	5
MAW	Mission Adaptive Wing . . . . .	12
AAW	Active Aeroelastic Wing . . . . .	12
FMC	Flexible Matrix Composite . . . . .	16
GUI	graphical user interface . . . . .	25
DOF	degree-of-freedom . . . . .	31
KKT	Karush-Kuhn Tucker . . . . .	75

# CONCEPTUAL LAYOUT OF WING STRUCTURE USING TOPOLOGY OPTIMIZATION FOR MORPHING MICRO AIR VEHICLES IN A PERCHING MANEUVER

## I. Introduction

### 1.1 *Morphing*

The ambiguous characterization of an aircraft as “morphing” could denote any one of sundry possible modifications to an aircraft’s structure during flight. Disregarding landing gear, incidence noses, and other geometry-changing devices, even further limiting the scope of the term “morphing” to that of the aircraft wing structure still lacks specificity. Whether the deforming wing-warping technique used on the Wright Flyer for roll control, the flaps found on nearly every modern day aircraft for the purposes of increasing lift and lowering stall speed, or the variable sweep wings of the F-111, all of these examples (and many others) are relatively simple modifications to the geometry that do not radically alter the optimal mission advantage. Additionally, wing shape-changing aircraft have been conceived and built since the inception of aviation, as will be surveyed in Chapter 2. But a more recent goal in this venue seeks to characterize morphing aircraft as those that can significantly modify their wing shape to adapt to multiple mission roles, hence developing high-performance in dissimilar flight regimes.

In 2003, the Defense Advanced Research Projects Agency (DARPA) began the 2 1/2 year Morphing Aircraft Structures (MAS) program in conjunction with the Air Force Research Laboratory’s (AFRL) Air Vehicles Directorate. The MAS program sought to design and build aircraft where, distinct from all of the historical morphing efforts of the past, morphing would entail radical, aerodynamically efficient shape change, thus transforming the mission of an aircraft. DARPA set forth the following description of a morphing aircraft: “a morphing aircraft is a multirole platform that

1. Changes its state substantially to adapt to changing mission environments.
2. Provides superior system capability not possible without reconfiguration.
3. Uses a design that integrates innovative combinations of advanced materials, actuators, flow controllers, and mechanisms to achieve the state change.” [45]

DARPA further elaborates that “the program envisions changing wing areas, wing spans, and other dimensions far more radically than before,” where “radical” describes a change on the order of 50%. Thus, merely rotating the sweep angle of the F-111 Aardvark further back does not fit the agenda of the MAS program. Rather, instead of solely rotating an otherwise fixed wing, morphing will take inspiration directly from nature, keeping in spirit with biomimicry, which ultimately led to the conception of aviation. A DARPA official elaborates, “We will develop a lightweight, actively controlled system of sensors (nerves), actuators (muscles), and structures (skin and bones) that mimic the ability of animals to adapt to widely changing environments and threats”. [45] Depending on the mission requirements, these efforts will likely convert military aircraft from large and heavy, piloted aircraft to relatively small, unmanned aerial vehicles (UAVs). That nature does in fact take advantage of drastic wing morphing through disparate flight maneuvers is piquantly and elegantly epitomized by the eagle owl in Figures 1.1–1.3, in which the planform shapes are outlined.

To justify the avant-garde undertaking of morphing, the potential to dramatically alter the aerodynamics and ultimately the nature of flight mission roles must be well based. Flight roles can be broken down into several different mission segments [20] consisting of takeoff, climb, cruise, acceleration, dash, endurance, and turn segment types. For these segments, performance metrics can be devised that evidence dependency on a few key parameters, namely thrust, weight, lift, drag, lift-to-drag ratio, and “best” velocities. Of the four principal forces acting on an aircraft, morphing of the wings will directly affect the lift and drag, and do nothing to alter the weight or thrust. Naturally adjusting the angle of attack will increase or decrease lift accordingly, so the aim of morphing is to modify the lift at a particular angle of





Figure 1.1: Eagle Owl Soaring [4]

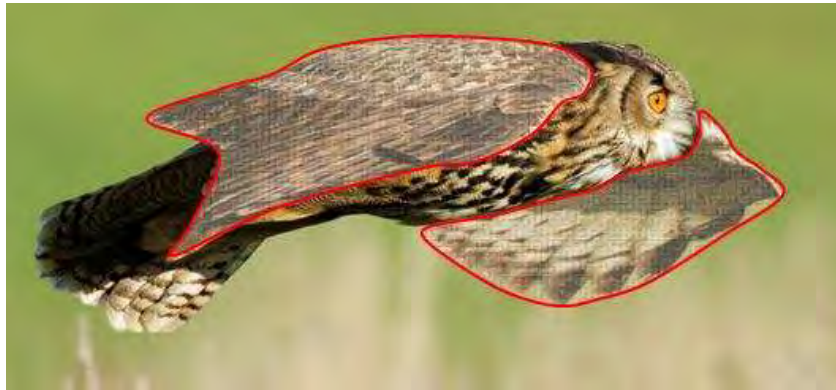


Figure 1.2: Eagle Owl Flyby [2]



Figure 1.3: Eagle Owl Perching [3]

attack. Modifying drag is crucial as lessening drag is a priority for both long range endurance missions and high speed maneuvers, and accumulating drag is paramount for rapid deceleration of a vehicle to a landing of near zero velocity. But producing a change in either lift or in drag should not be thought of as separate affairs, rather aerodynamic efficiency is influenced each time the lift-to-drag ratio is changed. The “best” velocity for a particular mission segment often depends directly on wing planform. An effective graphic juxtaposing conventional fixed geometry and morphing geometry is that of the spider plot shown in Figure 1.4. In the plot, the outermost radius of the plot is the best possible performance for a particular mission segment. On the legend, the “Firebee” represents the performance of a fixed-wing aircraft, the “airfoil” represents the same aircraft but with morphing airfoil capability, and the “geometry” series represents a study where wing area, wingspan, taper ratio, and sweep angle are free to vary independently of each other. Clearly a wing capable of telescoping, chord extension, and variable sweep behooves a multi-role platform.

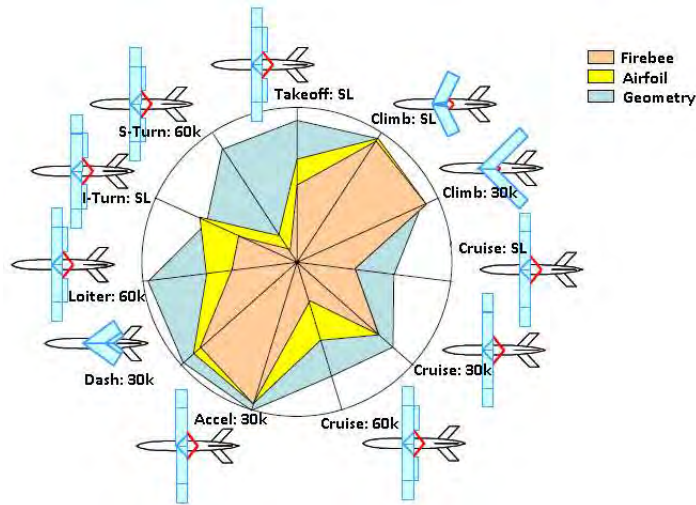


Figure 1.4: Comparison of Fixed-Wing and Morphing Performances [20]

Even a brief survey of aerodynamic dividends attainable by morphing aircraft promises a lucrative enterprise, but realizing morphing potential is a multifarious technical challenge. In an effort to resolve the many challenges, the first two phases of the MAS program sought to achieve four technical goals:

1. “Innovative, active wing structures that change shape;
2. Integration and aeronautical use of advanced sensors, skin and structure materials, internal mechanisms and distributed power sources;
3. Advanced capabilities for the military community;
4. Advanced shape-changing materials, efficient actuators and sophisticated, smart mechanisms.” [22]

According to the MAS program manager Terrence Weisshaar, morphing aircraft wings would ideally have at least a 50% area change, thus reconciling the reconnaissance mission requirement of a wide wingspan and large wing area with the combat requirement of minimum wing area for speeds of Mach 2 and 3. As the MAS program entered Phase III, two contractor teams, Lockheed-Martin Advanced Development Programs and NextGen Aeronautics, sought to demonstrate the advantageousness of morphing by comparing performance in both morphed and unmorphed configurations for climbing and turning maneuvers. Lockheed-Martin developed a UAV, dubbed “Z-Wing” (Figure 1.5), with two-position wings that essentially fold and tuck up alongside the fuselage, in effect, hiding a large part of the planform. After working around some billowing problems, Lockheed’s 1/8 inch thick skin, fabricated by Shape Memory Polymer (SMP) technology, was capable of stretching 100% upon stimulation of a current passing through it. Unlike the three-dimensional morphing of the Lockheed

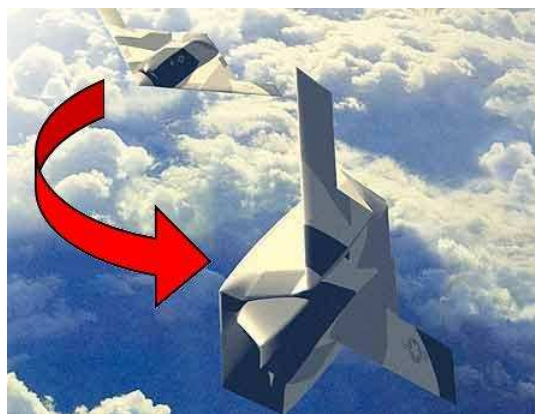


Figure 1.5: Lockheed-Martin’s Z-Wing [22]

UAV, NextGen’s morphing concept, called “Batwing” (Figure 1.6), consisted of in-plane shape change. With the same flexible skin challenge, NextGen achieved a 40% area change, 30% span change, and  $20^\circ$  change in sweep while successfully test flying a remotely piloted vehicle called the MFX-1. In light of the technical competencies hoped to be gleaned from the MAS program, progress with flexible skin, structural actuation, and suitable flight control technologies were realized.

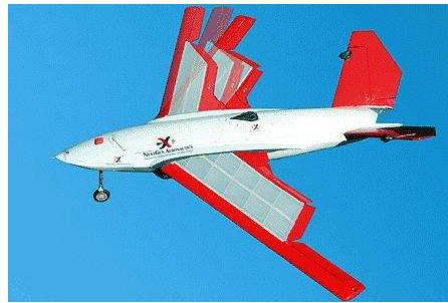


Figure 1.6: NextGen’s Batwing [22]

## 1.2 Optimization

As stated by the MAS program, the technical challenge of morphing wing design will involve the multidisciplinary integration of sensors, actuators, structures, mechanisms, and skins, a challenge far exceeding simple design of beams or four-bar linkages. Such a combinative problem is likely to have intricate solutions, not easily approachable on the basis of intuition alone. A promising design approach to a multifaceted problem is found in the character of optimization. Optimization allows a designer a direct way of stating any number of objectives—quantities to be minimized or maximized—and design variables—parameters that can assume a range of values throughout the optimization process. Then, once the designer has identified the underlying constraints (usually comprised of guiding engineering principles or simple geometric relationships), as well as governing bounds on the design variables (set by physical or other reasonable limitations), optimization solvers iterate through a set of equations numerous times until a stated objective converges within a change tolerance. With this manner of problem formulation and assuming the mechanics of a

problem have been correctly formulated, a designer focuses his attention on identifying the most desirable objectives to meet. Optimization is not without its difficulties, as challenges may lay in interpreting results or in the manufacturability of solutions, but it is certainly an advantageous tool to aid the morphing challenge.

In the emerging field of structural optimization, three types of optimization are defined based on the nature of the design variables.

1. *Sizing optimization*: The design domain and state variables are known *a priori* and are fixed. The design variables will represent some kind of structural thickness such as cross-sectional area or thickness distribution.
2. *Shape optimization*: In this case, the shape or contour of the domain now subsumes the design variables and is to be optimized. Neither the connectivity nor the boundary conditions of the structure is modified in the process.
3. *Topology optimization*: The most general of the three cases, topology optimization determines the features of a structure such as the connectivity of the domain, the number of members, and the position of holes. Topology optimization can be performed discretely or continuously.

All three of these structural optimization problems find application in aerospace technologies; however, the practicability of topology optimization is considered here.

Topology optimization seeks the optimal layout within a stated design domain (Figure 1.7). The prescribed quantities consist only of the loading and support conditions, the fraction of the design space to be filled with material (termed “volume fraction”), and any other desired design restrictions such as a requirement for a hole or to fill the boundary of the domain. The shape and connectivity are not prescribed and are determined from the optimization. The topology problem is called a distributed parameter system because the design variables represent a field or continuum with infinite degrees of freedom. Thus, formulating a calculable problem will require discretization of the field such that the field is comprised of a finite set of elements each with a finite number of degrees of freedom. The selected discretization scheme

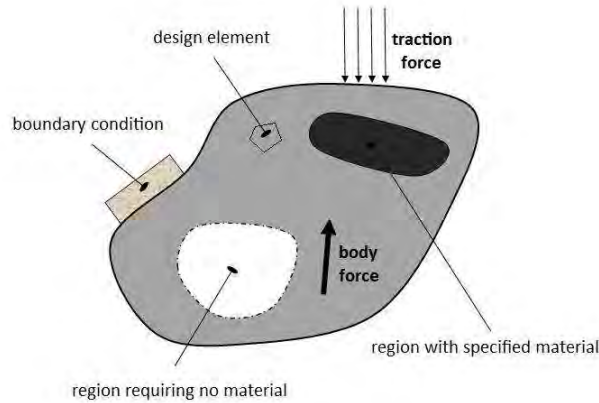


Figure 1.7: Generalized Topology Design Problem

will influence the resulting structure, and hence incites the enterprise of obtaining a grid-independent solution.

Though a continuum requires discretization, the distributed parameter system is not to be confused with a discrete parameter system which results when a structural problem only has a finite number of design variables. Such problems describe naturally discrete systems (i.e. the sizing of truss cross-sectional areas). Topology optimization is capable of driving towards a discrete or a continuous solution depending on the formulation, and either solution may be worthy of consideration, depending on how the structure is to be constructed or manufactured. While the objective function may encourage a discrete or a continuous solution, it is chiefly seeking to minimize a particular quantity. What that quantity represents is perhaps even more germane to the structural layout than is the discretization scheme. Of the many objective functions that can be employed, a natural starting point is that of compliance minimization, which is equivalent to stiffness maximization. Because a topological approach to design is very general (i.e. determination of the quantity and/or layout of beams, trusses, membranes, actuators, springs, pivot locations, etc.) and because an optimization problem can be formulated by an assortment of objective statements (i.e. compliance, strain or potential energy, fundamental eigenvalue, buckling load, etc.), topology optimization is highly practicable to morphing aircraft structures.

### ***1.3 Problem Description***

In view of the many alluring morphing challenges, the structural layout of morphing micro air vehicle (MAV) wings subjected to a gamut of aerodynamic loads as the MAV executes a perching maneuver will be investigated. A perching maneuver is a common landing scheme of birds in which they dive below a landing location, exchange kinetic energy for potential energy while pulling up to the perch, and then come to an abrupt vertical landing on the perch. The perching maneuver stimulates much interest because it is a miniature of a multi-role mission, in which an aircraft leaves a reference condition, such as a loiter wing configuration, “tucks” its wings into a dive configuration, and then flares them back out both to quickly ascend and to collect momentum-reducing drag. A few points along the perching trajectory will be extracted which are representative of the extremes of the maneuver. A three-dimensional estimation of the aerodynamic loads will be made and applied to the morphing wing shapes. By constructing a compliance optimization, conceptual wing layouts of the structure will be investigated for the dissimilar load cases.

### ***1.4 Thesis Outline***

This chapter has introduced the notion that morphing wings, engineered by sophisticated materials, sensors, actuation systems, and distributed power sources, will create multirole aircraft that can change the character of their mission mid-flight. Topology optimization is one discipline capable of shedding insight into a variety of structural problems that morphing wings will face. Chapter II will survey a brief history of morphing and then review current literature addressing a few aspects of morphing wing design. Chapter III will develop and validate a suitable finite element model for wings subjected to three-dimensional loading, and then formulate the minimum compliance optimization problem. Chapter IV will present the wing structural layout results from MATLAB simulations and discuss implications on wing structure at different design points. Conclusions and recommendations for future work will be laid out in Chapter V.

## II. Literature Review

To begin this chapter, a brief history of morphing aircraft is provided for the enrichment of the reader. Then the remainder of the chapter explores some of the current design concepts and methods that have been introduced as a means of addressing some of the technical goals posed by the MAS program. First, multidisciplinary optimization of the in-plane morphing wing will be reviewed. The multidisciplinary approach considers simultaneous mechanism, structure, and skin design for an adaptive wing. A baseline example will be first laid out, where only one planform shape is considered, followed by optimization that includes target shape-matching for multiple wing configurations. Next, some flexible skin design concepts will be described relating potential one-dimensional morphing skins, and then delineating the role of topology optimization in developing flexible skin. A two-step design process for synthesizing skins suitable for morphing is discussed. Finally, background behind perching maneuver trajectories is detailed.

### 2.1 *Early Morphing*

Through the 1920s and '30s, Geoffrey Hill designed a series of tailless aircraft dubbed the Westland-Hill Pterodactyl series. The fourth variant included variable wing sweep through a range of angles for longitudinal trim in the absence of elevators. In 1931, Ivan Makhonine, an expatriate Russian aircraft designer living in France, successfully flew the MAK-10 (Figure 2.1), his new telescoping wing concept. The MAK-10 could reduce its wingspan from 69 feet to 43 feet, which constituted a planform area decrease from 355 square feet to 204 square feet—a 42.5% area reduction. The improved MAK-123 completed its first flight in 1947, demonstrating with no adverse characteristics its extension-retraction system. In fact, the pilot was able to kill the engine, extend the wing tips and glide for an entire hour. A similar concept, the Gerin Varivol biplane (1936) had leading and trailing edges that could be unfurled from the fuselage out to the wingtips, greatly increasing the area of the otherwise thin wings. Also alike in nature, the Baksaev LIG-7 (Figure 2.2) was developed in the So-





Figure 2.1: MAK-10 with Telescoping Wing [5]

viet Union the following year with chordwise panels made of plywood that could be stowed entirely in the fuselage and then manually extended two-thirds of the way out along the wingspan. Though flight tests demonstrated successful retraction and extension of the panels, the fact that only 20% of the total drag was due to the wings severely limited the effect of morphing on performance. The Pterodactyl series used variable sweep primarily as a means for stability, but the MAK-10, the Varivol biplane, and the LIG-7 were all able to change the basic characterization of the aircraft by significantly increasing the aspect ratio of the wings for cruising conditions and reducing the aspect ratio for higher lift and subsequently slower landings. Another

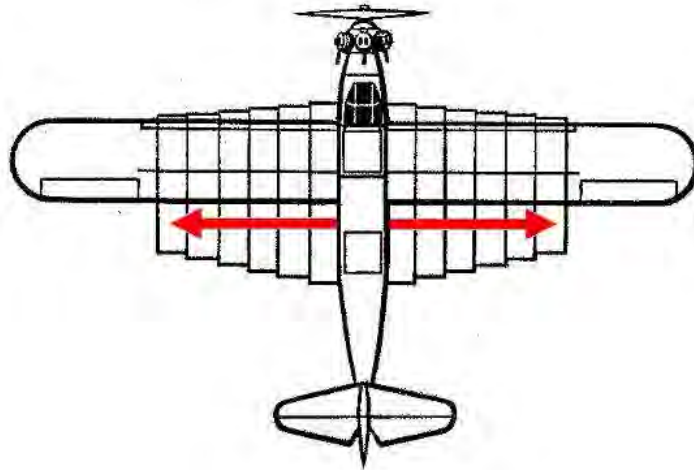


Figure 2.2: Bakshaev LIG-7 with Extendable Chordwise Panels [1]

example of an early morphology is that of the polymorphic Nikitin-Scheyevchyenko IS series also produced in the USSR just prior to WWII in 1938, which took advantage of both the climb capabilities of the biplane and, upon flipping the lower wing into the upper one, the speed of the monoplane. In light of all the early century innovative planform alteration ideas, the variable sweep wing concept certainly persisted throughout the 20th century leading to the development of the swing-wings of the experimental Bell X-5 (1951), and into production of the F-111 Aardvark (1964), the USN's F-14 Tomcat (1970), the Royal Air Force's Panavia Tornado (1979), and the strategic B-1 Lancer (1986).

Other interesting morphology has emerged in recent decades aside from two-dimensional planform changing devices. The F-8 Crusader (1955) had variable incidence wings in which the leading edges could tilt up during short takeoffs. The XB-70 Valkyrie (1964) had wing tips that folded down  $90^\circ$  to increase lift-to-drag ratio. Several aircraft possessed wings that took advantage of variable camber, such as the thin supercritical airfoil of NASA's X-29 (1984) on its forward swept wings and the flexible fiberglass leading and trailing edges of the AFTI/F-111 (1985) spurred on by the Mission Adaptive Wing (MAW) program. And in recent days, Active Aeroelastic Wing (AAW) Technology, as implemented on the Boeing X-53 (2006), in contrast to kinematically altering wing geometry, seeks to foster and utilize favorable aeroelastic wing twist to control and eliminate the phenomenon of "aileron reversal". This results in the benefits of reduced aerodynamic drag, increased control power, and increased design leeway with wing span, sweep, and thickness.

## ***2.2 Multidisciplinary Optimization of an In-plane Morphing Wing***

A discussion of morphing wing design efforts would be incomplete without giving consideration to addressing the integration of the entire adaptive structure, since adaptive structures are a multidisciplinary technology incorporating power systems, structures, mechanisms, and actuators. Maute and Reich [27] demonstrated that simultaneous optimization of mechanism layout, pivot point locations, and actuator

configurations has advantages over multi-step, decomposed procedures. Also, Joo et al. [19] demonstrated via experiment that a distributed actuation system is more conducive to increasing system efficiency for a relatively flexible scissor-like morphing wing mechanism than a single, stand-alone actuator. Therefore, it is necessary to synthesize rigid-body mechanisms to perform the shape change, structures to support the loading, and skins that do not inhibit the aerodynamics, in order to produce an efficient adaptive structure. To this end, an optimization formulation for a wing in a specified, single configuration is first discussed, followed by an optimization problem addressing a wing morphing through multiple configurations.

*2.2.1 Single Configuration.* A primal effort to optimize the topology of a three-layer model was undertaken by Joo and Sanders [18]. In order to achieve a prescribed motion, a truss element layer is intended to efficiently generate motion by virtue of rigid body rotation rather than through deforming the structure. A membrane layer represents the flexible skin and is bonded to a frame layer. A frame layer is added to support bending and torque caused by out-of-plane loads, and the combined frame and membrane layer is attached to the truss structure through springs at locations to be determined. The intended motion of the wing was to return from a swept position to a position of no sweep; however, a single point at the leading edge of the tip chord was directed in a chordwise path to simplify the problem. The multi-objective optimization problem was then formulated as a weighting of mutual potential energy of the point and strain energy stored in the structure. The mutual potential energy was to be maximized in order to produce the maximum work output while morphing, and the strain energy was to be minimized for the purposes of efficient morphing. The results of the study revealed effective combinations of structure and mechanism for both linear and nonlinear synthesis, as well as syntheses with and without the frame layer. Joo and Sanders also compared by analytical means the bending and membrane stiffness sensitivities of the entire wing, modeled as a sandwich beam, to the skin thickness. The total membrane, or in-plane, stiffness of the wing is

more sensitive than the total bending stiffness of the wing to changes in skin thickness. Thus an anisotropic engineered skin that is flexurally stiff, but flexible in-plane, is desirable in order to avoid thickening the skin for prevention of out-of-plane deflection (which would in turn produce inefficient in-plane morphing). However, the available skin model was simply a membrane model that did not include the capability of responding to bending loads, and bending loads were therefore not considered.

*2.2.2 Multiple Configurations.* An important aspect of the morphing mechanism involves determining the kinematics that will transform one planform shape into another. Emphasizing the formulation of the combined mechanization, actuator placement, and topology optimization problem, Inoyama et al. formulated objective functions seeking to minimize actuator usage and shape-matching error terms. For simplicity, point-matching was utilized to achieve shape change, rather than function-matching [24]. A sample in-plane morphing wing model with semi-ground connectivity and consisting of joint and line elements representative of their problem of interest is shown in Figure 2.3.

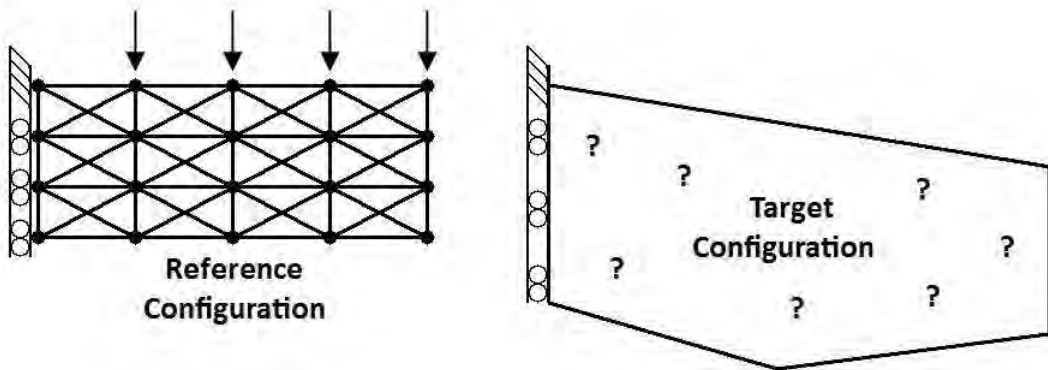


Figure 2.3: Sample In-Plane Morphing Problem Definition

In their model [14], the state of both the line elements and the joint elements were considered design variables and had to be determined by the optimization process. Elements could be designated as “soft” morphing elements, “stiff” non-morphing

load-bearing elements, or elements could be considered “voids” and removed from the structure. Line elements could be selected as telescoping members, actuators, or frame elements, and joint elements could be selected as either revolute or semi-rigid joints. The frame elements were conventional FE beams with both axial and bending stiffness. Telescoping members had relatively low axial stiffness, but they did have bending stiffness. The actuators behaved like telescoping two-force members. Additional constraints were applied to limit the stroke of the actuators and telescoping members. Whenever the optimization yielded “very low” stiffness for all of an element’s degrees-of-freedom, then the element was mechanically non-contributing and considered “void”. Revolute and semi-rigid joints differ in that revolute joints had relatively low rotational stiffness, whereas semi-rigid joints had relatively high rotational stiffness. Both joints had relatively high stiffness in all other directions. The model also necessarily contained flexible skin. However, the skin was represented simply by four-node rectangular membrane elements that did not support bending stiffness. Although out-of-plane schemes that can handle aerodynamic pressure loads were wanting, the methodology did demonstrate that a multidisciplinary approach is a viable solution. Further refinements of the methodology were made by Inoyama et al. in Reference [15]. Inoyama et al. [16] also exhibited the methodology for three configurations—a reference loiter configuration with  $15^\circ$  of sweep, a high-lift configuration with an 84% area increase, and a climb configuration with a  $30^\circ$  sweep angle and 14% area increase. Improvements were made accounting for actuator distribution control, external load dependency, morphing reversibility, and modified relative volume constraint.

Aside from the aforementioned investigations, other examples of utilizing topology optimization for more comprehensive wing design have been demonstrated. Maute and Allen [26] used material topology optimization to layout the geometry of a wing’s internal structure for conceptual design of aeroelastic structures with coupled fluid-structure interaction. Two different numerical examples were provided that determined three-dimensional wing stiffener layout, to then be interpreted as spars and

ribs. The first example was a three-layered plate model that essentially determined material layout from an in-plane or top plane view, and the second example began with a three-dimensional spar-rib design domain. Maute and Reich [27] studied three numerical experiments using topology optimization for shape-adaptive wings. The objective of the optimization was to maximize lift-to-drag ratio and hence determine mechanism and actuation system layouts.

### ***2.3 Flexible Skin Concepts***

Flexible skin that will cover morphing wings is certainly not least in the technical challenges faced by adaptive structures. Although some discussion of flexible skin design has already been mentioned in a multidisciplinary approach, the acclivity flexible skin design must climb merits its own separate discussion apart from the wholistic problem. Two general classifications of morphing structures with flexible skin implications can be distinguished: structures with sliding or rotating surfaces (such as Lockheed's Z-Wing) and structures with stretching surfaces (such as the NextGen's Batwing). The literature reviewed here considers skin suitable for in-plane stretching. Flexible skin for morphing wings will require two principal mechanical characteristics: low in-plane stiffness and high out-of-plane bending stiffness. Low in-plane stiffness is needed so that the skin can undergo large strains (on the order of at least 50%) vital to significant planform changes, and is desirable for minimizing the actuation energy required to achieve shape change. But the skin must also be very flexurally rigid such that it can support pressure loads normal to the planform without aerodynamically adverse deformation, such as billowing or even simply tearing. These generally are competing requirements and foster a substantial design challenge.

*2.3.1 One-Dimensional Morphing.* A few morphing applications only require distortion in one direction, such as changes in span, chord, or camber. Murray et al. [30] propose a Flexible Matrix Composite (FMC) as a solution. FMCs are composed of stiff fibers aligned and embedded in a soft matrix material capable of high

strain. Aligning the matrix-dominated direction with the direction of morphing and consequently the fiber-dominated direction with the non-morphing direction makes FMCs plausible for one-dimensional morphing. Thus, the fibers would be aligned chordwise for span morphing, and spanwise for chord or camber morphing. Both a theoretical model predicted and experimentation validated that a FMC can simultaneously have a high fiber modulus and a low matrix modulus. However, a FMC with low matrix modulus and high strain capability displays membraneous behavior, possessing virtually no inherent flexural stiffness. One way to attain flexural rigidity without adding stiffness in the matrix-dominated direction is by pretensioning the skin in the fiber-dominated direction. Tensioning a membrane will necessarily reduce the out-of-plane deflection, but only the fiber-dominated direction should be pretensioned for a couple reasons. Since the matrix-dominated direction is the morphing direction, pretension levels would vary as the wing morphs, were they applied in that direction. Secondly, the matrix modulus is quite low and the skin would likely rupture under the necessary pretension loads. Also confirmed by experimentation, the midpoint deflection (greatest deflection) of the FMC flexible skin panel has little sensitivity to the matrix modulus, and pretension has a much greater impact on reducing out-of-plane deflection than does the matrix modulus.

As using pretension to reduce out-of-plane deflection suggests, one interesting study from Song et al. [41], inspired by flying and gliding mammals such as bats and marsupial gliders, demonstrated excellent correlation between camber and so-called “Weber number” for various pretension levels. The simple theory employed represented static aeroelastic deformation of a compliant membrane by balancing aerodynamic pressure load generated by airfoil shape with the tension in the membrane. The Weber number is defined as the force normal to the chord and normalized by Young’s modulus and skin thickness. A parabola was selected as the deformed shape assumed by a membrane under such loading, which is the exact solution for a membrane subjected to uniform loading. The results of experimentation revealed that camber does indeed rise rapidly for increasing Weber number when no pretension is applied. With

higher pretensions, however, the relationship between camber and Weber number becomes nearly linear, indicating that camber is less dependent on Weber number, and the camber values are much smaller. This study confirms intuition that for a given aerodynamic load, increasing the Young’s modulus or the thickness of the skin (i.e. decreasing the Weber number for a given force) lessens the out-of-plane deflection and consequently the camber. This study of membrane wing aeromechanics further demonstrates that Young’s modulus and thickness are important design variables for the flexibility of skin.

*2.3.2 Skin Design Using Topology Optimization.* In contrast to using FMCs or shape memory polymers (SMPs) [9] (as in the case of Lockheed’s Z-Wing), cellular-based structures are yet another alternative to flexible skin fabrication. Olympio and Gandhi [35] used topology optimization to determine the best material distribution for cellular structures with voids, or very light filling. Both in-plane one-dimensional morphing and shear-compression morphing cases were conducted. The constituents of the skin considered were those of a topologically undetermined cellular core sandwiched between thin face sheets. The material comprising the core must possess high bending stiffness such that it can withstand out-of-plane loads, and the topology of the core must be capable of high in-plane strain such that the wings can morph with relatively little energy. The thin face sheet simply provides a smooth surface for air-flow, and is not intended to add any bending stiffness; its effect on in-plane morphing strain was not considered (Figure 2.4). The following four quantities were included in the multi-objective minimization problem: material volume, out-of-plane deformation, work required to morph the wing, and the ratio of the maximum local strain to maximum global strain of the skin (i.e. inverse of strain amplification). Minimizing the ratio of local to global maximum strain ensures that the local stresses remain lower than the yield stress of the material, and ultimately that morphing is reversible. The solutions determined by the optimization represent periodic units, or miniature panels that form a skin like a plate made of an orthotropic material when assembled.



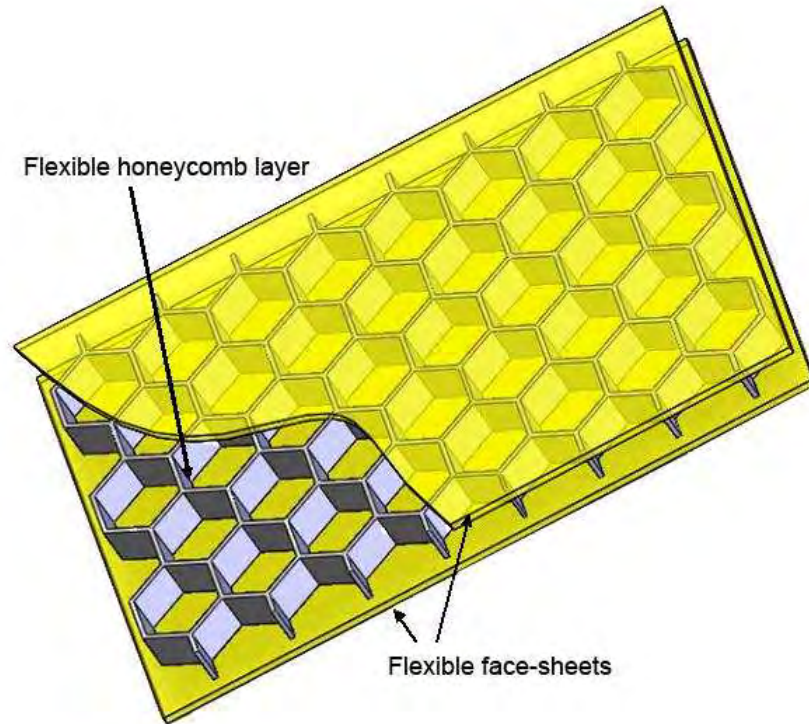


Figure 2.4: Cellular Flexible Skin Design [33]

Several core layouts resulted from the topology optimization performed by Olympio and Gandhi, due to using a Genetic Algorithm for the optimization solver which retains multiple “best” solutions. For the one-dimensional morphing case, three different groupings of solutions each with advantages and disadvantages were observed—plain sheet of material, honeycomb-like topologies, and horizontal fibers—although 43 different topologies were retained by the algorithm. The plain sheet of material was the heaviest but minimized out-of-plane deflection the most. The horizontal fiber solutions were the lightest and possessed the characteristic of not occupying the entire design space in the non-morphing direction, and hence had a Poisson’s ratio of zero [32]. Perhaps the best compromise was that of the honeycomb-like topologies with median weight and a strain amplification of 6.5. The solutions for the shear-compression cases all contained the same groupings with the addition of a few solutions that were a combination of honeycomb and fiber cells. Unlike one-dimensional morphing, the long, vertical fiber solutions require less work to morph and

greater strain amplification. No one topology prevailed, and several of the solutions may be valuable, depending on the application.

Several other variant features of the baseline cellular honeycomb core were studied and compared [34]. A filled honeycomb without a face sheet is one way to support out-of-plane loads, but it significantly increases the needed actuation energy and the weight of the wing, which ultimately might cause it to be noncompetitive compared to other designs. Also, any type of fill could create voids or leakage during wing morphing. Face sheets bonded to the core will also require higher deformation energy and could also be subject to tearing. In the absence of adequate bending stiffness and to avoid wrinkling, face sheets may need to be pre-strained to a greater extent than they can accommodate without tearing. Another alternative skin design is to add one discrete scale to each honeycomb. This successfully avoids an increase in the energy requirement but could adversely affect the aerodynamics. One final variant returns to the idea of sandwiching a voided honeycomb core between two face sheets, but instead of bonding the face sheets to the core, the two face sheets are connected to each other by elements running through the core. The advantages of a single encasement of the core is that the core would deform independently of the sheets. Thus, the face sheets would not stiffen the core, yet they would bear the aerodynamic loads through contact with the core.

*2.3.3 Flexible Skin Design Process.* This section recapitulates a two-step design process developed by Joo et al. [17] for engineering a skin suited to meet a target morphology of an adaptive wing. The design concept stems from the fact that a morphing mechanism likely has varying motion requirements along the planform. In contrast, for a purely telescopic wing, there is only one motion requirement and an engineered skin approach may be unnecessary. Thus, a distribution of flexibility requirements led to the decision to develop a single-layer engineered skin with distributed advanced materials, such as elastomers and SMPs. The first step of the method is to determine the material requirements for each “patch” of skin. The num-

ber of patches into which the planform is discretized can be driven by the extent to which the kinematics vary throughout the wing. A relatively straightforward change in planform may result in a mechanism that needs only 10-20 patches to capture the range of motion requirements; whereas more complicated planform changes may require more like 50-100 patches. In the second step, the microstructure of each patch is determined by decomposing the patch into a pattern based on a unique, repeating “unit cell” (Figure 2.5). Therefore, a topographically unique cell will have to be determined for each patch, as well as the number of repeating cells necessary to realize the constitutive properties of the patch.

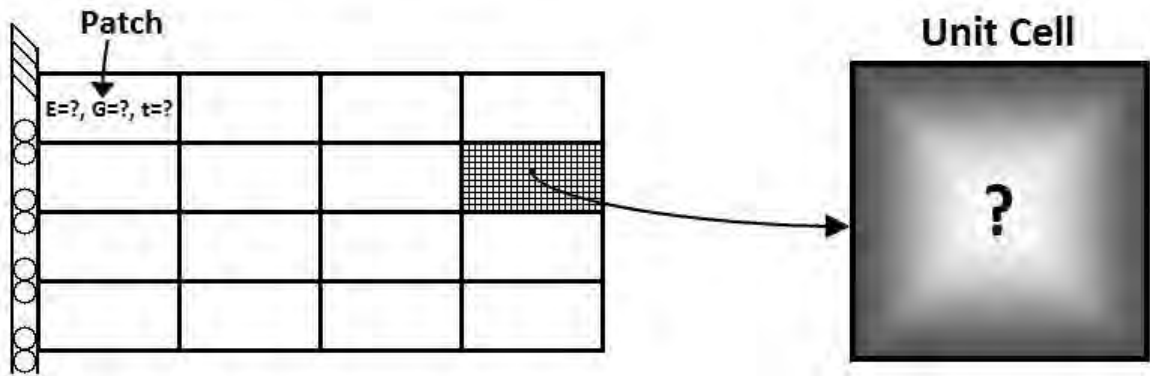


Figure 2.5: Two-Step Skin Design Process

The overall combined wing design follows from the wing design discussed in Section 2.2.1, where a mechanism layer is sandwiched between two thin skin layers. The mechanism layer attaches to the skin layer by virtue of “pivots”. In the optimization process, these pivots are represented by variable stiffness connector elements that when “active” indicates rigid attachment of mechanism to skin, and when “void” implies that the skin slides freely over the mechanism layer. Thus the skin patch deformation does not necessarily correspond to the mechanism reference cell (Figure 2.3). If the known mechanism was determined with a uniform skin, the mechanism optimization can be repeated once a distributed skin is determined. One can iterate through the two processes as needed.

To develop the material requirements for each patch of skin, design variables are chosen to represent any or all of the following: Young's modulus, shear modulus, element thickness, and Poisson's ratio. Additionally, ply thickness, orientation angle, and extra Young's moduli can be included as design variables if a laminated composite is the intended material. With these variables, the optimization has indirect control over the constitutive matrix and can independently vary bending and membrane stiffness through the effects of combining the constitutive properties. A variant of the SIMP method [8] is then used to distribute the material properties across the wing. Rather than an exponential variation of the design variables across the design space, a linear one is employed so as not to drive the variables towards discrete 1 and 0 values, since skin is obviously required to cover the entire wing. The objective function is a minimization of the target displacements of the skin from the realized ones.

After the material requirements for the skin patches has been set, step two of the skin design process is to optimally lay out given materials in a multi-phase model. The unit cell is discretized into a large number of finite elements. Although two-dimensional membrane elements were used, three-dimensional elements were recommended after comparing with the skin fabrication; because in the discretization, the in-plane dimensions were actually smaller than the thickness of the element. Using the homogenization method of elastic materials, the contribution of element stiffness to global stiffness is determined by the density of each element, and hence the SIMP method is again employed. Rather than developing a cell with voids, the SIMP method is instead used to determine two constitutive matrices, which then represent two phases of a SMP material before and after the glass transition temperature. Thus, in order to physically realize the material, heating elements will have to be placed within the cell to change the material properties. The optimization objective is simply to match the target constitutive matrix.

## 2.4 Perch Landing Maneuvers

Section 1.3 described the perch landing maneuver, which involves conveying a flight vehicle to a specified location with near-zero final velocity, to be absorbed upon impact. Seigler et al. [40] derived flight trajectories commonly observed in nature by minimizing actuation force required at low speeds in a basic optimal control problem. The typical trajectory profile (Figure 2.6) is made up of an initially descending movement that undershoots the target, and then a much quicker flare segment where the flight vehicle exchanges its gained kinetic energy for potential energy and ascends to the target destination. The basic optimal control formulation contrived by Seigler

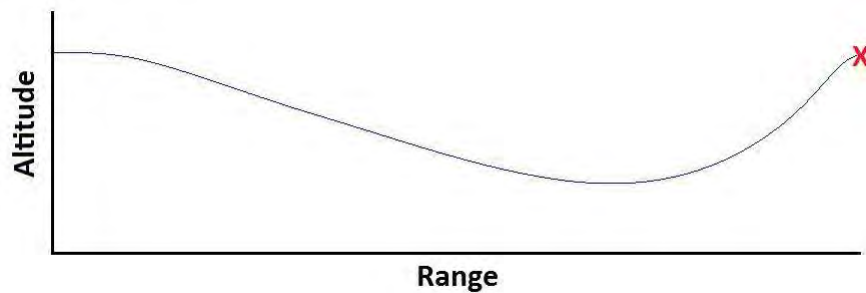


Figure 2.6: Typical Perching Maneuver Trajectory

et al. uses an exponential function in the performance measure to penalize required actuation force. The choice of exponent for the exponential controls the penalization of the low velocities. The rationale for penalizing low velocities stems from the fact that aircraft must exert the greatest effort at low speeds to account for lift no longer generated by higher speeds. The results of the problem reveal that the length of the descending segment increases for greater initial downward velocities, and consequently the length of the ascending segment decreases, intensifying the flare.

Seigler et al. [40] then simulated, with longitudinal flight dynamics, a trajectory analysis using a MAV with similar sizing and planform properties as a pigeon. Thrust was zeroed out to represent gliding flight. Three models were considered in the flight simulations: (1) fixed wings with an elevator as the only actuator, (2) rotating wings with a quasi-steady aerodynamic model [25], and (3) rotating wings with

a rate-dependent aerodynamic model. All three models matched the general concept of acquiring kinetic energy and then exchanging it for potential energy. However, the profiles between the three varied significantly depending on the penalization weighting of the actuators. In effect, the rate-dependent simulation predicted more lift generation by virtue of rotating the wings at a fast rate. The differences in trajectories point to the importance of considering non-steady wing rotation effects that increase lift generation capability. Robertson et al. further refined the aerodynamic model used to generate the trajectories by developing a vortex particle method, which showed good agreement with a CFD model as well as wind tunnel testing [39].

In this study, the trajectories were optimized to minimize landing velocity, and again the dynamic aerodynamics predicted a slower landing velocity and impact energy than did the static aerodynamics. Further transient post-stall aerodynamics are investigated by Reich et al. [37].

## ***2.5 Problem Statement***

Evident in all of the literature previously examined, a need for three-dimensional finite element modeling was clearly identified, such that the effect of out-of-plane bending loads on morphing wing structures can be included in design considerations. Both the multidisciplinary structure, mechanism, and skin optimization problems as well as the skin design process lacked out-of-plane loading. Thus a finite element model consisting of continuous elements (i.e. membrane and bending plate elements) is developed in Chapter 3. The intention of the finite element formulation is to be able to model the wing structure as a whole, and to be able to model the flexible skin specifically, once additional finite element models of truss and beam elements representing the structural layer of the wing are added to form a combined model. However, the former is the basis for the compliance optimization objective formulated in Chapter 3, and thus conceptual layout of wing structures are explored in Chapter 4. The out-of-plane loads imposed on the finite element model will be an estimation of the aerodynamic loads experienced by a MAV during a perching maneuver.

### III. Methodology

This chapter details the theoretical methodology leading to the development of the MATLAB codes found in Appendix B and Appendix C, as well as the results of Chapter 4. The initial task was to establish a geometry generation method, and select a few baseline geometries. Next the development and derivation of the finite element models needed to solve the equilibrium constraint in the optimization is described. A perching trajectory with corresponding flight data is then chosen, and the process for estimating the resulting aerodynamic loads to be applied to the FE model is related. Finally, the compliance minimization objective using the SIMP model is formulated, and the OC method used to solve the optimization problem is described.

#### 3.1 *Geometry Setup*

As previously mentioned in the Problem Description (Section 1.3), the scope of the current investigation is limited to planform-changing, two-dimensional morphing, akin to NextGen’s Batwing. In order to use simple bilinear shape functions to describe the geometry of the finite elements, planform shapes are framed by straight lines. Additionally, the wing is comprised of any number of trapezoidal sections. This restriction ensures that the leading and trailing edges have the same number of straight-edged line segments, and that the tip chord is parallel to the root chord. It is feasible for any geometry that meets this criteria to be discretized by a structured mesh, though meeting the criteria does not guarantee that the geometry will be conducive to a structured mesh. For instance, a wing with a near-zero taper for its last section will have very small cells at the tip, which may lead to bad results in both the finite element analysis and the aerodynamic analysis.

A MATLAB script [6] was developed to generate meshes for such geometries, and Figure 3.1 displays a three-sectioned, general wing planform in terms of the script’s input variables. The code provides a graphical user interface (GUI) for easy creation of a wing in any number of configurations (Figure 3.2). Once the user is content with his geometry, he can save the geometry into a .mat file format, which

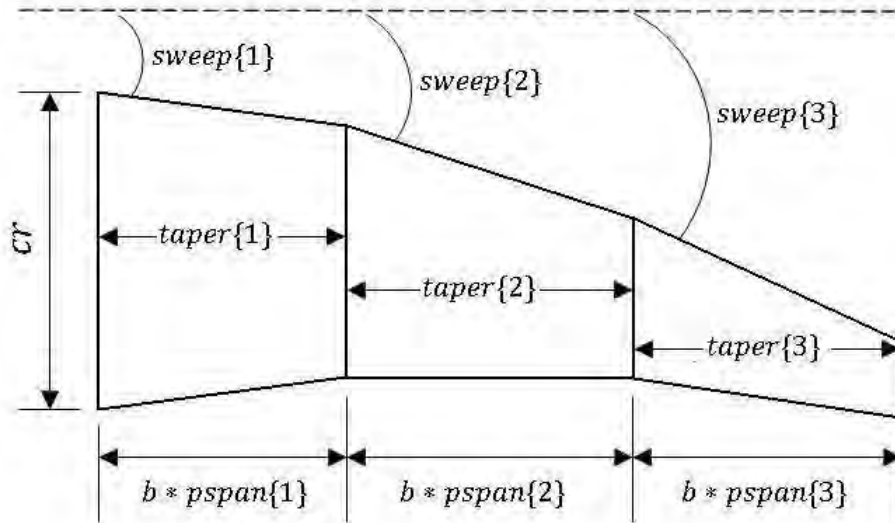


Figure 3.1: Geometry Layout for Three Section Wing

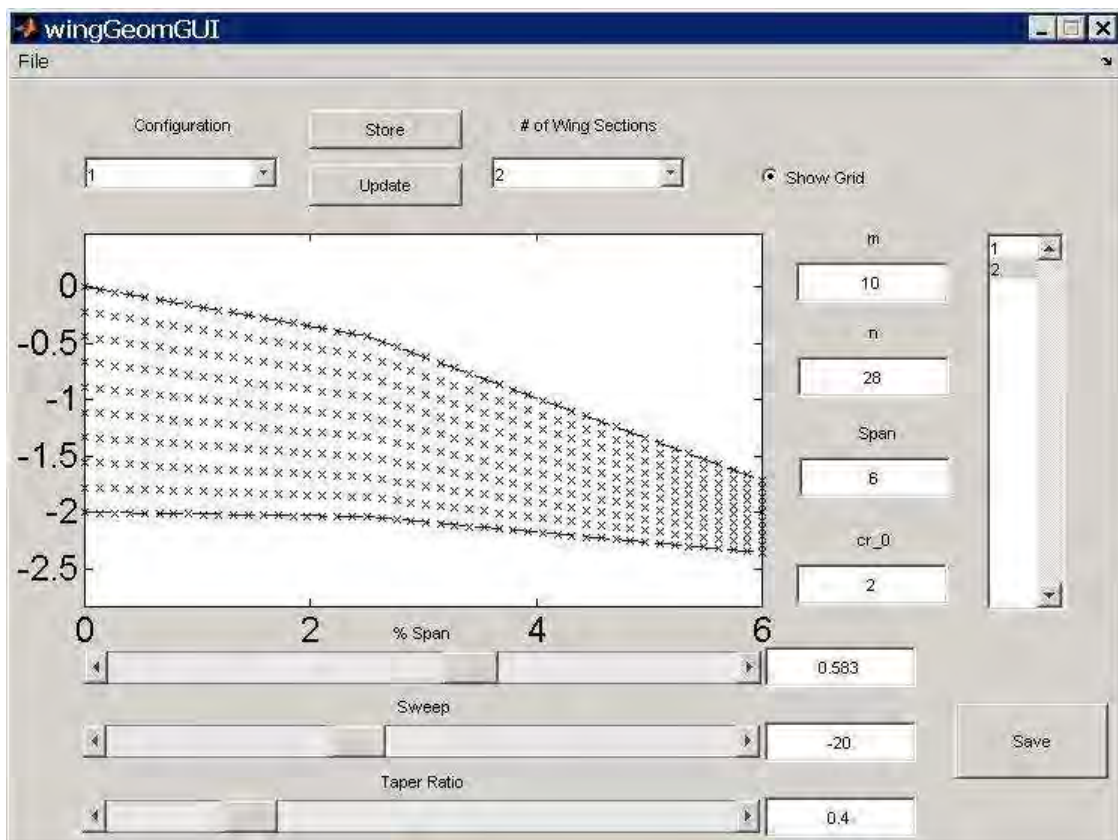


Figure 3.2: Wing Geometry GUI



consists of a structure “params” containing the input variables and the mesh. For each wing configuration, the input variables define a root chord and span for the entire wing, and then a taper ratio, sweep angle, and percent of the total span for each wing section. The mesh is created by linear interpolation in two directions with the number of grid points in both directions specified by the user. After initial development of the script, the upper profiles of a NACA 4-series airfoil and a reflexed airfoil were hardcoded into the script to add camber to otherwise flat wings. A script for viewing the wing mesh is given in Appendix D.1. An example of a two-sectioned wing with camber generated by the GUI and viewed by the script is shown in Figure 3.3.

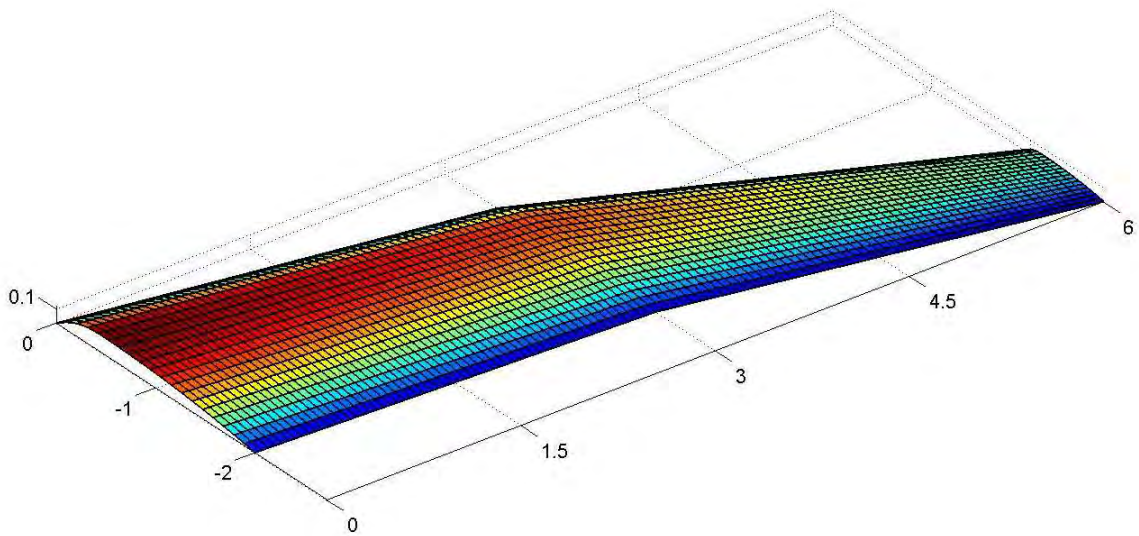


Figure 3.3: Example Wing Geometry

In order to capture similar planform configurations as the eagle owl (shown in Figures 1.1–1.3) is capable of morphing through, a rotating mechanism is envisioned that can sweep about  $5^\circ$  forward during a flare, and then sweep back roughly  $55^\circ$ – $60^\circ$  during a diving dash. The trapezoidal partitions and generated meshes for four configurations of this fictitious geometry (dubbed “birdwing”) are shown in Figures 3.4–3.7. Table 3.1 gives the geometry parameters that define the wing configurations. In later analysis, simple rectangular wings with and without sweep are also used for baseline comparison to the birdwing.

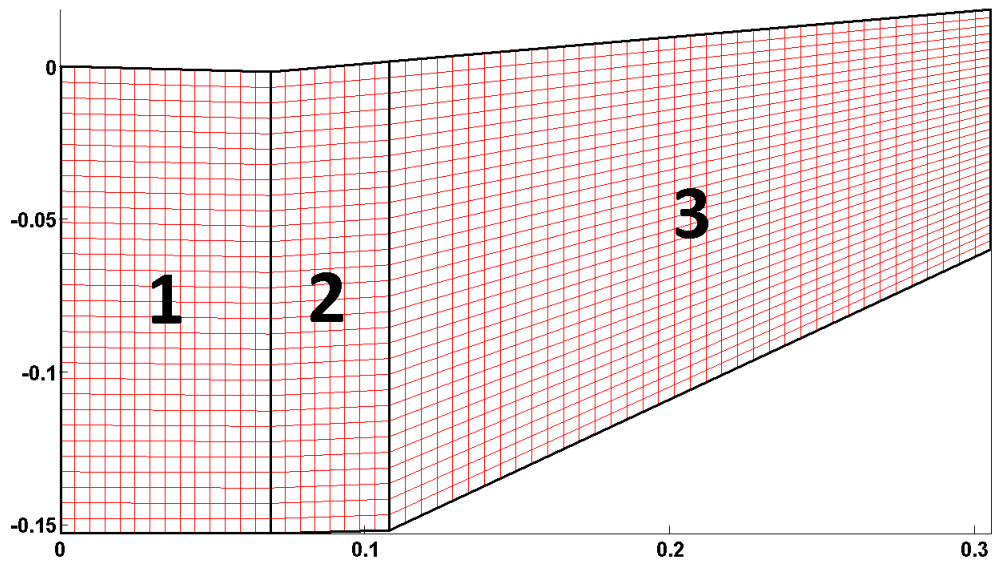


Figure 3.4: Birdwing in Forward Swept Configuration [m]

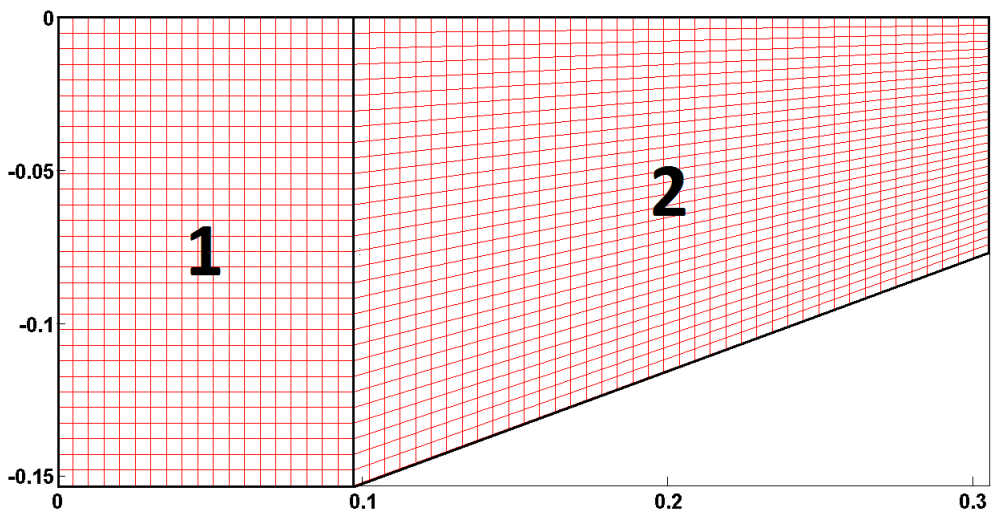


Figure 3.5: Birdwing in Zero Sweep Configuration [m]

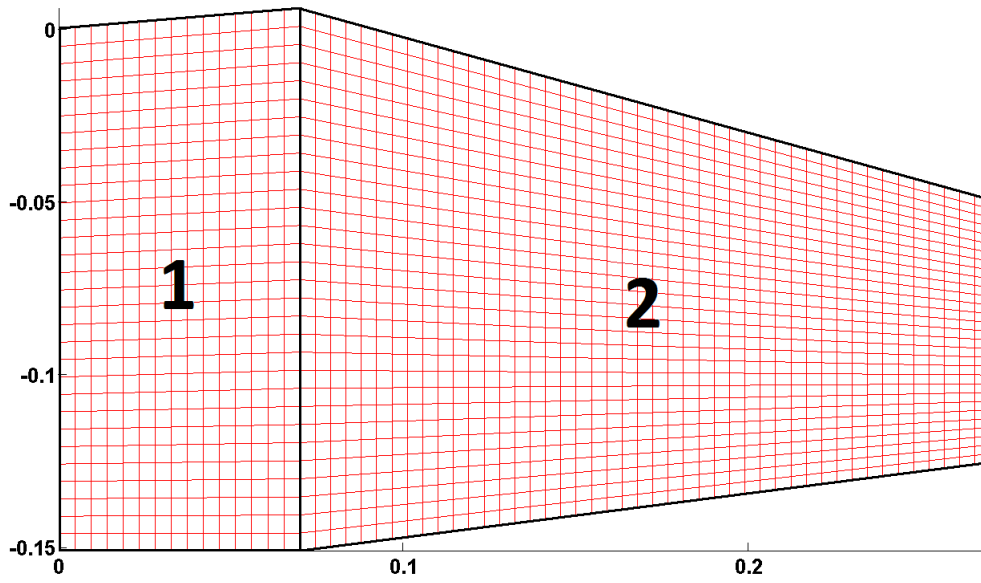


Figure 3.6: Birdwing in Back Swept Configuration [m]

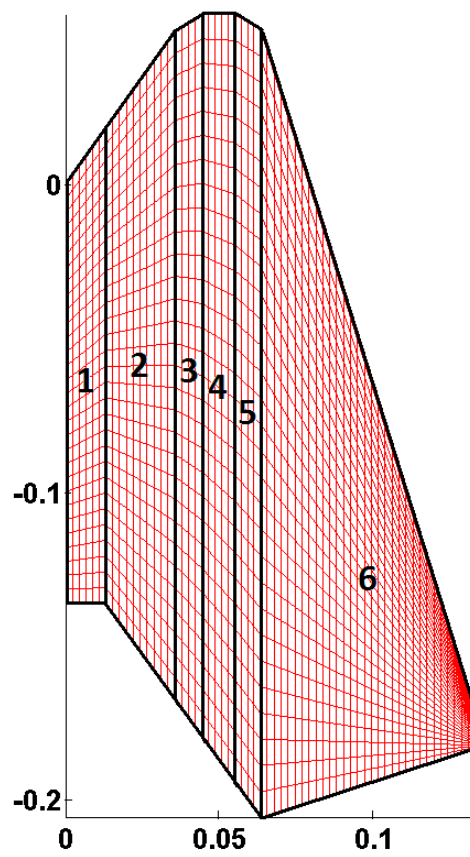


Figure 3.7: Birdwing in Dive Configuration [m]

Table 3.1: Sectioned Geometry for Multiple Configuration Birdwing

<b>Configuration:</b>			<b>5° Forward</b>	<b>0° Sweep</b>	<b>15° Back</b>	<b>Dive</b>
Partition #1	<i>Chord</i>	[m]	0.153	0.153	0.151	0.136
	<i>Span (%)</i>	[m]	0.069 (22.6)	0.097 (31.8)	0.070 (25.9)	0.013 (9.7)
	<i>Taper</i>	[—]	0.988	1.000	1.040	1.135
	<i>Sweep</i>	[°]	-1.56	0	4.90	54.74
	<i>QC Sweep*</i>	[°]	-1.17	0	3.68	46.69
	<i>Panels**</i>	[#]	14	19	15	6
Partition #2	<i>Chord</i>	[m]	0.152	0.153	0.157	0.154
	<i>Span (%)</i>	[m]	0.039 (12.8)	0.208 (68.2)	0.200 (74.1)	0.023 (16.9)
	<i>Taper</i>	[—]	1.014	0.501	0.483	1.408
	<i>Sweep</i>	[°]	4.97	0	-15.50	54.74
	<i>QC Sweep</i>	[°]	4.19	5.24	-9.95	35.63
	<i>Panels</i>	[#]	8	41	45	10
Partition #3	<i>Chord</i>	[m]	0.154	—	—	0.217
	<i>Span (%)</i>	[m]	0.197 (64.6)	—	—	0.009 (6.7)
	<i>Taper</i>	[—]	0.510	—	—	1.081
	<i>Sweep</i>	[°]	4.97	—	—	30.89
	<i>QC Sweep</i>	[°]	10.35	—	—	5.97
	<i>Panels</i>	[#]	38	—	—	4
Partition #4	<i>Chord</i>	[m]	—	—	—	0.234
	<i>Span (%)</i>	[m]	—	—	—	0.011 (8.0)
	<i>Taper</i>	[—]	—	—	—	1.063
	<i>Sweep</i>	[°]	—	—	—	0
	<i>QC Sweep</i>	[°]	—	—	—	-18.99
	<i>Panels</i>	[#]	—	—	—	5
Partition #5	<i>Chord</i>	[m]	—	—	—	0.249
	<i>Span (%)</i>	[m]	—	—	—	0.008 (6.2)
	<i>Taper</i>	[—]	—	—	—	1.027
	<i>Sweep</i>	[°]	—	—	—	-30.35
	<i>QC Sweep</i>	[°]	—	—	—	-38.07
	<i>Panels</i>	[#]	—	—	—	4
Partition #6	<i>Chord</i>	[m]	—	—	—	0.256
	<i>Span (%)</i>	[m]	—	—	—	0.070 (52.5)
	<i>Taper</i>	[—]	—	—	—	0.040
	<i>Sweep</i>	[°]	—	—	—	-72.65
	<i>QC Sweep</i>	[°]	—	—	—	-66.73
	<i>Panels</i>	[#]	—	—	—	31
<b>Total Span</b>			0.305	0.305	0.270	0.133

\**QC Sweep* is the quarter-chord sweep\*\**Panels* is the number of spanwise panels

### **3.2 Finite Element Models**

As was brought to attention in the literature review, previous investigations by Joo et al. [17, 18] and Inoyama et al. [14–16] lacked out-of-plane loading capability in their finite element models. Hence the goal of this section is to develop a six degree-of-freedom (DOF) model that can support membrane (in-plane) and bending (out-of-plane) loads and be easily implemented in MATLAB. The approach undertaken is to first develop membrane (Section 3.2.2.1) and bending (Section 3.2.2.2) stiffness matrices independently, and then superimpose the two to form the full six DOF model (Section 3.2.2.3). Section 3.2.2.3 also defines a transformation matrix so that non-planar geometries can be analyzed. After the element stiffness matrices are derived, the rest of the finite element method as it relates to the MATLAB coding process is described in Sections 3.2.3–3.2.5. Section 3.2.6 provides validation and verification of the MATLAB implementation of the membrane, bending, and combined membrane-bending elements.

In the following sections, the finite element method will be broken down into the following eight general steps:

**Step 1:** Discretize the Geometry

**Step 2:** Select the Element Types

**Step 3:** Select a Displacement Function

**Step 4:** Define the Strain/Displacement and Stress/Strain (or Equivalent) Relations

**Step 5:** Derive the Element Stiffness Matrix

**Step 6:** Assemble the Global Stiffness Matrix

**Step 7:** Apply Boundary Conditions and Solve for the Unknown Degrees of Freedom

**Step 8:** Solve for the Element Stresses

While Steps 1 and 6–8 are largely the same for the different element types, a stiffness matrix must be developed for each element type in Steps 2–5 (Section 3.2.2).

*3.2.1 Discretization.* Mentioned in the discussion of the geometry, the mesh type of choice is a structured arrangement of quadrilateral elements rather than an unstructured collocation of triangular elements, and is therefore easily created by two-dimensional linear interpolation. Another advantage of a structured mesh over an unstructured mesh is ease of coding. Each quadrilateral element has four definite neighboring cells and can be mapped to a rectangular computational domain, allowing a programmer to simply loop through the rows and columns of the mesh. The elements are ordered such that the upper left corner of the computational domain is the first element, and successive elements are then counted down through the rows and then right through the columns (see Figure 3.11). Using general quadrilaterals rather than rectangular elements allows a complex geometry to be fitted with a structured mesh. However, the disadvantage is that cells can be skewed and are more likely to have large aspect ratios, which typically increases the inaccuracy of the solution. The bird wing in the dive configuration (Figure 3.7) demonstrates this point precisely, where highly skewed cells dominate the sixth wing section and aspect ratio grows large as the section tapers nearly to a point.

*3.2.2 Element Stiffness Matrix.*

*3.2.2.1 Quadrilateral Membrane Element.* This derivation follows the nomenclature and methods of Reference [23, pp. 452–460].

**Step 2: Select the Element Type**

The general quadrilateral membrane element considered here stretches in two directions and thus has two degrees of freedom for each of its four nodes.

$$\{d\} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{Bmatrix} ; \quad \{d_1\} = \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} \quad (3.1)$$

Writing out the total eight displacements gives

$$\{d\} = \left\{ u_1 \quad v_1 \quad u_2 \quad v_2 \quad u_3 \quad v_3 \quad u_4 \quad v_4 \right\}^T \quad (3.2)$$

Since the element shape is an arbitrary quadrilateral, a transformation from the physical shape to a square element in the computational domain will be required in the displacement function selection.

### Step 3: Select a Displacement Function

In an isoparametric membrane formulation, the same shape functions that are used to define the element shape are also used to define the displacements within the element. Thus, the shape functions that map the natural coordinates  $r$  and  $s$  of any point in a square element to the global coordinates  $x$  and  $y$  in the general quadrilateral are also the shape functions that relate the internal displacements of the element  $u$  and  $v$  to its nodal displacements  $\{d\}$ . The transformation between the natural coordinate system and the global is depicted in Figure 3.8. The natural coordinates are attached to the element and rotate with the element. They need not be parallel with the global coordinates or orthogonal to each other. The four corners and edges of the quadrilateral are bounded by  $+1$  or  $-1$ . The displacement of any point  $P$  located at  $(x, y)$  in the element is described by  $u$  and  $v$ .

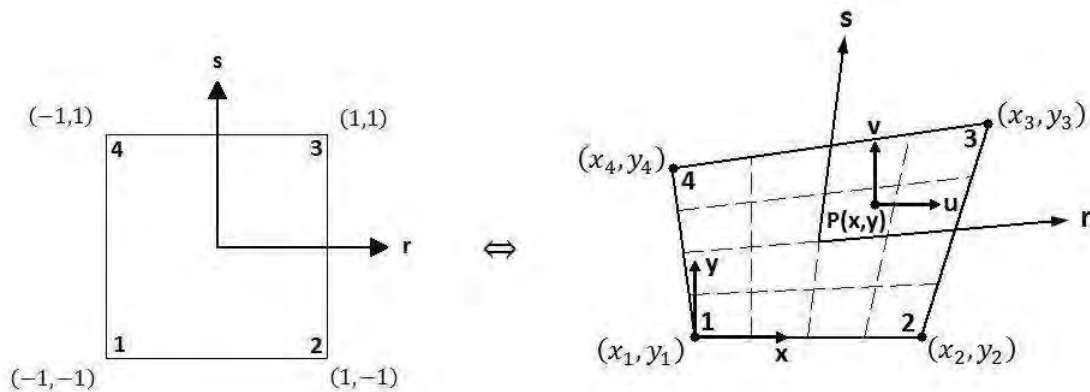


Figure 3.8: General Quadrilateral Element Mapping to Computational Domain [23]

The size and shape of the quadrilateral is determined by the eight nodal coordinates  $x_1, y_1, x_2, y_2, x_3, y_3, x_4,$  and  $y_4$ . Therefore, any internal coordinate can be determined by bilinear interpolation of the natural coordinates for first-order shape functions. The global coordinates can be expressed by eight unknown constants.

$$x = A_1 + A_2r + A_3s + A_4rs \quad ; \quad y = A_5 + A_6r + A_7s + A_8rs \quad (3.3)$$

The  $A$  coefficients are determined by evaluating  $x$  and  $y$  at the four nodes— $(-1,-1)$ ,  $(1,-1)$ ,  $(1,1)$ , and  $(-1,1)$  in natural coordinates and  $(x_1,y_1)$ ,  $(x_2,y_2)$ ,  $(x_3,y_3)$ , and  $(x_4,y_4)$  in global coordinates—and then simultaneously solving for the  $A$ 's.

$$\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{Bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{Bmatrix} \quad ; \quad \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{Bmatrix} A_5 \\ A_6 \\ A_7 \\ A_8 \end{Bmatrix} \quad (3.4)$$

Solution of these matrix equations for the  $A$ 's and substitution of the expressions for the  $A$ 's back into Eqs. (3.3) leads to the following mapping:

$$x = \frac{1}{4}[(1-r)(1-s)x_1 + (1+r)(1-s)x_2 + (1+r)(1+s)x_3 + (1-r)(1+s)x_4] \quad (3.5a)$$

$$y = \frac{1}{4}[(1-r)(1-s)y_1 + (1+r)(1-s)y_2 + (1+r)(1+s)y_3 + (1-r)(1+s)y_4] \quad (3.5b)$$

These coordinates can be expressed in matrix form in the following manner:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{Bmatrix} \quad (3.6)$$



Here the shape functions  $N_i$  relate the nodal coordinates  $(x_i, y_i)$  to any  $(x, y)$  position along the membrane. They are functions of the natural coordinates  $(r, s)$ .

$$N_1 = \frac{1}{4}(1-r)(1-s) \quad ; \quad N_2 = \frac{1}{4}(1+r)(1-s) \quad (3.7a)$$

$$N_3 = \frac{1}{4}(1+r)(1+s) \quad ; \quad N_4 = \frac{1}{4}(1-r)(1+s) \quad (3.7b)$$

The shape function partial derivatives with respect to the natural coordinates are

$$N_{1,r} = \frac{1}{4}(-1+s); \quad N_{2,r} = \frac{1}{4}(1-s); \quad N_{3,r} = \frac{1}{4}(1+s); \quad N_{4,r} = \frac{1}{4}(-1-s) \quad (3.8a)$$

$$N_{1,s} = \frac{1}{4}(-1+r); \quad N_{2,s} = \frac{1}{4}(-1-r); \quad N_{3,s} = \frac{1}{4}(1+r); \quad N_{4,s} = \frac{1}{4}(1-r) \quad (3.8b)$$

The partial derivatives of the global coordinates then become

$$\begin{Bmatrix} x_r \\ x_s \end{Bmatrix} = \frac{1}{4} \begin{bmatrix} -1+s & 1-s & 1+s & -1-s \\ -1+r & -1-r & 1+r & 1-r \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} \quad (3.9a)$$

$$\begin{Bmatrix} y_r \\ y_s \end{Bmatrix} = \frac{1}{4} \begin{bmatrix} -1+s & 1-s & 1+s & -1-s \\ -1+r & -1-r & 1+r & 1-r \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix} \quad (3.9b)$$

The displacement functions are now defined in the same manner as Eq. (3.6) with the shape functions defined by Eqs. (3.7):

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} \quad (3.10)$$

where  $u$  and  $v$  are displacements parallel to the global  $x$  and  $y$  coordinates. Eq. (3.10) can be represented simply as

$$\{\psi\} = [N]\{d\} \quad (3.11)$$

#### Step 4: Define the Strain/Displacement and Stress/Strain Relations

Now that the element has been selected and the displacement functions defined, the strain/displacement matrix  $[B]$  must be established such that

$$\{\varepsilon\} = [B]\{d\} \quad (3.12)$$

The usual relationship between strains and displacements [23, pp. 746-748] for the two-dimensional case is given as

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad ; \quad \varepsilon_y = \frac{\partial v}{\partial y} \quad ; \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad (3.13)$$

Eqs. (3.13) can be combined into the following matrix equation:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{\partial(\cdot)}{\partial x} & 0 \\ 0 & \frac{\partial(\cdot)}{\partial y} \\ \frac{\partial(\cdot)}{\partial y} & \frac{\partial(\cdot)}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} \quad (3.14)$$

or more succinctly

$$\{\varepsilon\} = [D']\{\psi\} \quad (3.15)$$

Since the global coordinates are themselves functions of the natural coordinates (Eqs. (3.6) and (3.7)), the Jacobian is needed to express the partial derivatives of the global coordinates. This variable transformation process is worked out in Appendix A.3 and Eqs. (A.38) and (A.39) are repeated here:

$$\frac{\partial(\cdot)}{\partial x} = \frac{1}{|J|} \left[ \frac{\partial(\cdot)}{\partial r} y_s - \frac{\partial(\cdot)}{\partial s} y_r \right] \quad ; \quad \frac{\partial(\cdot)}{\partial y} = \frac{1}{|J|} \left[ \frac{\partial(\cdot)}{\partial s} x_r - \frac{\partial(\cdot)}{\partial r} x_s \right] \quad (3.16)$$

$$\text{where } |J| = x_r y_s - x_s y_r \quad (3.17)$$

Substituting Eqs. (3.16) into Eq. (3.14) yields an alternate form of  $[D']$ :

$$[D'] = \frac{1}{|J|} \begin{bmatrix} \frac{\partial(\ )}{\partial r} y_s - \frac{\partial(\ )}{\partial s} y_r & 0 \\ 0 & \frac{\partial(\ )}{\partial s} x_r - \frac{\partial(\ )}{\partial r} x_s \\ \frac{\partial(\ )}{\partial s} x_r - \frac{\partial(\ )}{\partial r} x_s & \frac{\partial(\ )}{\partial r} y_s - \frac{\partial(\ )}{\partial s} y_r \end{bmatrix} \quad (3.18)$$

Substitution of Eq. (3.11) into Eq. (3.15) leads to the definition of the strain/displacement matrix  $[B]$  by Eq. (3.12):

$$\begin{array}{ccc} [B] & = & [D'] \quad [N] \\ (3 \times 8) & & (3 \times 2) \quad (2 \times 8) \end{array} \quad (3.19)$$

Thus  $[B]$  is computed by operation of  $[D']$  on  $[N]$ :

$$[B(r, s)] = \frac{1}{|J|} \begin{bmatrix} B_1 & B_2 & B_3 & B_4 \end{bmatrix} \quad (3.20)$$

where the submatrices of  $[B]$  are given by

$$[B_i] = \begin{bmatrix} (N_{i,r})y_s - (N_{i,s})y_r & 0 \\ 0 & (N_{i,s})x_r - (N_{i,r})x_s \\ (N_{i,s})x_r - (N_{i,r})x_s & (N_{i,r})y_s - (N_{i,s})y_r \end{bmatrix} \quad (3.21)$$

The strains are now related to the nodal displacements (Eq. (3.12)). The stresses are now related to the strains by the constitutive matrix  $[D]$  [23, pp. 748-751]

$$\{\sigma\} = [D]\{\varepsilon\} \quad (3.22)$$

where  $[D]$  is defined for a state of plane stress as

$$[D] = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix} \quad (3.23)$$

where  $E$  is Young's modulus and  $\nu$  is Poisson's ratio. Upon substitution of Eq. (3.12) into Eq. (3.13), the element stresses are related directly to the nodal displacements.

$$\{\sigma\} = [D][B]\{d\} \quad (3.24)$$

### Step 5: Derive the Element Stiffness Matrix

To derive an expression for the stiffness matrix  $[k]$ , the principle of minimum potential energy is employed, where the strain energy is given by

$$U = \frac{1}{2} \iiint_V \{\varepsilon\}^T \{\sigma\} dV \quad (3.25)$$

A derivation following this method is given in Reference [23, pp. 317-320], and results in the following expression for the stiffness matrix:

$$[k] = \iiint_V [B]^T [D] [B] dV \quad (3.26)$$

Since  $[B]$  does not vary through the thickness of the membrane element, the stiffness matrix becomes an area integral.

$$[k] = t \iint_A [B]^T [D] [B] dx dy \quad (3.27)$$

However,  $[B]$  is a function of the natural coordinates  $r$  and  $s$ , and not the global coordinates  $x$  and  $y$  which define the volume over which the integration is to be performed. Thus the variables must be transformed by using the determinant of the

Jacobian matrix  $J$  (a proof for a general transformation is given in [43]).

$$\iint_A f(x, y) dx dy = \iint_A f(r, s) |J| dr ds \quad (3.28)$$

Since the area is bounded by  $\pm 1$  in the natural coordinates,  $[k]$  becomes

$$[k] = t \int_{-1}^1 \int_{-1}^1 [B]^T [D] [B] |J| dr ds \quad (3.29)$$

Since  $[B]$  is a rather complicated expression, the integration determining the element stiffness matrix is best carried out numerically. The Gaussian quadrature method used to evaluate the integrals is presented in Appendix A.5.

*3.2.2.2 Quadrilateral Bending Element.* A general quadrilateral bending element formulation was not found among the finite element literature. Therefore a formulation of such a bending element was derived using classic Kirchoff plate theory and using the same shape functions that defined the membrane element shape in the previous section. The formulation is essentially a synthesis of Chapters 10 and 12 found in Reference [23]. A stand-alone derivation of the bending element stiffness matrix is provided in Appendix A. The results of Steps 2–5 are repeated here for completeness.

### Step 2: Select the Element Type

A plate element can translate and rotate out-of-plane. This gives the plate element three degrees of freedom per node—a transverse displacement  $w$  and two rotations  $\theta_x$  and  $\theta_y$ .

$$\{d\} = \begin{Bmatrix} d_i \\ d_j \\ d_m \\ d_n \end{Bmatrix} ; \quad \{d_i\} = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{Bmatrix} \quad (3.30)$$

Writing out the 12 displacements gives

$$\{d\} = \left\{ w_i \ \theta_{xi} \ \theta_{yi} \ w_j \ \theta_{xj} \ \theta_{yj} \ w_m \ \theta_{xm} \ \theta_{ym} \ w_n \ \theta_{xn} \ \theta_{yn} \right\}^T \quad (3.31)$$

As with the membrane element, a transformation from the arbitrary quadrilateral to a square element in the computational domain is required.

### Step 3: Select a Displacement Function

Although the same shape functions that are used to define the membrane element shape are also used to define the bending element shape, the formulation is not technically isoparametric because the nodal displacements are out-of-plane and have their own shape functions. The same element shape mapping is shown in Figure 3.9 for the bending element, where representative loads are shown at node  $i$ .

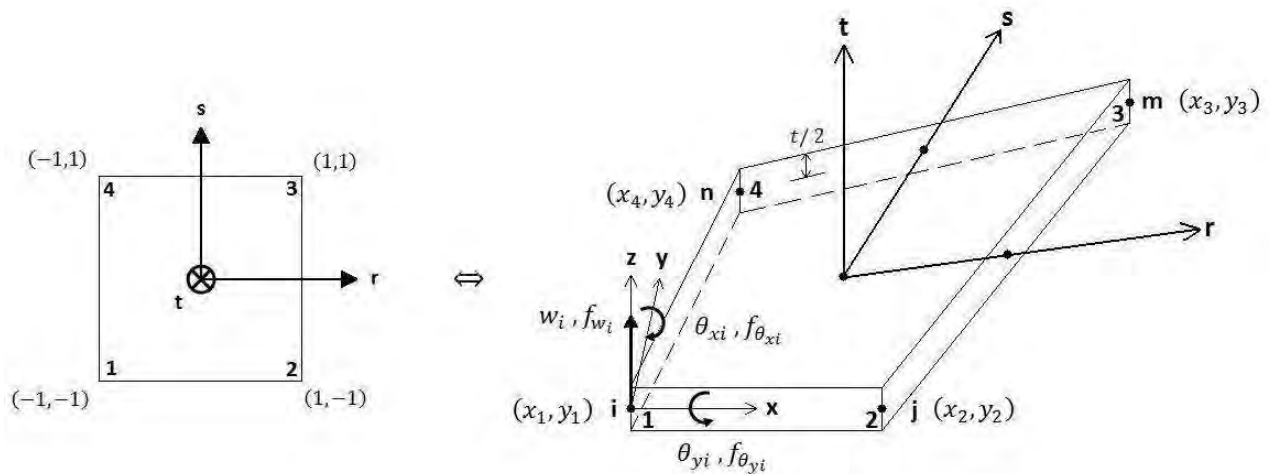


Figure 3.9: Mapping of General Quadrilateral Element to Computational Domain

As with the membrane element, the size and shape of the quadrilateral is determined by the eight nodal coordinates by Eqs. (3.6) and (3.7), and obviously have the same partial derivatives (Eqs. (3.9)). Given that there are 12 degrees of freedom, a 12-term polynomial is selected for the displacement function that enables rigid body

motion and constant strain.

$$\begin{aligned}
w = & a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 + a_7x^3 + a_8x^2y \\
& + a_9xy^2 + a_{10}y^3 + a_{11}x^3y + a_{12}xy^3
\end{aligned} \tag{3.32}$$

The rotations follow from Eq. (3.32) since

$$\theta_x = \frac{\partial w}{\partial y} \quad ; \quad \theta_y = -\frac{\partial w}{\partial x} \tag{3.33}$$

The displacements can be put into the form of Eq. (3.11) by following the process of Eqs. (A.28)–Eq. (A.33).

#### Step 4: Define Curvature/Displacement and Moment/Curvature Relations

Rather than relating the displacements to the strains (Eq. (3.12)) which vary linearly through the thickness of the bending element, displacements are instead related to the curvatures (Eq. (A.6)):

$$\{\kappa\} = [B]\{d\} \tag{3.34}$$

The curvature/displacement matrix  $[B]$  is calculated by the process explained in Appendix A.3. Because the stresses also vary linearly through the thickness of the bending element, the curvatures are related to the moments instead.

$$\{M\} = [D]\{\kappa\} \tag{3.35}$$

where the constitutive matrix  $[D]$  is defined as

$$[D] = D \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{3.36}$$

Here  $D$  is expressed as

$$D = \frac{Et^3}{12(1 - \nu^2)} \quad (3.37)$$

Combining Eqs. (3.34) and (3.35) relates the moments to the displacements.

$$\{M\} = [D][B]\{d\} \quad (3.38)$$

### Step 5: Derive the Element Stiffness Matrix

Step 5 must be modified slightly from the process followed in the membrane derivation, since moments are computed rather than stresses. The potential energy of the plate is given as

$$U = \frac{1}{2} \iint (M_x \kappa_x + M_y \kappa_y + M_{xy} \kappa_{xy}) dA \quad (3.39)$$

Minimizing the potential energy leads to a slight difference from Eq. (3.27), where thickness no longer sits in front of the integral.

$$[k] = \iint_A [B]^T [D] [B] dx dy \quad (3.40)$$

The rest of Step 5 is the same for the bending element and results in a [12 x 12] stiffness matrix.

*3.2.2.3 Combined Quadrilateral Membrane-Bending Element.* A shell element is a thin curved surface with six degrees of freedom at each node. When the radius of curvature approaches infinity, the shell element approaches a flat plate. When a surface is modeled by many small shell elements, each element may be considered as a flat plate with a particular orientation in space. A conventional finite plate bending element, such as discussed in the previous section, does not allow in-plane membrane deformation. In the analysis of three-dimensional structures, membrane and bending stiffnesses are uncoupled for small displacements. Thus, a shell ele-



ment can be formed by superimposing the membrane and bending stiffness matrices together in the following manner:

$$[k]_{20 \times 20} = \left[ \begin{array}{c|c} [k_m]_{8 \times 8} & [0]_{8 \times 12} \\ \hline [0]_{12 \times 8} & [k_b]_{12 \times 12} \end{array} \right] \quad (3.41)$$

where subscripts  $m$  and  $b$  denote membrane and bending deformations of the shell element. In this formulation, the bending and membrane elements do not share common degrees of freedom and hence are uncoupled. Note that this results in a stiffness matrix of order  $[20 \times 20]$ , or five degrees of freedom per node. The element structure stiffness equations are

$$\left\{ \begin{array}{c} \{f_m\}_{8 \times 1} \\ \{f_b\}_{12 \times 1} \end{array} \right\} = \left[ \begin{array}{c|c} [k_m]_{8 \times 8} & [0]_{8 \times 12} \\ \hline [0]_{12 \times 8} & [k_b]_{12 \times 12} \end{array} \right] \left\{ \begin{array}{c} \{d_m\}_{8 \times 1} \\ \{d_b\}_{12 \times 1} \end{array} \right\} \quad (3.42)$$

For a single shell element, there is no twisting, in-plane rotation  $\theta_z$  (referred to as a “drilling” rotation), but for an assemblage of elements, a bending rotation in one element may result in a drilling rotation in an adjacent element. To account for this, the drilling degree of freedom must be added to each node, raising the order of the stiffness matrix to a  $[24 \times 24]$ .

$$\left\{ \begin{array}{c} \{f_m\}_{8 \times 1} \\ \{f_b\}_{12 \times 1} \\ \{f_{\theta_z}\}_{4 \times 1} \end{array} \right\} = \left[ \begin{array}{c|c|c} [k_m]_{8 \times 8} & [0]_{8 \times 12} & [0]_{20 \times 4} \\ \hline [0]_{12 \times 8} & [k_b]_{12 \times 12} & \\ \hline [0]_{4 \times 20} & [0]_{4 \times 4} & \end{array} \right] \left\{ \begin{array}{c} \{d_m\}_{8 \times 1} \\ \{d_b\}_{12 \times 1} \\ \{d_{\theta_z}\}_{4 \times 1} \end{array} \right\} \quad (3.43)$$

Since there is no stiffness associated with the drilling degrees of freedom, the stiffness matrix will be singular if all the elements joining a node are coplanar. If flat geometry is to be analyzed by superposition, the on-diagonal null matrix is replaced with an artificial stiffness matrix which causes the twisting rotations to produce corresponding

moments  $M_z$ . One suggested fictitious stiffness matrix [11, p. 352] is the following:

$$\begin{Bmatrix} M_{z1} \\ M_{z2} \\ M_{z3} \\ M_{z4} \end{Bmatrix} = \alpha EV \begin{bmatrix} 1.0 & -0.5 & -0.5 & -0.5 \\ -0.5 & 1.0 & -0.5 & -0.5 \\ -0.5 & -0.5 & 1.0 & -0.5 \\ -0.5 & -0.5 & -0.5 & 1.0 \end{bmatrix} \begin{Bmatrix} \theta_{z1} \\ \theta_{z2} \\ \theta_{z3} \\ \theta_{z4} \end{Bmatrix} \quad (3.44)$$

where  $\alpha$  is a small number such as 0.3,  $E$  is Young's modulus, and  $V$  is the elemental volume. The element volume can be calculated directly from its coordinates by the following formula:

$$V = \frac{t}{2} \left( \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & x_3 \\ y_2 & y_3 \end{vmatrix} + \begin{vmatrix} x_3 & x_4 \\ y_3 & y_4 \end{vmatrix} + \begin{vmatrix} x_4 & x_1 \\ y_4 & y_1 \end{vmatrix} \right) \quad (3.45)$$

where  $t$  is the element thickness.

The current order of the displacement matrix in Eq. (3.43) is

$$\{d\} = \{u_i \ v_i \ u_j \ v_j \ u_m \ v_m \ u_n \ v_n \ w_i \ \theta_{xi} \ \theta_{yi} \ w_j \ \theta_{xj} \ \theta_{yj} \\ w_m \ \theta_{xm} \ \theta_{ym} \ w_n \ \theta_{xn} \ \theta_{yn} \ \theta_{zi} \ \theta_{zj} \ \theta_{zm} \ \theta_{zn}\}^T \quad (3.46)$$

Following the degrees-of-freedom indexing scheme that will be described in Section 3.2.3, the displacement vector should be ordered as follows:

$$\{d\} = \{u_i \ v_i \ w_i \ \theta_{xi} \ \theta_{yi} \ \theta_{zi} \ u_j \ v_j \ w_j \ \theta_{xj} \ \theta_{yj} \ \theta_{zj} \ u_m \\ v_m \ w_m \ \theta_{xm} \ \theta_{ym} \ \theta_{zm} \ u_n \ v_n \ w_n \ \theta_{xn} \ \theta_{yn} \ \theta_{zn}\}^T \quad (3.47)$$

Before rearranging the combined stiffness matrix  $[k]$ , the membrane and bending stiffness matrices can be further broken down into submatrices.

$$[k_m]_{8 \times 8} = \begin{bmatrix} [k_{11}]_m & [k_{12}]_m & [k_{13}]_m & [k_{14}]_m \\ [k_{21}]_m & [k_{22}]_m & [k_{23}]_m & [k_{24}]_m \\ [k_{31}]_m & [k_{32}]_m & [k_{33}]_m & [k_{34}]_m \\ [k_{41}]_m & [k_{42}]_m & [k_{43}]_m & [k_{44}]_m \end{bmatrix} \quad (3.48)$$

$$[k_b]_{12 \times 12} = \begin{bmatrix} [k_{11}]_b & [k_{12}]_b & [k_{13}]_b & [k_{14}]_b \\ [k_{21}]_b & [k_{22}]_b & [k_{23}]_b & [k_{24}]_b \\ [k_{31}]_b & [k_{32}]_b & [k_{33}]_b & [k_{34}]_b \\ [k_{41}]_b & [k_{42}]_b & [k_{43}]_b & [k_{44}]_b \end{bmatrix} \quad (3.49)$$

The fictitious stiffness matrix of Eq. (3.44) can be rewritten slightly as follows:

$$[k_f]_{4 \times 4} = \begin{bmatrix} k_f & k'_f & k'_f & k'_f \\ k'_f & k_f & k'_f & k'_f \\ k'_f & k'_f & k_f & k'_f \\ k'_f & k'_f & k'_f & k_f \end{bmatrix} \quad (3.50)$$

where  $k_f$  is defined as

$$k_f = \alpha EV \quad (3.51)$$

and  $k'_f$  is defined as

$$k'_f = -0.5 \alpha EV \quad (3.52)$$

Using the notation of Eq. (3.48)–Eq. (3.50), the total membrane and bending stiffness matrix is combined [36, p. 389] in the following manner:



Substituting the local displacements and forces of Eqs. (3.54) into the local stiffness matrix equation yields

$$[T]\{f\} = [\hat{k}][T]\{d\} \quad (3.56)$$

which leads to

$$\{f\} = [T]^{-1}[\hat{k}][T]\{d\} \quad (3.57)$$

Thus the global stiffness matrix is computed via the following relation:

$$[k] = [T]^{-1}[\hat{k}][T] \quad (3.58)$$

Since  $[T]$  is an orthogonal matrix, the transpose of  $[T]$  can be used instead of its inverse to save computational time.

$$[k] = [T]^T[\hat{k}][T] \quad (3.59)$$

Local forces can also be mapped back to the global forces using the transpose.

$$\{f\} = [T]^T[\hat{f}] \quad (3.60)$$

The transformation matrix [21, p. 388] relating global degrees of freedom to local is expressed as

$$\begin{Bmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \\ \hat{\theta}_x \\ \hat{\theta}_y \\ \hat{\theta}_z \end{Bmatrix} = \begin{bmatrix} l_{11} & l_{12} & l_{13} & 0 & 0 & 0 \\ l_{21} & l_{22} & l_{23} & 0 & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & l_{11} & l_{12} & l_{13} \\ 0 & 0 & 0 & l_{21} & l_{22} & l_{23} \\ 0 & 0 & 0 & l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix} \quad (3.61)$$

where the  $l_{ij}$ 's are the direction cosines given as

$$l_{11} = \cos(\mathbf{e}_x, \mathbf{e}_r) \quad ; \quad l_{12} = \cos(\mathbf{e}_y, \mathbf{e}_r) \quad ; \quad l_{13} = \cos(\mathbf{e}_z, \mathbf{e}_r) \quad (3.62a)$$

$$l_{21} = \cos(\mathbf{e}_x, \mathbf{e}_s) \quad ; \quad l_{22} = \cos(\mathbf{e}_y, \mathbf{e}_s) \quad ; \quad l_{23} = \cos(\mathbf{e}_z, \mathbf{e}_s) \quad (3.62b)$$

$$l_{31} = \cos(\mathbf{e}_x, \mathbf{e}_t) \quad ; \quad l_{32} = \cos(\mathbf{e}_y, \mathbf{e}_t) \quad ; \quad l_{33} = \cos(\mathbf{e}_z, \mathbf{e}_t) \quad (3.62c)$$

Eq. (3.61) transforms the displacements at a particular node, for example, at node  $i$ :

$$\{\hat{d}_i\} = [T_n]\{d_i\} \quad (3.63)$$

Thus, the transformation for a four node element becomes

$$[T]_{24 \times 24} = \begin{bmatrix} [T_n]_{6 \times 6} & 0 & 0 & 0 \\ 0 & [T_n]_{6 \times 6} & 0 & 0 \\ 0 & 0 & [T_n]_{6 \times 6} & 0 \\ 0 & 0 & 0 & [T_n]_{6 \times 6} \end{bmatrix} \quad (3.64)$$

The direction cosines can of course be calculated from the familiar calculus formula

$$l_{ij} = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|} \quad (3.65)$$

However, determining the unit vectors is not as straight forward. The global unit vectors are simply

$$\mathbf{e}_x = [1 \ 0 \ 0] \quad ; \quad \mathbf{e}_y = [0 \ 1 \ 0] \quad ; \quad \mathbf{e}_z = [0 \ 0 \ 1] \quad (3.66)$$

but the unit vectors of the natural coordinates will require some effort to calculate. The element normal unit vector  $\mathbf{e}_t$  can be computed from the cross product of two vectors lying in the element, for example, vectors extending from node 1 to nodes 2

and 4 of the element as shown in Figure 3.10.

$$\mathbf{e}_t = \frac{\vec{v}_{21} \times \vec{v}_{41}}{\|\vec{v}_{21} \times \vec{v}_{41}\|} \quad (3.67)$$

In order to determine the unit vectors  $\mathbf{e}_r$  and  $\mathbf{e}_s$ , the midpoint of the element and the midpoints of the sides  $r = 1$  and  $s = 1$  are needed (shown as  $P_5$ ,  $P_6$ , and  $P_7$  in Figure 3.10). The  $x$  and  $y$  coordinates of these three points are calculated by evaluating Eqs. (3.6) and (3.7) at  $(r, s) = (0, 0)$ ,  $(1, 0)$ , and  $(0, 1)$ . The plane of the element can be defined by its normal vector and one of its four sets of nodal coordinates:

$$e_{tx}(x - x_1) + e_{ty}(y - y_1) + e_{tz}(z - z_1) = 0 \quad (3.68)$$

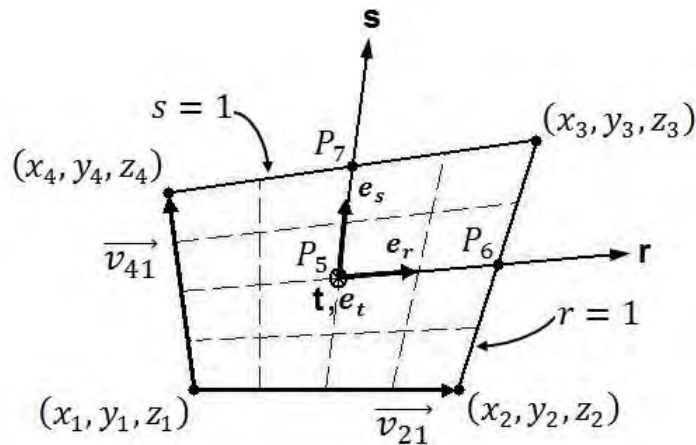


Figure 3.10: Natural Coordinate Unit Vectors

With the  $x$  and  $y$  coordinates obtained from the shape function mapping, Eq. (3.68) can be evaluated to determine the  $z$  coordinates.

$$z = z_1 - \frac{1}{e_{tz}} [e_{tx}(x - x_1) + e_{ty}(y - y_1)] \quad (3.69)$$

Thus, unit vectors  $\mathbf{e}_r$  and  $\mathbf{e}_s$  are the normalized vectors calculated with the coordinates of points 5, 6, and 7.

$$\mathbf{e}_s = \frac{\langle P_{6x} - P_{5x}, P_{6y} - P_{5y}, P_{6z} - P_{5z} \rangle}{\|\langle P_{6x} - P_{5x}, P_{6y} - P_{5y}, P_{6z} - P_{5z} \rangle\|} \quad (3.70)$$

$$\mathbf{e}_s = \frac{\langle P_{7x} - P_{5x}, P_{7y} - P_{5y}, P_{7z} - P_{5z} \rangle}{\|\langle P_{7x} - P_{5x}, P_{7y} - P_{5y}, P_{7z} - P_{5z} \rangle\|} \quad (3.71)$$

With the unit vectors  $\mathbf{e}_r$ ,  $\mathbf{e}_s$ , and  $\mathbf{e}_t$ , Eq. (3.65) can be applied nine times to compute the direction cosines of Eqs. (3.62).

*3.2.3 Global Stiffness Matrix.* Step 6 of the finite element method involves assembling the local stiffness matrices together to form the global stiffness matrix for the structure. But before the global stiffness matrix can be compiled, the elemental stiffness matrices must each be evaluated by Eq. (3.29), Eq. (3.40), or Eq. (3.53). A procedure is constructed to loop through each element in the computational domain. Mentioned in Section 3.2.1, the elements are indexed by counting down through the rows and then to the right through the columns. The degrees of freedom are also numbered in this fashion along the nodes. An example two DOF membrane mesh is shown in Figure 3.11. The degrees of freedom corresponding to each element are then ordered in a vector. The arrows in Figure 3.11 indicate that the degrees of freedom are counted counterclockwise starting with the bottom left hand corner of the element (see also Figure 3.8). Indices representing the column and row number of the element's position within the mesh facilitate selecting the corresponding degrees of freedom for the element based off its number. Along with the degrees of freedom, the nodal coordinates corresponding to each element will also need to be assembled to compute the strain-displacement matrix, and ultimately evaluate the stiffness matrix. The nodal coordinates define the shape of an element, which alone differentiates one element's stiffness over another's.

Once all of the elemental stiffness matrices  $[k_e]$  have been evaluated, the direct stiffness method [23, p. 37] forms the global stiffness matrix  $[K]$  by directly assembling



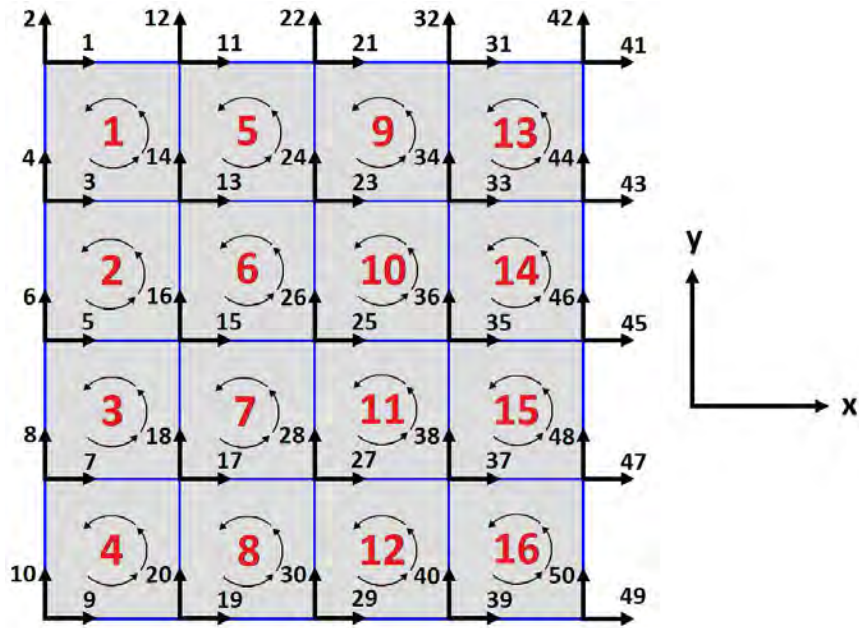


Figure 3.11: Computational Domain Indexing Example

individual element stiffness matrices. In other words, elemental stiffness matrices are superimposed to form the total structure stiffness matrix.

$$[K] = \sum_{e=1}^N [k_e] \quad (3.72)$$

*3.2.4 Displacement Solution.* In Step 7 of the finite element method, the boundary conditions consisting of the external forces and any constrained degrees of freedom are applied to the global structure stiffness equation. In a manner similar to the global stiffness matrix, the global nodal loads are obtained by lumping the body forces, distributed loads, and concentrated nodal loads at the proper nodes into a column vector. Section 3.3.2 describes how the aerodynamic load calculated for the bird wing is divided among the nodes.

$$\{F\} = \sum_{e=1}^N \{f_e\} \quad (3.73)$$

Resolving body and surface forces into concentrated nodal loads usually involves a single or double integral. An example of integrating a surface load can be found in Reference [23, pp. 460–462]. The fixed and the free degrees of freedom are specified in two different vectors. For instance, one boundary condition choice for the bird wing is to clamp the root chord. In this case, the user would specify the degrees of freedom along the root to be fixed and all other degrees of freedom as free. A script entitled “dofview” is included in Appendix D.2 to aid the user in determining which degrees-of-freedom indices correspond to nodes in a particular area of the geometry. The script displays a text label at each node specifying the first degree-of-freedom index at that node. It is called from the command window by “dofview(params,[dof])”, where “params” is the MATLAB structure associated with a geometry file (pre-loaded in the workspace) and [dof] is the number of degrees of freedom per node (i.e. two for a membrane model). An example output of “dofview” is shown in Figure 3.12 for a two-sectioned, 3-DOF model.

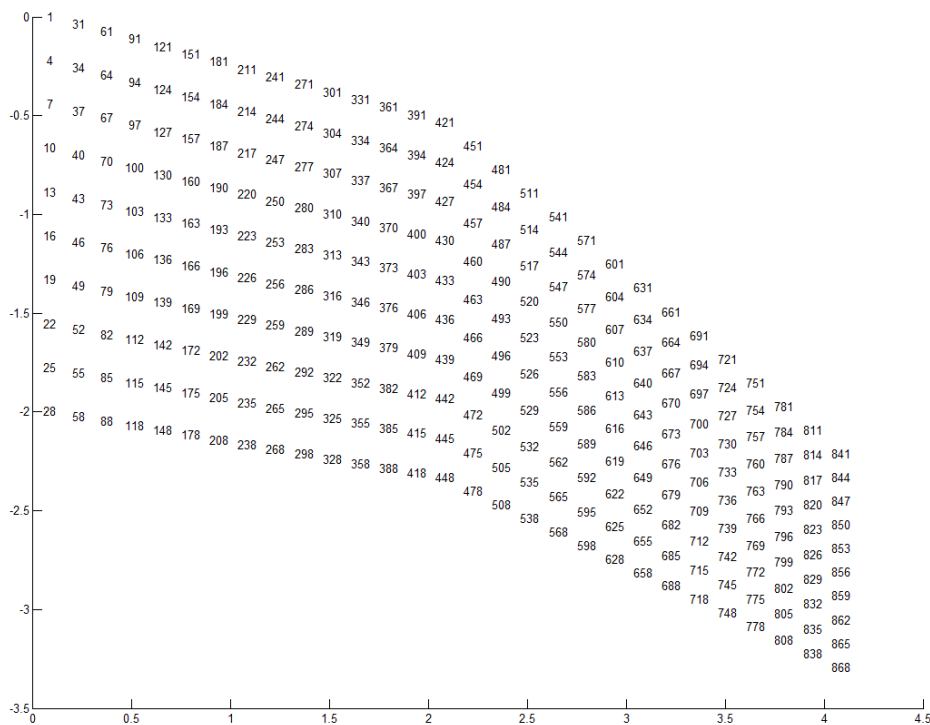


Figure 3.12: Computational Domain Indexing Example

With the fixed degrees of freedom specified, the corresponding displacements can be directly set to zero in the total structure displacement matrix  $\{d\}$ . Thus the global structure stiffness equations can be solved.

$$\{F\} = [K]\{d\} \quad (3.74)$$

It is important to note that  $\{F\}$  and  $\{d\}$  may represent vectors of moments and rotations rather than forces and translations. With the free degrees of freedom specified, Eq. (3.74) can be elegantly solved in MATLAB by the following command:

$$d(\text{freedofs}, 1) = K(\text{freedofs}, \text{freedofs}) \setminus F(\text{freedofs}, 1) \quad (3.75)$$

*3.2.5 Element Stresses.* The final step of the method consists of visualizing the results of the displacement solution. After calculating the displacements  $\{d\}$ , the elemental strains and stresses are computed from Eqs. (3.12) and (3.24), respectively. For the two-dimensional state of stress, three independent stresses exist and are represented by the column vector

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \quad (3.76)$$

The principal stresses are determined from Mohr's Circle [12, pp. 474–499].

$$\sigma_{max} = \frac{\sigma_x + \sigma_y}{2} + \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} \quad (3.77a)$$

$$\sigma_{min} = \frac{\sigma_x + \sigma_y}{2} - \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} \quad (3.77b)$$

The von Mises effective stress  $\sigma_{vm}$  [31, pp. 239–247] is defined as the uniaxial tensile stress that would create the same distortion energy as is created by the actual com-

combination of applied stresses, and thus treats multiaxial stresses as if they were due to pure tensile loading. Von Mises stress for the two-dimensional case can be calculated with the principal stress.

$$\sigma_{vm} = \sqrt{\sigma_{max}^2 - \sigma_{max}\sigma_{min} + \sigma_{min}^2} \quad (3.78)$$

When plotting a structure’s deformed state, the element patches can be colored by the von Mises stress for effective visualization.

*3.2.6 Validation.* This section provides validation of the membrane and of the bending models against classical analysis. Comparisons with the individual membrane and bending models are made for the combined membrane-bending model.

*3.2.6.1 Membrane Element.* A MATLAB code entitled “FEm” is included in Appendix B.1 that implements the membrane model. Before running the model, a geometry with the “params” structure must be loaded into the command window, and the force vector (Line 92) and fixed degrees of freedom (Line 93) must be set within the script. The function is called from the command line by “FEm(params)” and outputs four plots, three elemental stresses (Eq. (3.76)) and the von Mises stress (Eq. (3.78)).

Perhaps unconventional for membrane elements, the membrane model developed in Appendix B.1 is used to model the deflection of a cantilever beam. The solution is validated against elementary beam theory and another membrane finite element solution created by M. Gosz [13, pp. 176–180]. The cantilever beam shown in Figure 3.13 has a concentrated force,  $P = 10 \text{ kN}$ , at the right end, and the following material properties: Young’s modulus  $E = 200 \text{ GPa}$  and Poisson’s ratio  $\nu = 0.3$ . The beam has a length of  $3 \text{ m}$ , a height of  $20 \text{ cm}$ , and a thickness of  $20 \text{ mm}$ . Because the ratio of the length and height to the thickness of the beam is 15 and 150 respectively (Figure 3.13 not shown proportionately), a membrane model provides an accurate solution given that the mesh is adequately refined.

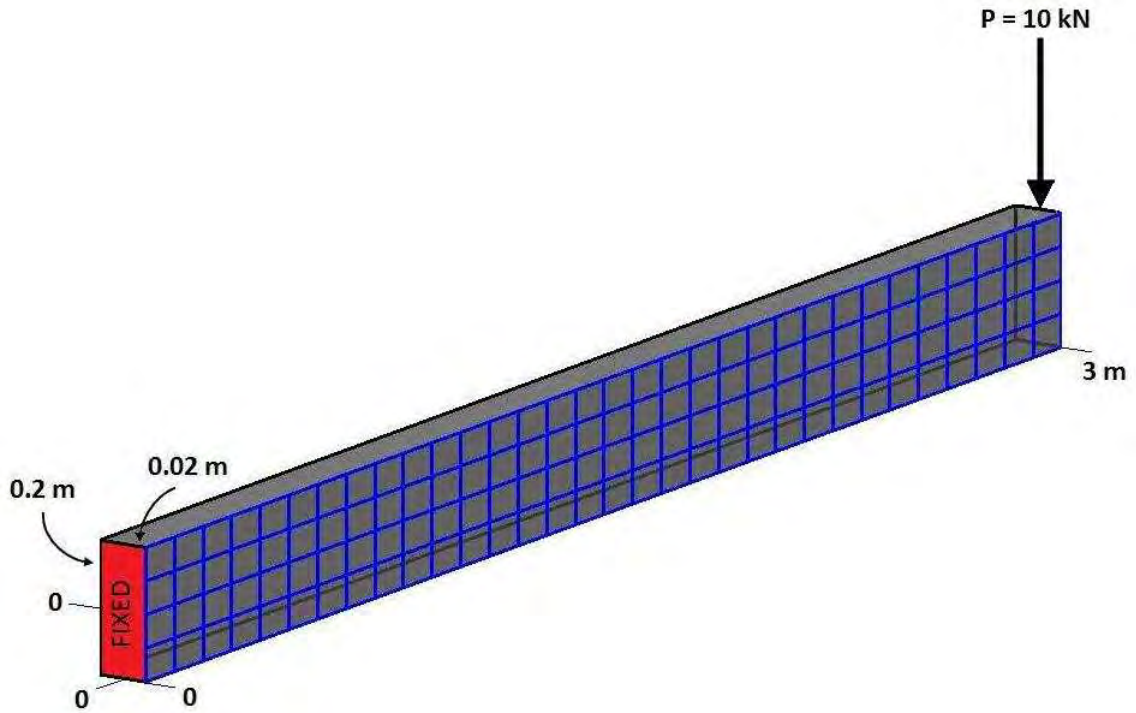


Figure 3.13: Cantilever Beam Mesh

From beam theory [12, p. 906], the deflection profile of a cantilever beam loaded on the right end by a concentrated force  $P$  is given as

$$\delta(x) = \frac{Px^2}{6EI}(3L - x) \quad (3.79)$$

where  $E$  is Young's modulus,  $I$  is the moment of inertia of the beam's cross section about the neutral axis, and  $L$  is the length of the beam. The deflection at  $L$  is simply

$$\delta(L) = \frac{PL^3}{3EI} \quad (3.80)$$

For the beam at hand, the analytical deflection at the end of the beam is

$$\delta(L) = \frac{(-10000 \text{ N})(3 \text{ m})^3}{3(200 \times 10^9 \text{ N/m}^2) \left[ \frac{(0.02 \text{ m})(0.2 \text{ m})^3}{12} \right]} = -33.75 \text{ mm} \quad (3.81)$$

The end deflection of the beam predicted by “FEm” is compared with the analytical solution and Gosz’s finite element solution in Table 3.2, using four different mesh refinements: 2 x 5 mesh, 2 x 20 mesh, 2 x 40 mesh, and 4 x 40 mesh (which is displayed on the beam in Figure 3.13). It is evident that with increasing mesh refinement, “FEm” nears the analytical solution. The “FEm” solution has less than 0.1 percent error with Gosz’s finite element solutions, and thus can be said to perfectly match Gosz’s membrane model for all practical purposes.

Table 3.2: End Deflection [mm] of Cantilever Beam

<b>Mesh Size</b>	$\delta_L$ (Gosz)	$\delta_L$ (FEm)	% Error (FEm–Analytical)	% Error (FEm–Gosz)
2 x 5	-7.5259	-7.5262	-77.7002	0.0037
2 x 20	-27.2109	-27.2057	-19.3905	-0.0191
2 x 40	-31.3006	-31.3251	-7.1850	0.0782
4 x 40	-31.8934	-31.8720	-5.5643	-0.0670

Figure 3.14 displays contour plots of the von Mises stress for the four different meshes. The 2 x 5 grid has very large aspect ratio cells and under-predicts the deflection at the end of the beam by a 78 percent error. This implies that there are not enough cells along the axis of the beam to capture the cubic deflection profile (Eq. (3.79)). Not as many cells are needed along the height since the bending stress varies linearly along the height. Hence, the next two refinements add more elements in the axial direction. The 2 x 40 mesh now only has a 7 percent error with the analytical solution. Only having two elements along the height of the beam effectively captures one stress magnitude that is equal and opposite above and below the neutral axis of the beam. With the 4 x 40 refinement, only a 5 percent error with the analytical solution remains. To obtain an even more accurate solution, more elements should be added in both directions, and the ratio of the number of cells in either direction should be such that the mesh has square cells. A plot of the bending stress  $\sigma_x$  is also included in Figure 3.15 which clearly shows the tension and compression of the beam above and below the neutral axis.

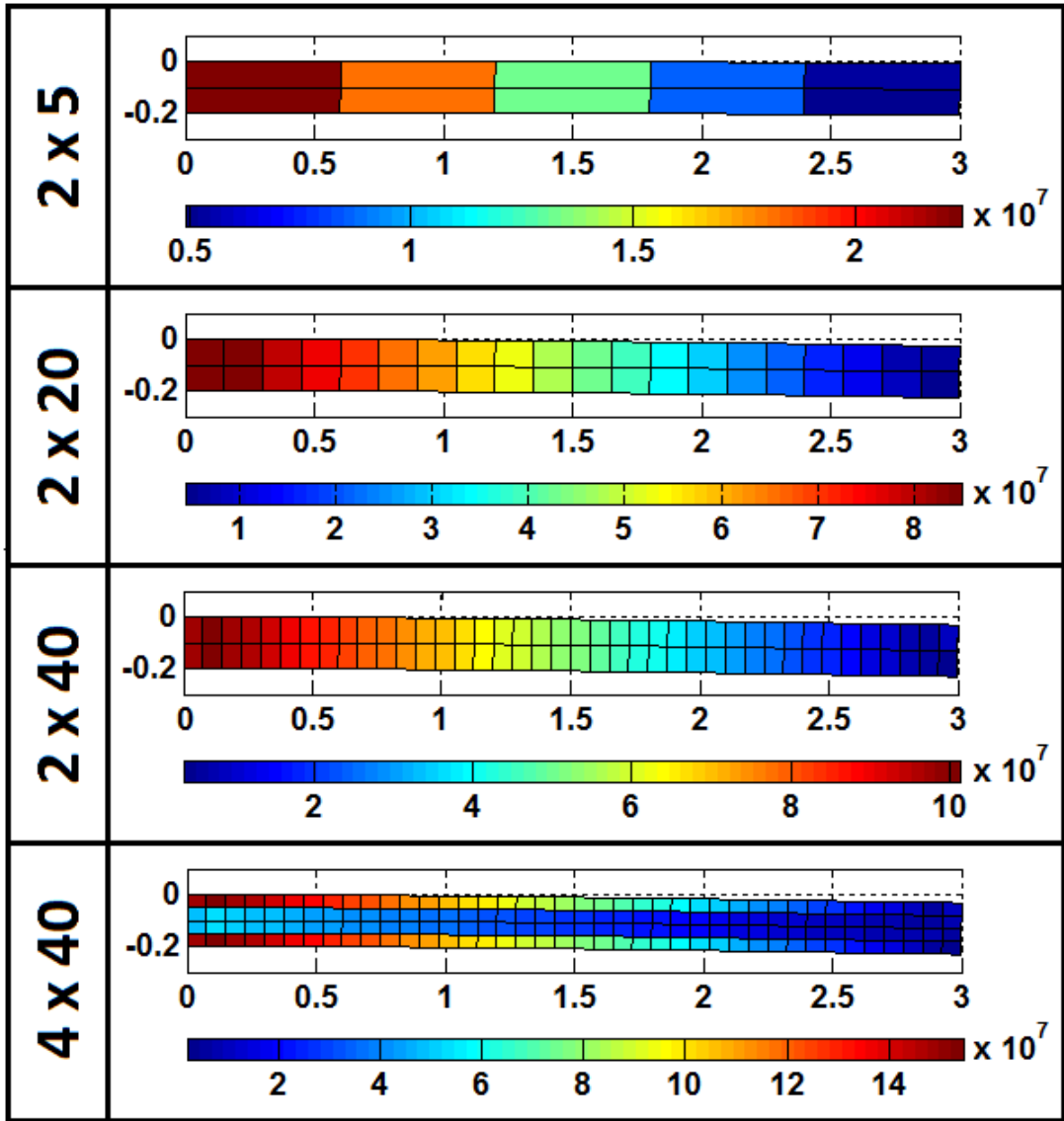


Figure 3.14: Von Mises Stress [Pa] of Cantilever Beam for Four Meshes

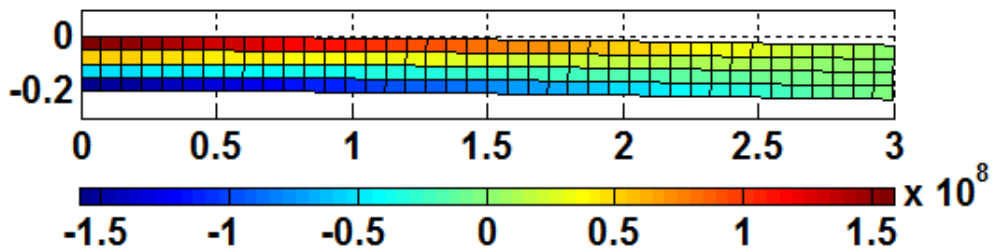


Figure 3.15: Bending Stress [Pa] of Cantilever Beam for 4 x 40 Membrane Mesh

3.2.6.2 *Bending Element.* A MATLAB code entitled “FEb” is included in Appendix B.2 that implements the bending model. Just like running the “FEm” code, the “params” structure must be preloaded into the command window, and the force vector (Line 87) and fixed degrees of freedom (Line 88) must be set within the script. The function is called from the command line by “FEb(params)” and outputs four plots, three elemental stresses calculated at  $z = t/2$  (Eqs. (A.11)) and the resulting von Mises stress (Eq. (3.78)).

The bending model developed in Appendix B.2 is used to model the deflection of a simply-supported square plate with a concentrated load at the center. The solution is validated against the simply-supported rectangular plate formulation from Timoshenko’s classic plate solutions [44, pp. 141–143]. The plate shown in Figure 3.16 has a concentrated force,  $P = 10 \text{ kN}$ , at the center, and the following material properties: Young’s modulus  $E = 200 \text{ GPa}$  and Poisson’s ratio  $\nu = 0.3$ . It has side lengths of  $2 \text{ m}$  and a thickness of  $20 \text{ cm}$ .

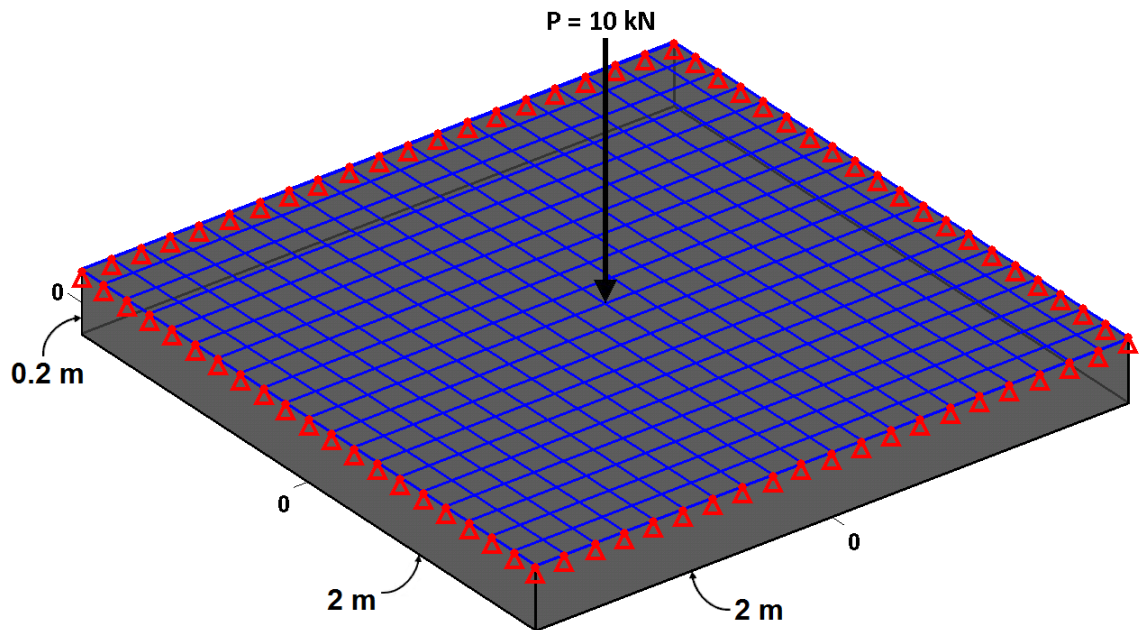


Figure 3.16: Plate Bending Mesh



Timoshenko's solution for the central or maximum deflection of a rectangular plate with side lengths  $a$  and  $b$  is given by

$$w_{max} = \alpha \frac{Pa^2}{D} \quad (3.82)$$

where  $\alpha$  is a numerical factor that depends on the ratio  $b/a$ , and the flexural rigidity  $D$  is calculated by Eq. (3.37). For a square plate ( $b/a=1$ ), Timoshenko calculates  $\alpha$  to be 0.01160. Thus for a square plate with side length  $L$ , the central deflection is

$$w_{max} = 0.01160 \frac{PL^2}{D} \quad (3.83)$$

For the plate at hand, the maximum deflection is

$$w_{max} = 0.01160 \frac{(-10000 \text{ N})(2 \text{ m})^2}{\left[ \frac{(200 \times 10^9 \text{ N/m}^2)(0.2 \text{ m})^3}{12(1 - (0.3)^2)} \right]} = -3.167 \text{ } \mu\text{m} \quad (3.84)$$

The plate was discretized with a 20 x 20 grid, as shown in Figure 3.16. To add simple supports to the edges of the plate, the transverse displacements around the plate's circumference were fixed in MATLAB by the following command:

$$\textit{fixeddofs} = [1 : 3 : 63 \quad 64 : 63 : 1198 \quad 1261 : 3 : 1323 \quad 124 : 63 : 1258]$$

The concentrated force at the center of the plate was added by the command:

$$F(661, 1) = -10000$$

With the BCs set, "FEb" calculated a central deflection of  $-3.180 \text{ } \mu\text{m}$ , which is a 0.42 percent error with Timoshenko's calculation and demonstrates excellent agreement of the model with Timoshenko's plate. Figure 3.17 shows contour plots of the maximum normal, shear, and von Mises stress of the plate, which occurs at  $z = t/2$  (Eqs. (A.11)).

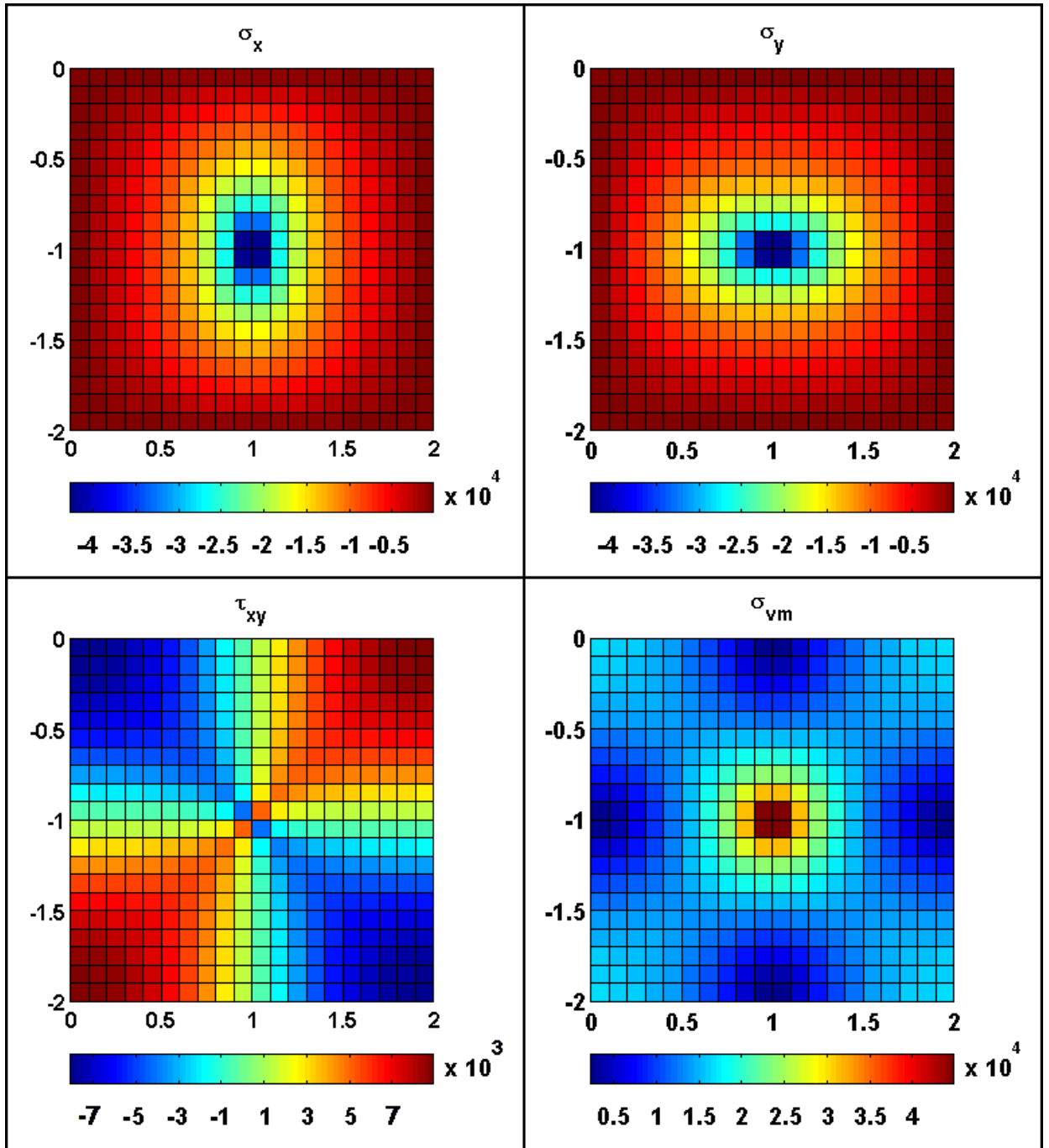


Figure 3.17: Stress Contours for Simply-Supported Bending Plate with Central Deflection

### 3.2.6.3 Membrane/Bending Element. A MATLAB code entitled “FEmb”

is included in Appendix B.3 that implements the superimposed membrane-bending model. The inputs and outputs are the same as they were for both the membrane and the bending model. Both the cantilever beam and the simply-supported plate problems examined in the membrane and bending validations were tested on the combined model. As expected, “FEmb” produced the same deflections for both, accurate to at least four decimal places. To demonstrate the different effects of membrane and bending deformation on stress contours, a simple rectangular geometry with an aspect ratio of 3 is shown in Figures 3.18–3.20. A Young’s modulus of  $200 \times 10^9 \text{ N/m}^2$ , a Poisson’s ratio of 0.3, and a thickness of  $0.1 \text{ m}$  were used. A load of  $10 \text{ kN}$  was applied to the membrane model at the upper right-hand node in the negative  $y$ -direction, a load of  $100 \text{ N}$  was applied to the bending model at the upper right-hand node in the negative  $z$ -direction, and both of these loads were applied to the combined model.

The membrane loading contours of Figure 3.18 can be seen to match those of the membrane loading of the cantilever beam in Figure 3.14, though an even better match would result if Figure 3.14 were more refined in the  $y$ -direction. The neutral axis of the “beam” is easily observable at  $y = -1$ . The bending contours of Figure 3.19 match the bending contours a cantilever beam experiences on its upper and lower surfaces, with a slight “twist” due to non-symmetric loading. In fact, both Figure 3.18 and Figure 3.19 are similar to different cantilever beam contours, where the membrane model reveals stress contours on the side of the beam and the bending model reveals stress contours on the top and bottom of the beam. In the combined loading of Figure 3.20, the applied bending load was scaled down 100 times from the membrane load in order to obtain stress contours that were a good hybrid between membrane and bending contours. This scale factor indicates the dominance of bending loads over membrane loads of the same magnitude. A MATLAB code entitled “FEmb3D” is included in Appendix B.4 that implements the superimposed membrane-bending model and accepts non-planar geometry. Nearly identical stress contours are observed for simple rectangular geometries with the NACA 4-series camber versus flat plate.

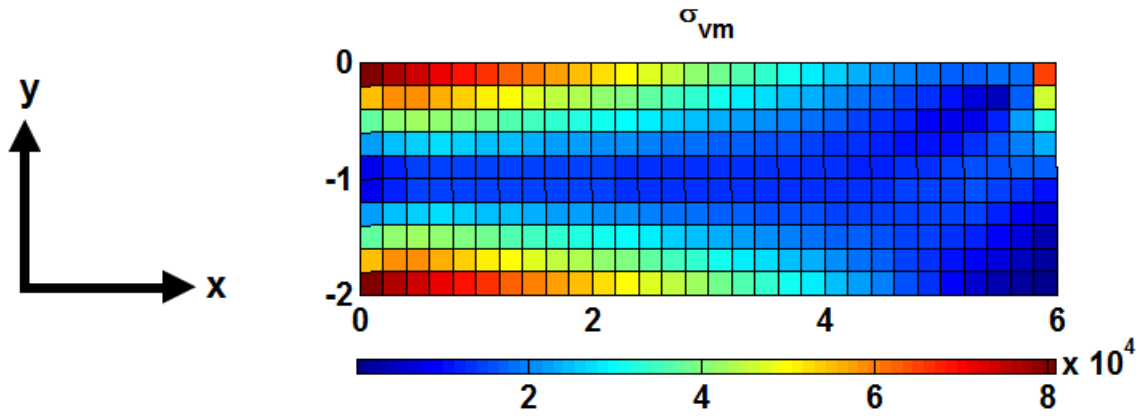


Figure 3.18: Membrane-Bending Model with Only a Membrane Load

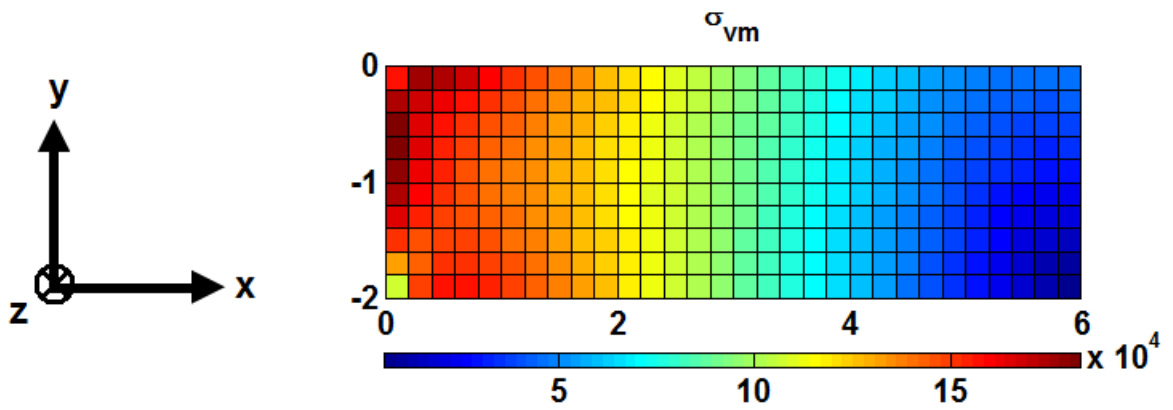


Figure 3.19: Membrane-Bending Model with Only a Bending Load

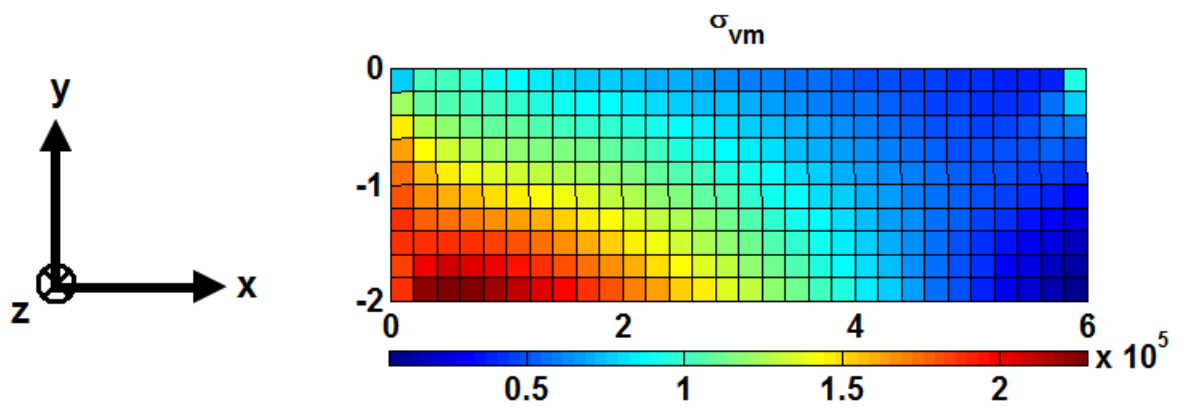


Figure 3.20: Membrane-Bending Model with Combined Loading

### 3.3 Aerodynamics

*3.3.1 Perching Maneuvers.* Section 2.4 explored some research that estimated perch landing maneuvers of birds. A typical altitude versus range profile was given in Figure 2.6. In the interest of capturing wing loads on a MAV during a similar maneuver, perching maneuver data was selected from Reference [38]. Four points along the maneuver were selected from which the aircraft and aerodynamic data were extracted. The four selected points are marked on Figure 3.21–Figure 3.24, which give the data curves of altitude versus range, range versus time, angle of attack versus range, pitch versus range, and velocity versus range. Table 3.3 provides the data estimated from the figures at each point.

The four points are intended to represent the more extreme aerodynamic conditions of the maneuver. Point 1 is at the beginning of the trajectory when the MAV is likely in the  $15^\circ$  back sweep position (Figure 3.6). The dive can be said to have begun since the near  $-100^\circ/\text{s}$  angle of attack and pitch derivatives indicate the vehicle is pitching down fast from its cruising or loitering condition. At Point 2, the MAV is midway through the dive and experiences the peak velocity of the maneuver. The MAV starts to pitch up again, though the pitch angle is still negative. The wings are in the dive configuration (Figure 3.7), much like the those of the eagle owl in Figure 1.2. At Point 3, the MAV is at the bottom of the dive and begins ascending. The

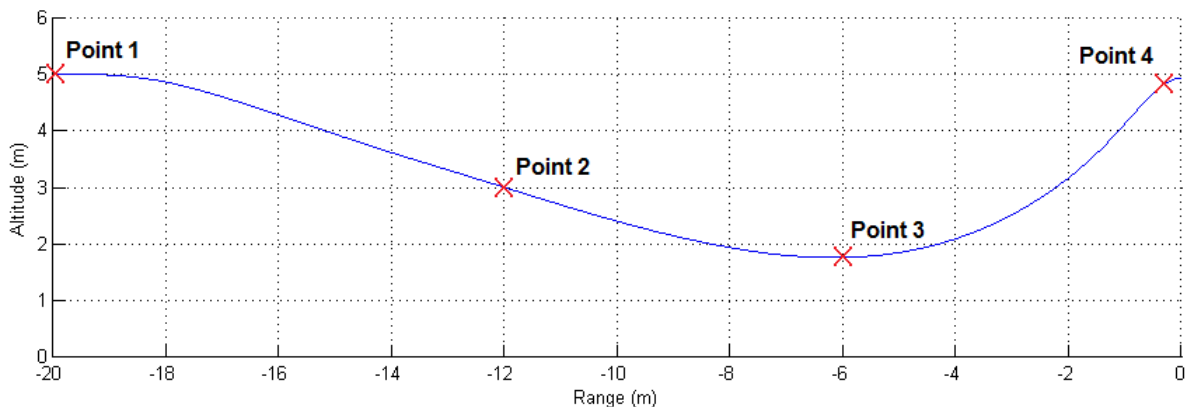


Figure 3.21: Perching Maneuver: Altitude vs. Range

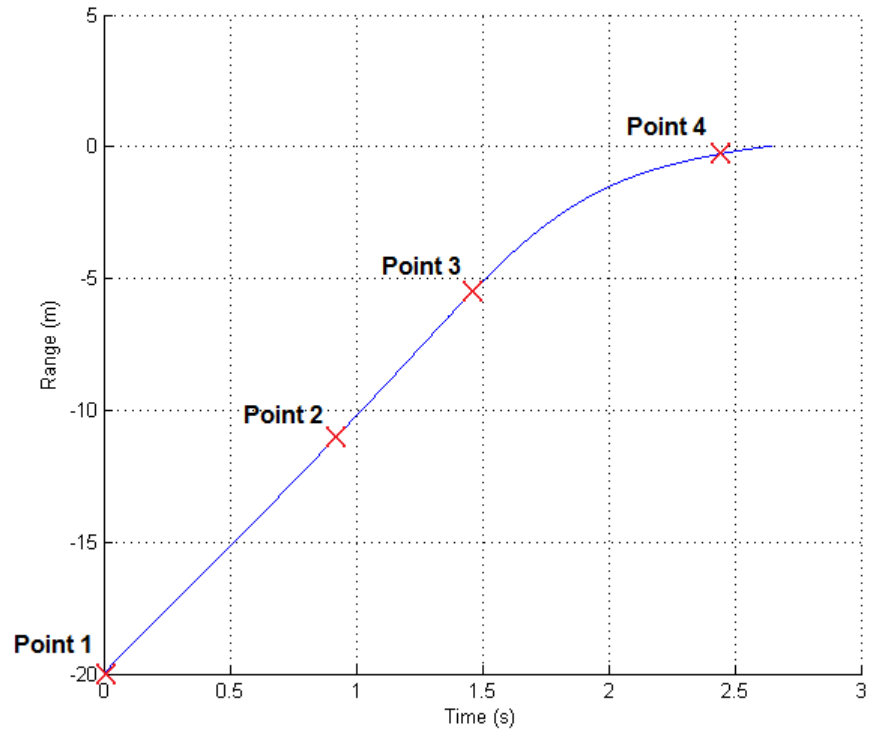


Figure 3.22: Perching Maneuver: Range vs. Time

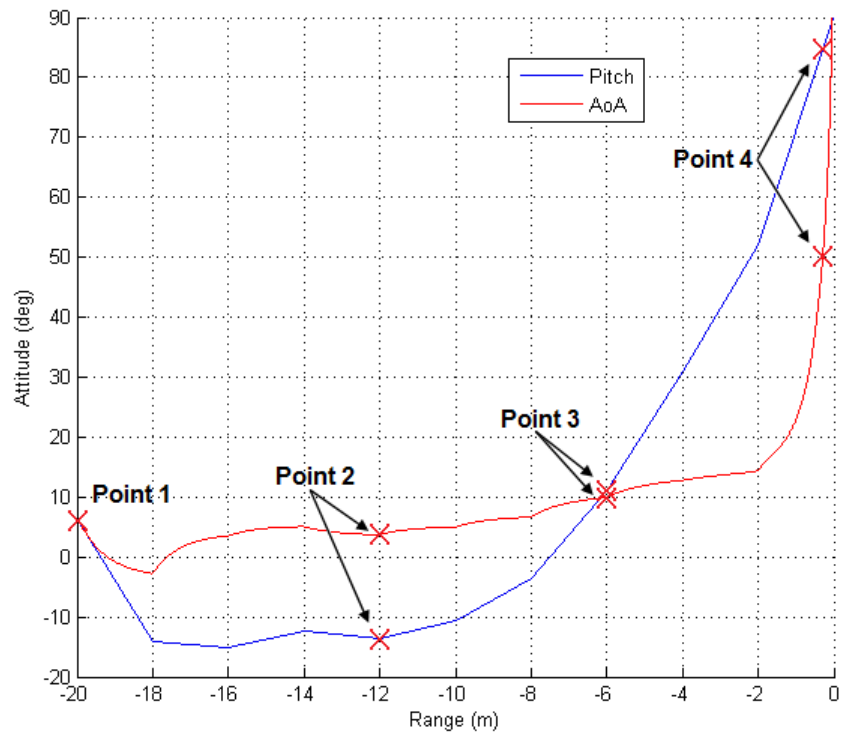


Figure 3.23: Perching Maneuver: Attitude vs. Range

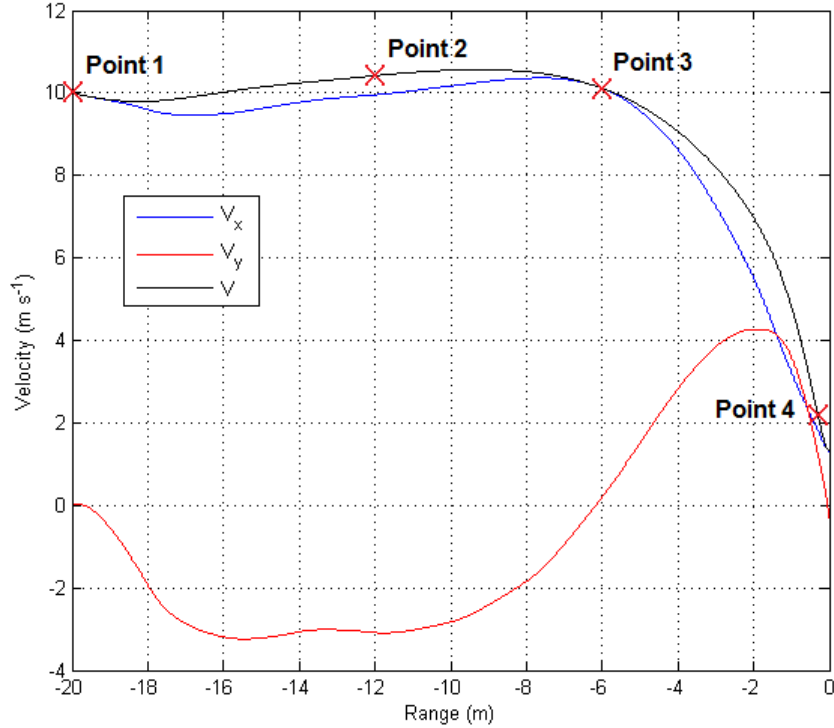


Figure 3.24: Perching Maneuver: Velocity vs. Range

pitch rate is at a maximum due to how fast the MAV is still moving. The wings are transitioning from the dive to forward sweep configuration, and so the configuration with no sweep (Figure 3.5) is selected at this point. Finally at Point 4, the MAV has covered 98.5% of the range and is at a low velocity as it is pulling up into a vertical stance. Consequently the angle of attack is post-stall and the angle of attack rate is at a very high value of  $216^\circ/\text{s}$ . The wings are fully flared with forward sweep (Figure 3.4) so that they can collect as much drag as possible to bring the vehicle to a halt.

Table 3.3: Data for Selected Points along Perching Maneuver

Selected Location:		Point 1	Point 2	Point 3	Point 4
<i>Range</i>	[m]	-20	-12	-6	-0.31
<i>Altitude</i>	[m]	5	3	1.78	4.80
<i>Time</i>	[s]	0	0.92	1.46	2.44
<i>Velocity</i>	[m/s]	10	10.41	10.11	2.19
<i>AOA</i>	[ $^\circ$ ]	6	3.75	10	50
<i>AOA Rate</i>	[ $^\circ/\text{s}$ ]	-98.92	0	14.6	216.2
<i>Pitch Rate</i>	[ $^\circ/\text{s}$ ]	-98.82	0	68.3	26.4

*3.3.2 Force Estimation.* With the geometry and flight conditions selected, a method for estimating the wing loading was established. A vortex lattice program<sup>1</sup> developed in MATLAB called “Tornado” [28] was the product of the Master’s thesis referred to in Reference [29]. Figure 3.25 shows the main menu of Tornado.

The first menu item allows the user to define a new geometry or load one of the many existing wings. A new geometry is built by defining any number of wing partitions in the same manner as Figure 3.1 with the parameters given in 3.1. Thus, the vortex panel meshes and the finite element meshes were created to be identical to one another. The geometries of the fictitious sweeping mechanism shown in Figure 3.4–Figure 3.7 were created in this manner with “flat plate” airfoils. Next, the flight conditions of the four points selected in Table 3.3 were created and saved in menu item 2. The lattice was then generated with menu item 5, where the fixed wake, standard vortex lattice method was selected. Figure 3.26 shows a sample geometry output, and Figure 3.27 shows the vortex panels generated for the sample output.

```
TORNADO Version 135 Release version
build 2010 03 20 14:07 UTC
Main Menu

Input operations.
[1]. Aircraft geometry setup
[2]. Flight condition setup
[3]. Change rudder setting
[4]. Move reference point

Lattice operations.
[5]. Generate lattice.

Computation operations.
[6]. Processor access

Post processing and interactive operations.
[7]. Post processing, Result/Plot functions
[8]. Keyboard access

Auxiliary operations.
[10]. About / Release Info
[100]. Help files
[0]. Exit Tornado

Please enter choice from above:
```

Figure 3.25: Main Menu of Tornado Software

---

<sup>1</sup>The code can be downloaded from the website <http://www.redhammer.se/tornado/DL.html>



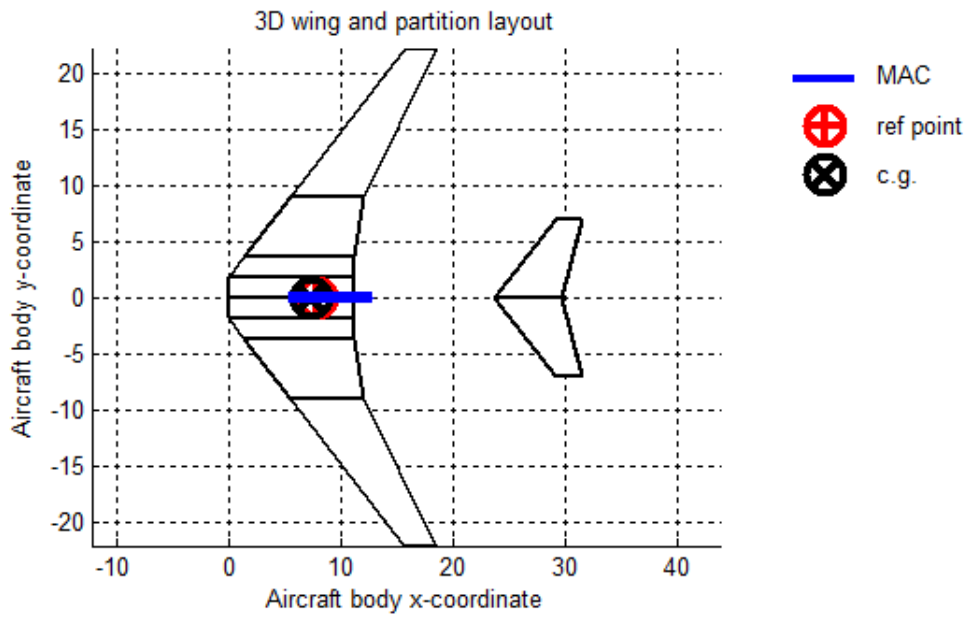


Figure 3.26: Example of Tornado Geometry Output [28]

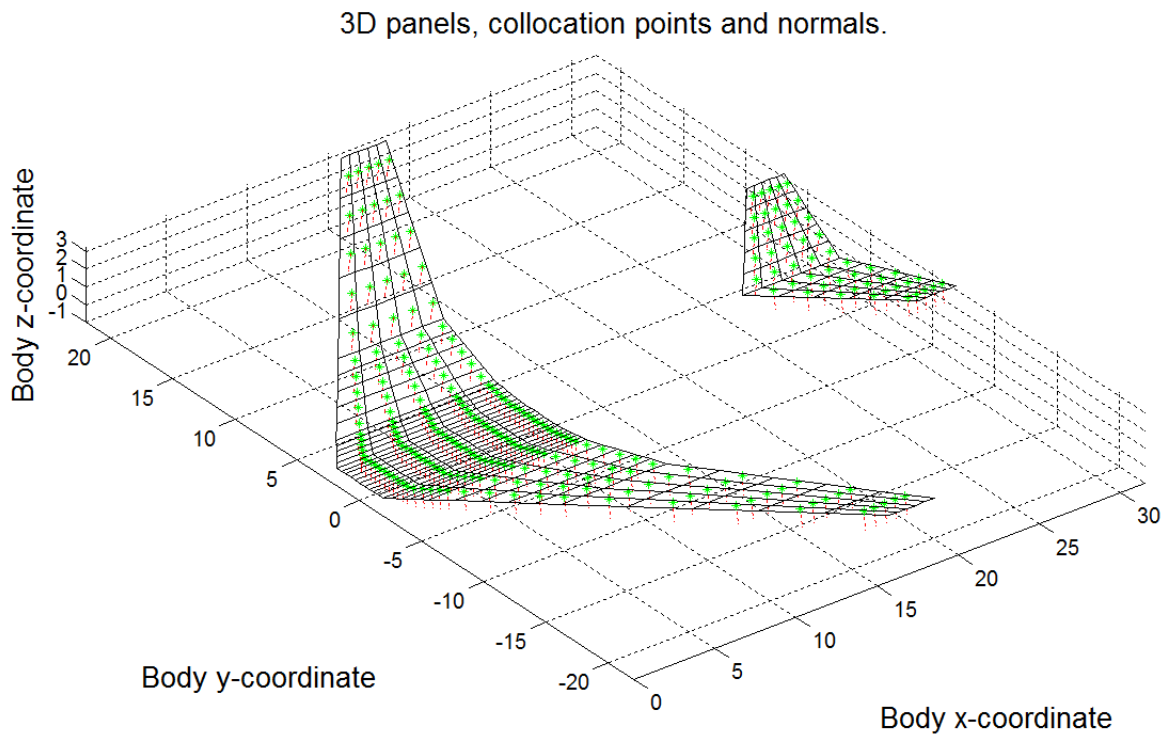


Figure 3.27: Example of Tornado Vortex Panels Output [28]

In light of the several processing options given in menu item 6, only static computations at the specified flight states were performed. Tornado outputs five different MATLAB structures: `geo`, `state`, `lattice`, `ref`, and `results`. The `geo` and `state` structures simply return the geometry and flight conditions defined by the user. The `lattice` structure returns the mesh data. The `ref` structure is not useful, since the mean aerodynamic chord and center of gravity location are only needed for full aircraft analysis. The pertinent data is reported in the `results` structure, which gives three column vectors of the x, y, and z force components acting on each panel. Since vortex lattice methods are inherently inviscid, an alternative method for viscous drag estimation was made. An auxiliary processing option was added to the current version of Tornado to predict zero-lift, flat-plate drag, which gives an estimate of the skin friction drag  $C_{Do}$ . For the perching maneuver, the form drag due to flow separation as the MAV exceeds the stall angle of attack has an even bigger impact on the wing loading, since at  $90^\circ$ , the loading is almost entirely due to drag. (Although it should be noted that that drag loading is still a bending load, as is the lift loading at low angles of attack). Therefore to estimate the form drag as it varies with angle of attack, the following equation was used:

$$D = q_\infty S [C_{Do} + (1 - C_{Do}) \sin \alpha] \quad (3.85)$$

where  $q_\infty$  is the dynamic freestream pressure,  $S$  is the planform area, and  $\alpha$  is the angle of attack. The effect of Eq. (3.85) is shown in Figure 3.28. In this fashion, the drag is exactly equal to the zero-lift drag prediction at  $0^\circ$  angle of attack. At  $90^\circ$  angle of attack or when  $C_{Do}$  is equal to one, the drag is equal to  $q_\infty S$ , as  $C_D$  equal to one implies.

To apply the aerodynamic forces calculated from Tornado to the finite element model, the single resultant force per each panel calculated by the vortex lattice method needed to be divided onto the nodes of the given element. Since the locations of the resultant force acting on each panel were not known, the forces were assumed to act through the centroid of the element. Following this assumption, one fourth of

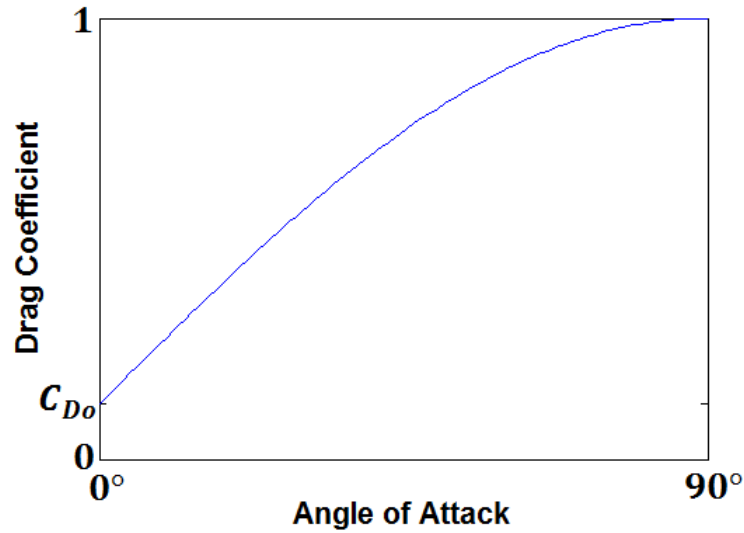


Figure 3.28: Viscous Drag Estimation Curve

each of the three components of the force vector were applied to the four nodes of the element. Thus at each node, one quarter of the force acting through each of the adjacent elements to that node were added together. To distribute the estimated drag force over the nodes, the total estimated drag was first divided among the chords, weighting the division by the local chord length. Then, the component of the total drag for a given chord was distributed evenly among the nodes along that chord. In this manner, the estimated drag force at each node was then added to the lift and inviscid drag loads calculated by Tornado.

### 3.4 Optimization

*3.4.1 Compliance Objective Using the SIMP Model.* In structural topology, the maximum stiffness or rigidity of a structure is often sought by distributing a limited amount of material throughout the design domain. In this pursuit, a common restatement of the maximum stiffness objective is minimization of flexibility. The basic relationship of flexibility to rigidity can be seen with the fundamental linear spring in elementary mechanics, where the flexibility is simply the reciprocal of the spring stiffness.

In order to minimize the flexibility of a structure, a suitable measure of flexibility must be selected to formulate the objective of an optimization problem. Though there are a number of measures of flexibility, the compliance  $c$  of a structure will be used here. Compliance can be thought of as the strain energy [12, p. 116] in a deformed solid body. Provided no energy is gained or lost in the form of heat, the work required to deform a body is equivalent to the strain energy of that body in its deformed state. Work is, of course, the product of force and displacement. Thus, for a discrete finite element formulation, compliance is simply  $\{F\}^T\{d\}$ , where  $\{d\}$  are the nodal displacements and  $\{F\}$  are the external forces at the nodes (see References [8, pp. 1–9] and [10, pp. 179–180] for a continuous formulation of compliance). Using compliance as a measure of flexibility has the advantage that it is a convex function of the design variables, whereas a measure such as  $\{d\}^T\{d\}$  is a nonconvex function of the design variables. Also, a body where compliance has been minimized will have the same specific strain energy in each element (discussed later).

Though the flexible skin process summarized in Section 2.3 suggested that skin design should include Young’s modulus, shear modulus, thickness, and possibly Poisson’s ratio in the design variables, only element thickness is considered here in the design variables as a simpler starting point. Thus, for a finite element discretization with  $N$  elements, the design variables are given in the vector

$$\{\rho\} = \left\{ \rho_1 \quad \rho_2 \quad \dots \quad \rho_e \quad \dots \quad \rho_N \right\}^T \quad (3.86)$$

where  $\rho_e$  is the thickness of element  $e$ . It will be discussed later whether only including element thickness in the design variables leads to results that can be correlated to skin design.

With the stated objective and design variables, the design constraints must now be specified. The nodal displacements  $\{d\}$  in the objective function will be solved by the finite element method (Eq. (3.75)), and therefore a simultaneous optimization

formulation includes Eq. (3.74), repeated here, as a constraint.

$$\{F\} = [K]\{d\} \quad (3.87)$$

Substituting in Eq. (3.72) in for the global stiffness  $[K]$  gives

$$\{F\} = \left( \sum_{e=1}^N [k_e] \right) \{d\} \quad (3.88)$$

As stated in the opening sentence of this section, compliance minimization is only beneficial if a limit on the material distributed throughout the design domain  $\Omega$  is imposed. In the absence of such a constraint, compliance minimization would simply lead to infinite thickness throughout the entire design domain; but with a limited volume  $V$  of material, the optimization must “economize” the material distribution. Expressing the material constraint as an inequality gives:

$$\sum_{e=1}^N \rho_e a_e \leq V \quad (3.89)$$

where  $a_e$  is the area of element  $e$ . In matrix form, this constraint is

$$\{\rho\}^T \{a\} - V \leq 0 \quad (3.90)$$

Additionally, for a physical solution to occur, the design variables must be confined by an upper and lower bound.

$$0 < \rho_{min} \leq \rho_e \leq \rho_{max}, \quad e = 1, \dots, N \quad (3.91)$$

Though  $\rho_{min}$  and  $\rho_{max}$  can be considered inequality constraints, most optimization solution methods take into account optimization variable bounds without explicitly adding them as constraints. Design variable bounds are sometimes referred to as “box” constraints.

Putting the objective function and constraints together, a simultaneous compliance minimization with two constraints is given by

$$\begin{aligned}
& \min_{\boldsymbol{\rho}, \mathbf{d}(\boldsymbol{\rho})} \{F\}^T \{d\} \\
& \text{s.t.} \quad \left( \sum_{e=1}^N [k_e] \right) \{d\} = \{F\}, \\
& \quad \quad \{ \rho \}^T \{ a \} - V \leq 0, \quad 0 < \rho_{min} \leq \rho_e \leq \rho_{max}, \quad e = 1, \dots, N
\end{aligned} \tag{3.92}$$

A better programmatic formulation of the problem is to nest the equilibrium constraint in the compliance objective, thus making the objective statement a function in the design variables only, and not in  $\mathbf{d}(\boldsymbol{\rho})$  as well. This also reduces the number of constraints down to the single material constraint.

$$\begin{aligned}
& \min_{\boldsymbol{\rho}} c(\boldsymbol{\rho}) \\
& \text{s.t.} \quad \{ \rho \}^T \{ a \} - V \leq 0, \quad 0 < \rho_{min} \leq \rho_e \leq \rho_{max}, \quad e = 1, \dots, N
\end{aligned} \tag{3.93}$$

where the compliance  $c$  is given as

$$c(\boldsymbol{\rho}) = \{F\}^T \{d\}, \quad \text{where } \{d\} \text{ solves:} \quad \left( \sum_{e=1}^N [k_e] \right) \{d\} = \{F\} \tag{3.94}$$

While the values of the design variables are allowed to range between  $\rho_{min}$  and  $\rho_{max}$ , the optimization problem of Eq. (3.93) is really a sizing problem, as defined in Section 1.2. In a topology design, the optimal placement of a material within the design domain  $\Omega$  is desired such that points in space will either have material or they will be void. In this case, the structural geometry is rendered as a black and white image, albeit pixelated due to the finite element discretization. Thus topology optimization is seeking to determine the optimal subset  $\Omega^{mat}$  of material points. Ideally, the element stiffness matrices would be defined piecewise over the

domain  $\Omega$  in the following manner:

$$[k(x)] = 1_{\Omega^{mat}}(x)[k_e(x)], \quad 1_{\Omega^{mat}}(x) = \begin{cases} 1 & : x \in \Omega^{mat} \\ 0 & : x \in \Omega \setminus \Omega^{mat} \end{cases} \quad (3.95)$$

where  $[k_e(x)]$  indicates that the element stiffness matrix varies over the domain. In this distributed, discrete 0-1 problem, the material constraint is represented as

$$\int_{\Omega} 1_{\Omega^{mat}} d\Omega = Vol(\Omega^{mat}) = V \quad (3.96)$$

Unfortunately, a number of problems arise with the character of a discrete problem. Allowing finite elements to have zero stiffness leads to singularities in the solution process. Also, optimization algorithms where design variables take only discrete values are not efficient for large-scale problems with lots of variables. Thus, a model that allows intermediate values in the design variables but also penalizes them, the so-called Solid Isotropic Material with Penalization (SIMP) model, is used.

In the SIMP model, the constitutive matrix is directly penalized. For the membrane element, the constitutive matrix (Eq. (3.23)) takes the form

$$[D] = \frac{\rho^q Et}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix} \quad (3.97)$$

where the penalty  $q$  is some constant (typically  $\geq 1$ ). Since the upper bound  $\rho_{max}$  is set to 1, the physical thickness  $t$  is still included in the constitutive matrix such that when an element is fully present ( $\rho = 1$ ), the actual physical thickness is scaled by  $t$ . The quantity  $\rho^q$  can be thought of as the “effective” thickness. To avoid numerical problems induced by zero stiffness, the lower bound on the thickness is  $\rho_{min} = \varepsilon \approx 0$  ( $\varepsilon = 0.001$  is typical). Figure 3.29 shows the effect the choice of exponent  $q$  has on penalization. When  $q = 1$ , the stiffness of an element is proportional to the amount of material designated to that element, and hence the thicker an element is, the stiffer

it becomes. This is not the case for  $q > 1$ . Putting some values to Figure 3.29, at half the available material ( $\rho = 0.5$ ), the effective thickness is only about 0.2 for  $q = 3$ . In this case, the effective thickness does not reach one half until a  $\rho$  of about 0.8. What choice of exponent  $q$  performs the best is a matter of numerical experimentation.

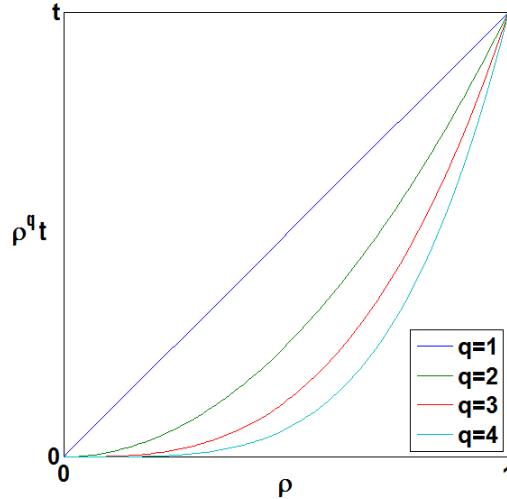


Figure 3.29: SIMP Penalization of Element Thickness

However, Bendsøe and Sigmund point out [8, p. 7] that in order for a material to be realizable from the a 2-D SIMP model,  $q$  must satisfy the following relationship with Poisson's ratio:

$$q \geq \max \left\{ \frac{2}{1 - \nu}, \frac{4}{1 + \nu} \right\} \quad (3.98)$$

Thus, for a material with  $\nu = 1/3$ , the smallest  $q$  is 3. In fact,  $q = 3$  has been found to be a good choice for the exponent, and will be used here unless otherwise stated. For a plate bending element, the constitutive matrix (Eq. (3.36)) reveals that the element stiffness already has cubic dependence on the thickness. This implies that the optimal design naturally prefers to achieve either the upper or lower bound on thickness, and an additional penalty need not be imposed (consult Reference [8, p. 58] for more information).

Rather than keep the design variables in the constitutive matrix, the global stiffness matrix can be more readily seen as a function of the design variables when



they are placed outside of the element stiffness matrix:

$$[K] = \left( \sum_{e=1}^N \rho_e^q [k_e] \right) \quad (3.99)$$

With the amendment to the compliance objective with the SIMP model, the optimization statement is modified slightly.

$$\begin{aligned} \min_{\boldsymbol{\rho}} \quad & c(\boldsymbol{\rho}) \\ \text{s.t.} \quad & \{\boldsymbol{\rho}\}^T \{a\} - V \leq 0, \quad \varepsilon \leq \rho_e \leq 1, \quad e = 1, \dots, N \end{aligned} \quad (3.100)$$

where the compliance  $c$  is given as

$$c(\boldsymbol{\rho}) = \{F\}^T \{d\}, \quad \text{where } \{d\} \text{ solves: } \left( \sum_{e=1}^N \rho_e^q [k_e] \right) \{d\} = \{F\} \quad (3.101)$$

Once  $\{d\}$  has been calculated from the finite element method, a substitution can be made for the global force vector  $\{F\}$  such that

$$c(\boldsymbol{\rho}) = \{d\}^T \left( \sum_{e=1}^N \rho_e^q [k_e] \right)^T \{d\} \quad (3.102)$$

The total compliance can be computed by summing up the compliance of each element, much in the same way that the direct stiffness method adds element stiffnesses and forces together to obtain the global stiffness and force vectors. Thus Eq. (3.102) becomes

$$c(\boldsymbol{\rho}) = \sum_{e=1}^N c_e(\rho_e) = \sum_{e=1}^N \rho_e^q \{d_e\}^T [k_e]^T \{d_e\} \quad (3.103)$$

The Karush-Kuhn Tucker (KKT) optimality conditions [7, p. 130], which give the first-order necessary conditions, are now derived for the compliance minimization of Eq. (3.100). The Lagrangian function is given as

$$\mathcal{L}(\boldsymbol{\rho}, \lambda) = c(\boldsymbol{\rho}) + \lambda (\{\boldsymbol{\rho}\}^T \{a\} - V + s^2) \quad (3.104)$$

where  $\lambda$  is the Lagrange multiplier for the single volume constraint, and  $s$  is the slack variable for the inequality constraint. A formal derivation of the compliance gradient with respect to the design variables is given in Reference [10, pp. 98–99], where both a direct analytical method and the adjoint analytical method are employed to attain the following:

$$\frac{\partial c_e}{\partial \rho_e} = 2\{d_e\}^T \frac{\partial \{f_e\}}{\partial \rho_e} - \{d_e\}^T \frac{\partial [k_e]^T}{\partial \rho_e} \{d_e\} \quad (3.105)$$

The first term in Eq. (3.105) is only included when the global force vector is a function of the design variables, for example, when the weight of the structure is included in the loading. With the modification of  $[k_e]$  by the SIMP model and excluding thickness dependency of the loading, the compliance sensitivity reduces to

$$\frac{\partial c_e}{\partial \rho_e} = -q\rho_e^{q-1} \{d_e\}^T [k_e]^T \{d_e\} \quad (3.106)$$

The gradient condition for the design variables thus becomes

$$\frac{\partial \mathcal{L}}{\partial \rho_e} = -q\rho_e^{q-1} \{d_e\}^T [k_e]^T \{d_e\} + \lambda \rho_e a_e = 0, \quad e = 1, \dots, N \quad (3.107)$$

and the volume constraint is recovered from the gradient condition

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \{\rho\}^T \{a\} - V + s^2 = 0 \quad (3.108)$$

A feasibility check on the inequality requires that

$$s^2 \geq 0. \quad (3.109)$$

The switching condition are derived from the derivative of the Lagrangian function with respect to the slack variable.

$$\lambda s = 0 \quad (3.110)$$

Finally, the lagrange multiplier is required to be nonnegative since it is associated with an inequality.

$$\lambda \geq 0 \tag{3.111}$$

The KKT conditions produce enough equations for the unknowns and, in theory, the equations could be solved simultaneously. However, the nested compliance optimization problem has been shown to be convex [10, p. 86] (although the simultaneous formulation is not) and can therefore take advantage of convex algorithms.

*3.4.2 Optimality Criteria Method.* Though there are many ways to solve the compliance optimization problem, a general procedure for solving the nested formulation using first order algorithms is described in the following steps:

1. Start with an initial design  $\boldsymbol{\rho}^0$  and set the iteration counter  $k = 0$ . A typical initial design is  $\boldsymbol{\rho}^0 = Vol(\Omega^{mat})/Vol(\Omega)$ .
2. Perform a finite element analysis for the current design, solving for the displacement vector  $\mathbf{d}(\boldsymbol{\rho}^k)$  from the equilibrium condition  $\mathbf{K}(\boldsymbol{\rho}^k)\mathbf{d}(\boldsymbol{\rho}^k) = \mathbf{F}$ .
3. For the current iteration, calculate the compliance  $c(\boldsymbol{\rho}^k)$ , the constraint function  $(\boldsymbol{\rho}^k)^T \mathbf{a} - V$ , and the corresponding sensitivities.
4. Evaluate an appropriate update scheme (i.e. by solving an explicit, convex approximation) to give a new design  $\boldsymbol{\rho}^{k+1}$ .
5. Update the counter  $k = k + 1$  and return to step 2 unless a stopping criteria has been met.

Step 4 provides leeway in selecting an update scheme for the design variables and indeed, there are many, such as SLP [7], SQP [7], CONLIN [10], and MMA [42]. However, the Optimality Criteria (OC) method is a historically older, classical approach that can be seen as a special case of the explicit convex approximation method, and has turned out to be very efficient at solving topology optimization problems.

A more descriptive presentation of the OC method is given in Reference [10], but only a brief description is presented here. The OC method takes advantage of the fact that the material volume is a monotonically decreasing function of the Lagrange multiplier  $\lambda$ . Hence, a stationary point is achieved when the volume constraint is satisfied. The update of the design variables is calculated by

$$\rho_e^{k+1} = \min \left\{ \max \left[ \rho_e^k \left( \frac{q\rho_e^{q-1}(\mathbf{d}_e^k)^T \mathbf{k}_e^T \mathbf{d}_e^k}{\lambda a_e} \right)^\eta, \rho_{min} \right], \rho_{max} \right\} \quad (3.112)$$

such that the volume constraint

$$\sum_{e=1}^N a_e \rho_e^{k+1}(\lambda) - V = 0 \quad (3.113)$$

is satisfied, where  $\rho_e^{k+1}$  is now a function of  $\lambda$ . The update scheme can be written a little more compactly by substituting the compliance sensitivity (Eq. (3.106)) into Eq. (3.112).

$$\rho_e^{k+1} = \min \left\{ \max \left[ \rho_e^k \left( -\frac{\partial c_e / \partial \rho_e}{\lambda a_e} \right)^\eta, \rho_{min} \right], \rho_{max} \right\} \quad (3.114)$$

In this manner, each thickness is updated independently. The min and max functions are used so that the box constraints are not violated. The exponent  $\eta$  acts like a damping knob on the update process and is chosen by experiment (a value of 0.5 is typical). Note that the design has converged when the quantity in parenthesis is equal to unity, or if

$$\frac{q\rho_e^{q-1}(\mathbf{d}_e^k)^T \mathbf{k}_e^T \mathbf{d}_e^k}{a_e} = \lambda = \text{constant}, \quad \text{for all } e = 1, \dots, N \quad (3.115)$$

This quantity is twice the specific strain energy of an element. The implication of Eq. (3.115) is that the OC method is closely related to a fully stressed design, where all elements have the same strain energy, and therefore the same stress. However, with the penalization method, the specific strain energy is only constant for elements with intermediate densities, since it is lower in regions with  $\rho = \rho_{min}$  and higher in

regions with  $\rho = \rho_{max}$ . The design update does however add material to elements with a specific strain energy higher than  $\lambda$ , and it removes material from elements with a specific strain energy lower than  $\lambda$ , assuming the bounds on  $\rho$  are not violated.

A bisectioning routine is used to implement Eqs. (3.114) and (3.113). First, an upper and a lower bound on the Lagrange multiplier are selected, for example,  $\lambda_{upp} = 100,000$  and  $\lambda_{low} = 0$ . Next, the middle point between those bounds is set as  $\lambda_{mid}$ . The update is computed with Eq. (3.114) using  $\lambda_{mid}$ . The volume constraint is then checked with the updated variables. If the constraint is violated with excessive material, then the lower bound is set to  $\lambda_{mid}$ , and if the constraint is violated with an inadequate amount of material, then the upper bound is set to  $\lambda_{mid}$ . This process continues until the value of  $\lambda$  is found within a certain tolerance which yields a design update that satisfies the volume constraint.

Included in Appendix C are MATLAB implementations of the compliance optimization with the SIMP model, which use the OC method that has been described. Four variants of the code are provided which are based on the four finite element models developed in Section 3.2. These “SIMP” codes, as they are named, are structured in the same fashion as a 99-line topology optimization code for compliance minimization<sup>2</sup> [8, App. 5.1] written by Ole Sigmund. They receive as inputs the “params” geometry structure (see Section 3.1), as well as some other outputs from an aerodynamics program (described in the following section). An example output of “SIMPm” is shown in Figure 3.30 where the rectangular design domain topology, representing the right wing of an aircraft, has been optimized for an in-plane load applied at the upper, right-hand corner of the domain. The boundary conditions are fixed such that the root chord is clamped.

As a final note, several complications can arise with the formulation given in this section. A common one that severely inhibits a good solution is that of the so-called checkerboard pattern. A checkerboard pattern occurs when the design func-

---

<sup>2</sup>The MATLAB code can be downloaded from the website <http://www.topopt.dtu.dk>

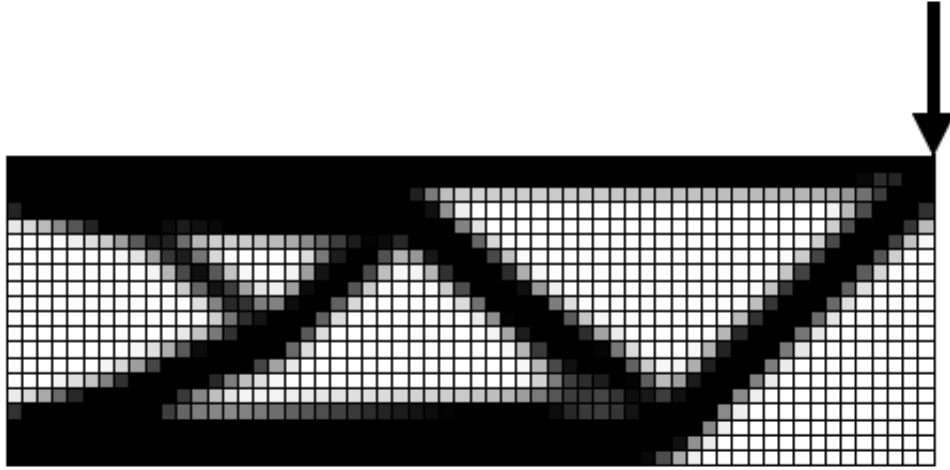


Figure 3.30: Example “SIMP” Output

tion alternates between 0 and 1 and produces a numerically artificial stiffness. This problem is described in detail in Reference [8, pp. 39–47]. Sigmund devised a mesh-independency filter in his 99-line code for the compliance sensitivities that also eliminates the checkerboard pattern. Only one additional parameter used for specifying the radius of the filter is added to the method. A radius of less than 1 indicates the filter is turned off. This filter is included in the “SIMP” codes in Appendix C. The problem in Figure 3.30 is repeated again in Figure 3.31, displaying the erroneous results that occur when the checkerboard filter is turned off.

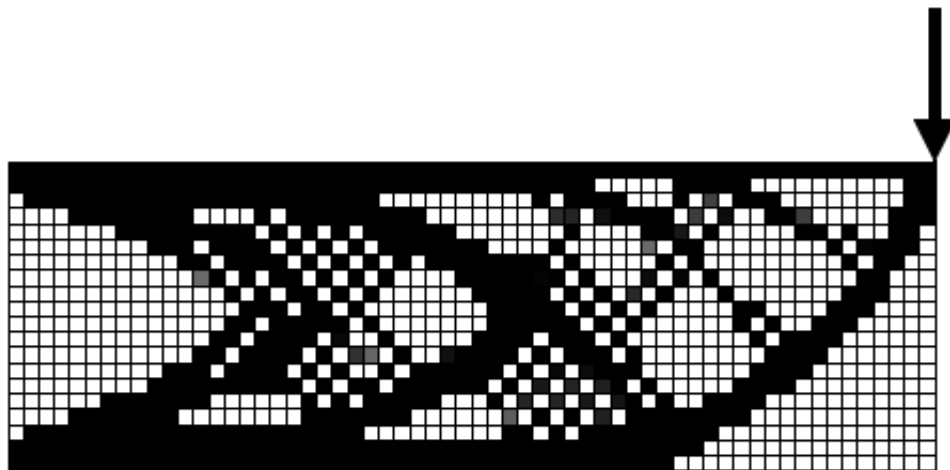


Figure 3.31: Example “SIMPm” Output with No Checkerboard Filter

*3.4.3 Parameter and Convergence Study.* Before investigating the real geometries and loading scenarios of interest, the influence of the SIMP model parameters on determining structural layout, and the effect of the OC parameters on convergence rate are studied for a simple rectangular domain. This should lead to a good working knowledge of the parameters, which will help shorten the computation time for the larger cases of particular interest, and develop a sense of the influence of the parameters in the solutions. A reference problem is shown in Figure 3.32, where a 60 x 30 discretization of a rectangular geometry with an aspect ratio of 2 has been selected. The reference boundary conditions consist of a single point load in the negative y-direction and clamped nodes along the root chord.

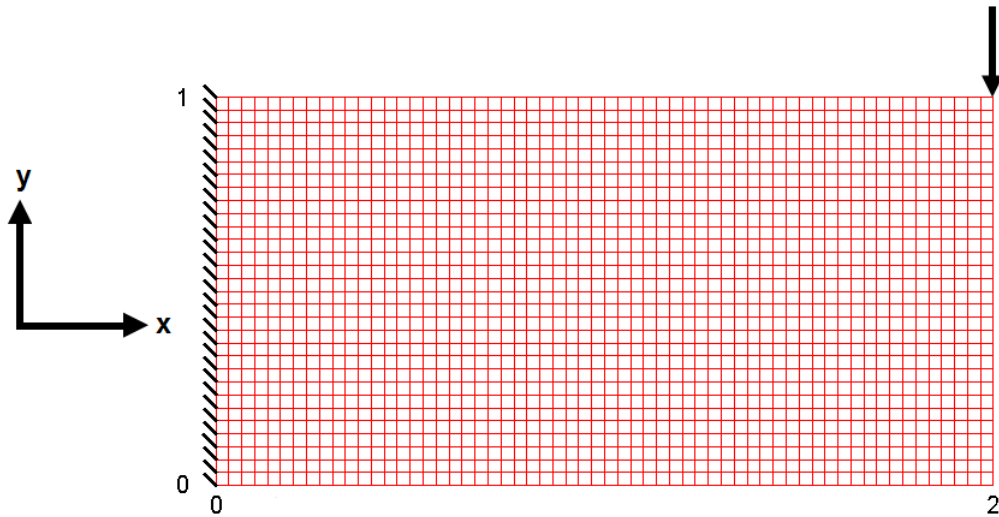


Figure 3.32: Reference Problem with a 60 x 30 Mesh

*3.4.3.1 Varying the SIMP Parameters.* As was shown in Figure 3.29, the penalization of the SIMP model is entirely controlled by the exponent  $q$ , which is some number greater than 1. Selecting a higher value of  $q$  penalizes the element thicknesses to a greater extent, and thus a “sharper” solution with fewer intermediate thickness values results. However, selecting an excessively high value for  $q$  leads to singularity in the finite element solution. In Figure 3.33, the reference problem of Figure 3.32 is solved six different times within a convergence of 10%, each with

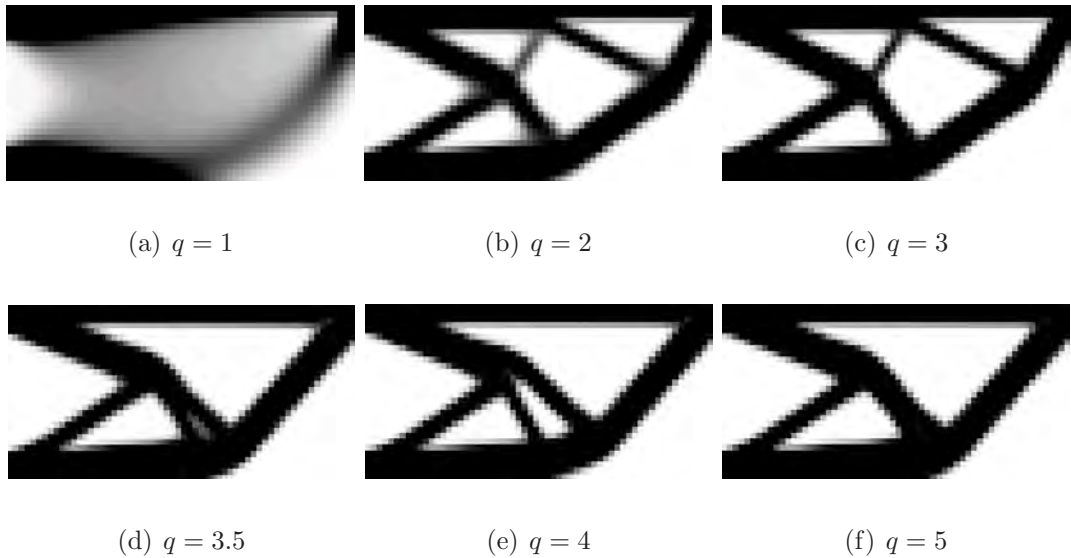


Figure 3.33: Membrane Structures for Reference Problem with Varying Penalizations ( $Vol_f = 0.5$ ,  $r_{filter} = 1.5$ )

different penalizations. Figure 3.33(a) demonstrates the inability of  $q = 1$  to drive the domain to a discrete structure. For penalties of 2 and 3, the reference problem converged to the same structure, although the latter converged in half the iterations that the former did. Unlike Figures 3.33(b) and (c), Figures 3.33(d)–(f) reveal a structure with fewer and bulkier members. As the penalization increases from 3 to 3.5, the upper two smaller internal members vanish and their material is added to the remaining internal members. This process occurs as the solution progresses. Hence, there is a penalty (about 3.5) at which the basic layout of the structure crosses over from that of Figure 3.33(b) to that of Figure 3.33(d). With this new layout, the solution of Figure 3.33(e) begins to split a main internal member into two members, but this gap is ultimately not maintained as the penalty increases to 5. Thus, it can be concluded from Figure 3.33 that there may be a penalization value (about 3 to 4) at which the basic layout of a truss-like structure reduces the total number of members and uses the material to thicken fewer members. Also, a penalty less than three will likely require significant additional iterations to reach the same point of convergence.



Equally important as the penalization parameter is the volume fraction  $Vol_f$ , which refers to the fraction of the design domain to be filled with material. Figure 3.34 shows the evolution of the reference problem structure with increasing volume fractions, for a penalty of  $q = 3$ . With increasing material at the disposal of the structure, two types of changes can occur—the current members can thicken, or new members can be added. For the problem of Figure 3.34, increasing the volume fraction from 30 to 40 percent does not add any new members, whereas the progression of the other volume fractions both thickens current members and adds new members. The similarity between Figure 3.34(c) and Figure 3.33(f) indicate that the same structural configuration (though not necessarily thickness of members) may be approached by increasing the penalization parameter, or by decreasing the volume fraction. Volume fractions greater than 0.5, while interesting, probably lead to an inefficient structure, whereas volume fractions as low as 0.2 may certainly be worthy of investigation. A designer will likely select the lowest volume fraction that still yields satisfactory compliance or maximum displacement of the structure.

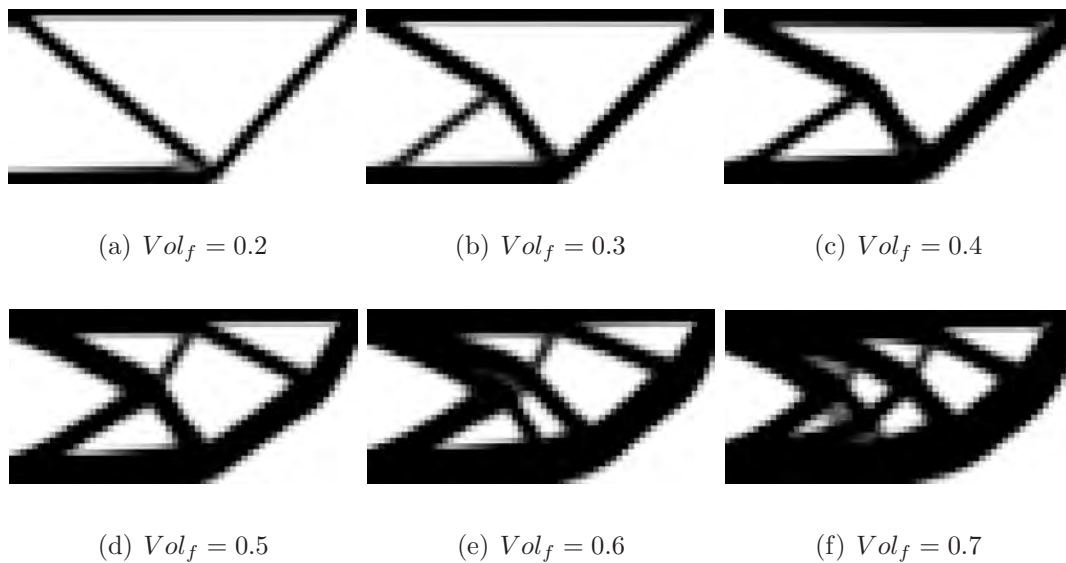


Figure 3.34: Membrane Structures for Reference Problem with Varying Volume Fraction ( $q = 3$ ,  $r_{filter} = 1.5$ )

One other parameter of the SIMP model warrants investigation, namely the radius of the checkerboard filter  $r_{filter}$ , mentioned briefly at the end of 3.4.2. The filter removes the numerically stiff checkerboard pattern that can artificially appear (Figure 3.31), but the filter also has the effect of eroding the number of members in a structure and ultimately “blurring” the image when the radius is overextended. Figure 3.35 repeats the reference problem for six different filter radii, where  $q = 3$  and a volume fraction of 0.5 have been used. Aside from blurring the structure, increasing the filter radius seems to produce a similar effect as increasing the penalization. Whereas the filter is clearly overextended when the radius exceeds a value of 2, the structure is also rather coarse when the radius is less than a value of 1.25. Therefore a filter radius ranging from about 1.25 to 2 seems to result in the best structural image.

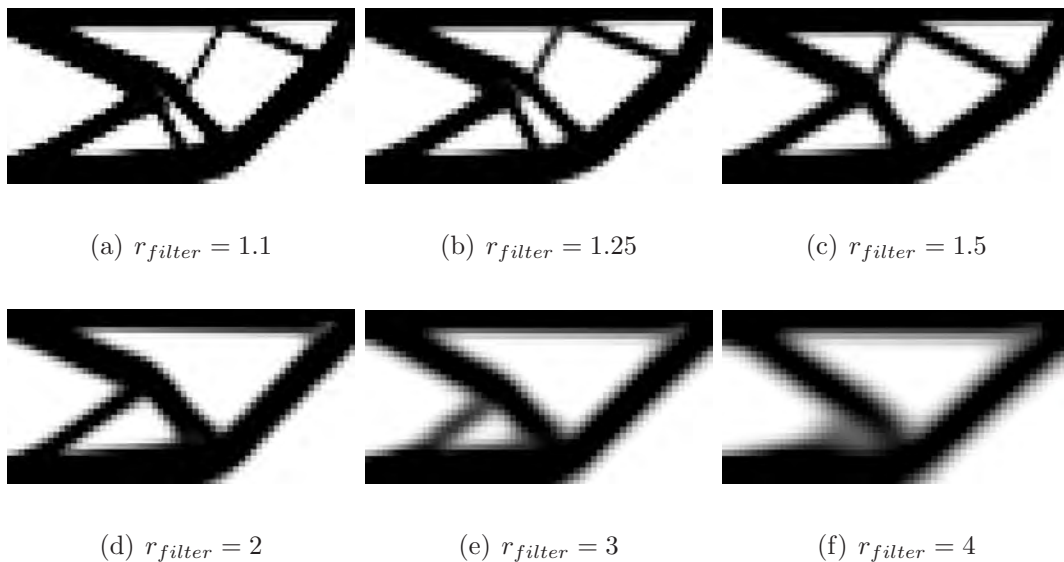


Figure 3.35: Membrane Structures of Reference Problem with Varying Filter Radii

*3.4.3.2 Varying the Convergence Parameters.* The goal of conceptual wing structure design is to qualitatively determine the topology or anatomy assumed within a design domain under specified boundary conditions. Once the optimization has ended, the resulting structure must still be post-processed to transfer the structure to the next design level. Post-processing will require an interpretation scheme which

may modify the geometry of the structure. Thus, a topology need not meet a tight convergence criteria, but may be considered sufficiently converged within a 0.01 or even 0.02 change between iterations. The change from one iteration to the next is a measure of the maximum difference of the thickness between any one element.

To avoid an unnecessary degree of convergence, the change in the objective value and in the structural layout of the reference problem through a number of iterations is studied. Figure 3.36 shows the reference problem initially at five iteration increments, and then at iteration 50 and 298, at which point the thicknesses have converged within a 0.01 change. After 20 iterations, the same general layout has emerged as the final layout of iteration 298. The only change after an additional 278 iterations is a slight shift of the two smaller internal members to the left and the consequent minor bending of the right perimeter member. Table 3.4 provides the convergence data for iteration

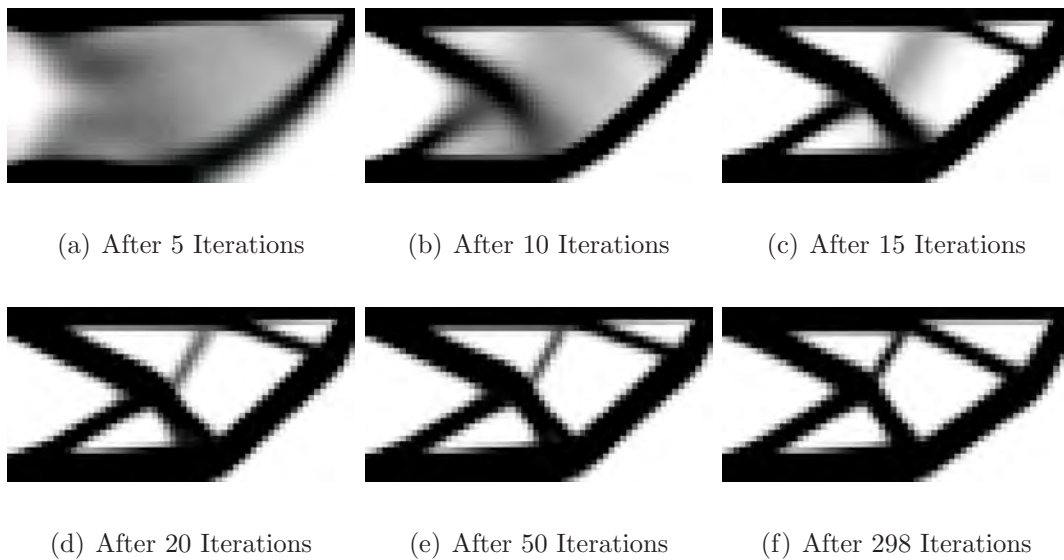


Figure 3.36: Membrane Structure of Reference Problem at Progressing Iterations

0 (where the thickness of the elements is uniformly distributed throughout the design domain), the six iterations of Figure 3.36, and for iteration 996, where the change reaches a value of 0.001. Clearly, the additional 698 iterations needed to bring the change from 0.010 to 0.001 is needless, since the objective value actually increases. The extreme convergence requirement of 0.001 drives the objective value slightly higher

Table 3.4: Convergence Data for Figure 3.36

Iteration	Objective	% Diff. (Obj.)	Change
0	360.8092	—	—
5	117.1957	101.9	0.200
10	89.3646	26.9	0.195
15	77.7898	13.8	0.196
20	74.4566	4.4	0.066
50	73.9472	0.7	0.011
298	72.9587	1.3	0.010
996	73.0748	-0.2	0.001

just to find a solution that changes so minutely. Although a change of 0.011 is achieved as early as 50 iterations, a change of 0.010 is not attained for another 248 iterations, during which the objective function value is reduced an additional 1.3%.

Table 3.4 is not intended to give the impression that the change from one iteration to the next is always decreasing, because the change typically rises and decreases several times throughout the optimization, as members early on in the process are removed. Converging within a change of 0.01 is generally acceptable for purposes of rendering the basic structural concept, and even converging within a change of as high as 0.030 or 0.040 may likewise bring the objective function value to a low percent difference of 1–2%. The structural layout generally changes very little after 30–50 iterations. A good change criterion may be said to be met when the solution consistently lingers (about 10–20 iterations) at a change value of about 0.01.

The rate of convergence can be influenced through two different parameters—the move limit and the damping  $\eta$ . Although both parameters affect the update of the design variables, only the damping parameter is present in Eq. (3.114). The update scheme of Eq. (3.114) identifies the upper and lower bounds on the design variables with the use of the min and max functions, but the design variables can be further bounded by imposing another set of bounds. The so-called move limit is the maximum amount that a thickness can change between iterations. Both additional move limits and the damping parameter can be assets to the convergence rate, and thus the effect of varying them is explored in Table 3.5 and Table 3.6.

Table 3.5 shows the effect of varying the move limit from a value of 0.05 to 0.8 and for  $\eta = 0.5$  (recommended value in Reference [8]). Each solution converges to within a change of 0.01, and minimizes the objective function to an average value of 72.98 with a standard deviation of 0.01%. The resulting topologies were all virtually indistinguishable from those of Figures 3.36(d)–3.36(f), and the move limit had negligible affect on the topology solution. However a few observations regarding the

Table 3.5: Variation of Move Limit for Figure 3.36

Move Limit	Eta	Iterations	Objective	Change
0.05	0.5	105	72.9910	0.01
<b>0.1</b>	<b>0.5</b>	<b>52</b>	<b>72.9875</b>	<b>0.01</b>
0.2	0.5	298	72.9587	0.01
0.3	0.5	156	72.9577	0.009
0.4	0.5	94	72.9838	0.009
0.5	0.5	79	72.9927	0.009
0.6	0.5	77	72.9897	0.01
0.7	0.5	77	72.9897	0.01
0.8	0.5	77	72.9897	0.01

general iteration count versus move limit trend can be inferred. Obviously, overly restricting the design variable updates severely hinders progress and thus the move limit should not be too small. This is the case for a move limit of 0.05. Increasing the move limit to 0.1—which is still a conservative bound, but not overly restrictive—halves the iteration count to 52. The increase in iteration count corresponding to move limits of 0.2–0.4 can be accounted for by the slightly decreased objective value. That is, imposing even less restriction on the design variables effected a lot more iterations for very little reduction of the objective value. The relatively identical iteration count and objective value once the move limit increases past 0.4 indicates that the update scheme does not naturally change any of the design variables more than about 0.4, and thus the move limit is no longer of any effect. In this case, imposing a conservative move limit of 0.1 on the design variables led to the fewest iterations for convergence within a change of 0.01, albeit the objective value was slightly higher than for less restrictive move limits.

Using a move limit of 0.1, the affect of the damping parameter as it varies from 0.1 to 0.8 is explored in Table 3.6. As evident by  $\eta$  of 0.1 and 0.2, overdamping the variable update hinders the progress of the solution. However,  $\eta$ 's of 0.3 and 0.4 show a great increase in iteration count. With these transitional damping values, the topology is slightly changing, much like it did from iteration 50 to 298 in Table 3.4, and a similar compliance decrease of about 1% transpires in the process. With a damping of 0.5, that same slightly modified topology is achieved much quicker in only 52 iterations. After this point, the update scheme is underdamped and rather unstable, hence the solution is still at a maximum change of 0.1 after 300 iterations. A few other cases were run where the move limit value was increased, but the solutions still did not converge when the damping was above 0.5. Unlike the move limit, the damping may alter the topology a little. It is not entirely clear what the standard value of  $\eta$  should be, but an  $\eta$  between 0.2 and 0.5 is deemed suitable.

Table 3.6: Variation of Damping for Figure 3.36

Move Limit	Eta	Iterations	Objective	Change
0.1	0.1	106	73.6303	0.01
0.1	0.2	72	73.7818	0.01
0.1	0.3	210	73.0599	0.01
0.1	0.4	104	72.9903	0.01
<b>0.1</b>	<b>0.5</b>	<b>52</b>	<b>72.9875</b>	<b>0.01</b>
0.1	0.6	300	73.0446	0.1
0.1	0.7	300	73.4234	0.1
0.1	0.8	300	73.2698	0.1

In conclusion, a penalty of about 3, a filter radius of about 1.5, a change of 0.01, a damping of 0.5, and a move limit of 0.1 are all generally considered to be good values and are the values used for the first attempt of any run. A range of volume fractions will however be considered, because each different volume fraction represents a physically different problem.

## IV. Results

This chapter explores wing structure layouts on a conceptual level for several wing shapes under various loading conditions. First, the aerodynamic results from Tornado of the birdwing geometry (Figure 3.4–Figure 3.7) at the four points in the perching trajectory are presented. Then, the compliance minimization runs for each birdwing layout are compared. Finally, a brief grid independence check is provided.

### 4.1 Aerodynamic Results

The birdwing of Figures 3.4–3.7 with zero camber (i.e flat plate) maneuvering through the perching trajectory of Figure 3.21 is considered here. The wing is in the back swept position at Point 1, the dive position at Point 2, the zero sweep position at Point 3, and the forward swept position at Point 4.

The Tornado calculations of the birdwing at the four points along the perching trajectory are summarized in Table 4.1. The velocity and angle of attack have been included for reference.

Table 4.1: Aerodynamic Data for Birdwing Along Perching Trajectory

		Point 1	Point 2	Point 3	Point 4
<i>Vel.</i>	[m/s]	10	10.41	10.11	2.19
<i>AOA</i>	[°]	6	3.75	10	50
<i>Drag</i>	[N]	0.0176	0.0033	0.0796	0.0456
<i>Side</i>	[N]	0.0061	0.0075	-0.0007	-0.0087
<i>Lift</i>	[N]	0.459	0.112	1.241	0.196
<i>F<sub>x</sub></i>	[N]	-0.0304	-0.0040	-0.1371	-0.1208
<i>F<sub>y</sub></i>	[N]	0.00610	0.00749	-0.00066	-0.00871
<i>F<sub>z</sub></i>	[N]	0.458	0.112	1.236	0.161
<i>C<sub>L</sub></i>	[—]	0.220	0.076	0.512	1.701
<i>C<sub>D</sub></i>	[—]	0.0085	0.0022	0.0328	0.3958
<i>C<sub>Y</sub></i>	[—]	0.0029	0.0051	-0.0003	-0.0757
<i>R<sub>e</sub></i>	[—]	90054	137712	91412	19987
<i>C<sub>Do</sub></i>	[—]	0.0101	0.0082	0.0101	0.0113
<i>S<sub>wet</sub></i>	[m <sup>2</sup> ]	0.0681	0.0444	0.0775	0.0785
<i>D<sub>vis</sub></i>	[N]	0.2368	0.1077	0.4410	0.0885
<i>Normal</i>	[N]	0.0248	0.0070	0.0766	0.0678
<i>Axial</i>	[N]	0.2355	0.1075	0.4343	0.0569

The lift forces at first glance seem to be inordinately small, as they are on the order of 1  $N$ . However, since the wing is a flat plate, an estimate of the lift can be made assuming a lift slope of  $2\pi$ . For the back swept configuration at Point 1, the lift coefficient is then

$$c_l = 2\pi\alpha = 2\pi(6\pi/180) = 0.6580 \quad (4.1)$$

The lift per unit span becomes

$$L' = c_l q_\infty c = 0.6580 \left[ \frac{1}{2} (1.2244 \text{ kg/m}^3) (10 \text{ m/s}) \right] (0.153 \text{ m}) = 6.1630 \text{ N} \quad (4.2)$$

If the wing were rectangular with uniform lift per unit span across the span, the total lift force for the wing would be

$$L = L' * b = (6.1630 \text{ N})(0.270 \text{ m}) = 1.664 \text{ N} \quad (4.3)$$

The lift is in fact on the order of 1  $N$  for such a small wing at a very low speed (about 23  $mph$ ). Since the wing is tapered and the lift distribution is certainly not uniform across the span, the resulting lift values are necessarily lower than predicted by the simple preceding equations. As expected, the highest lift occurs at Point 3 as the birdwing begins its ascent, and the lift at Point 2 is the lowest, about a tenth of the value at Point 3.

The drag calculated by Tornado is solely the induced drag, and therefore is related to the square of the lift coefficient. The higher drag values occur in the ascending portion of the trajectory, when the lift coefficient is highest. The fact that the drag at Point 4 is lower than at Point 3 is due to the very low velocity of 2.19  $m/s$ . In retrospect, selecting a point along the trajectory between Points 3 and 4 where the velocity is only somewhat reduced and the angle of attack is relatively high would have produced a higher drag load. The drag at Point 2 is smaller than the drag at Point 3 by a factor of 24, due to the reduced wing area and the smaller angle of attack.



The distributions of the pressure coefficient differences between the upper and lower surfaces of the wing are shown in Figure 4.1. The pressure coefficient differences are much greater for the forward swept configuration at the end of the maneuver, where the angle of attack is very high. Contrariwise, the pressure coefficient differences are very low for the birdwing as it dives, because it is slicing through the air at a low angle of attack. In each of the four plots, the pressure coefficient difference is about four times that of the general distribution (the yellow areas) over the wing.

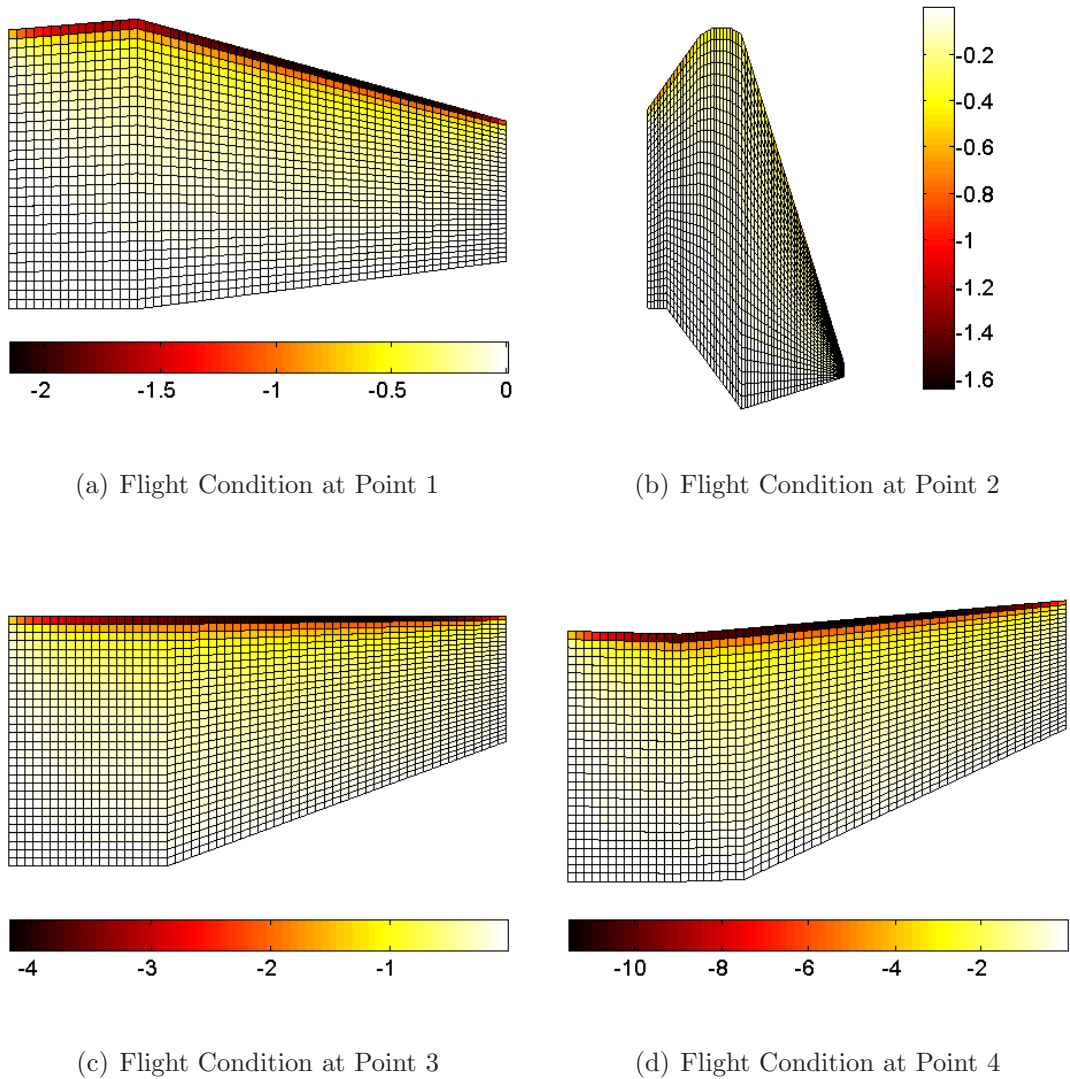


Figure 4.1:  $\Delta C_p$  for Birdwing Geometries Along Perching Trajectory

The distribution of the force components and magnitudes is shown for each configuration of the birdwing in Figures 4.2–4.5. The forces shown include the addition of the viscous drag estimates to the body forces. The viscous drag is split into a normal and axial component due to the angle of attack. The positive x-axis extends towards the tip of the wing in the spanwise direction, and the y-axis extends towards the leading edge in the chordwise direction.

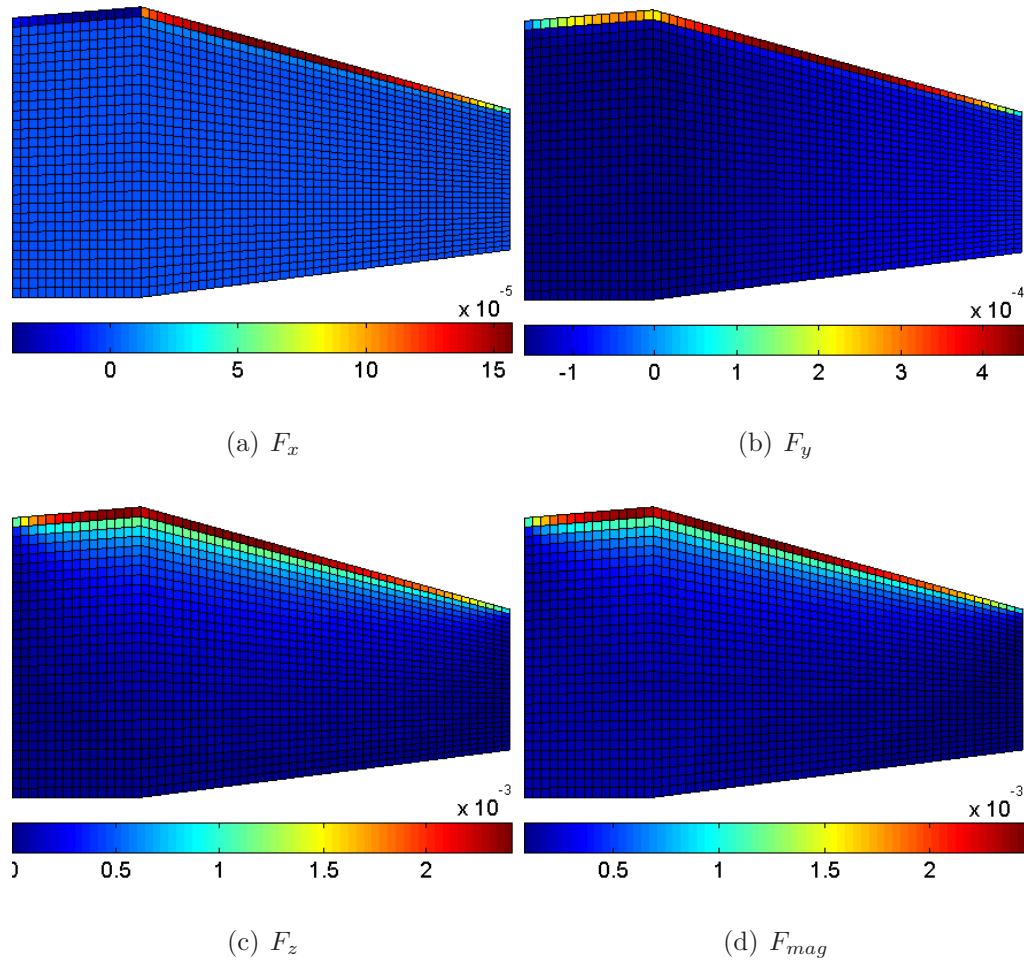


Figure 4.2: Force [ $N$ ] for Birdwing in Swept Back Configuration at Point 1

At each point in the maneuver, the side forces ( $F_x$ ) are relatively negligible and not expected to influence the structural layout much. The axial forces stimulate interest because, at the leading edge, the forces act in the positive y-direction, whereas everywhere else on the wing, the axial forces act in the negative y-direction (as in-

tuition suggests). The fact that the leading edge is seemingly being pulled forward is due to the domination of the axial component of the lift over the axial component of the drag. The lift calculated by Tornado at each panel over the surface of the whole wing dominated the drag in the axial direction; however, the added viscous drag reversed the net force over the all of the wing except the leading edge. Hence the structure is being stretched in both axial directions.

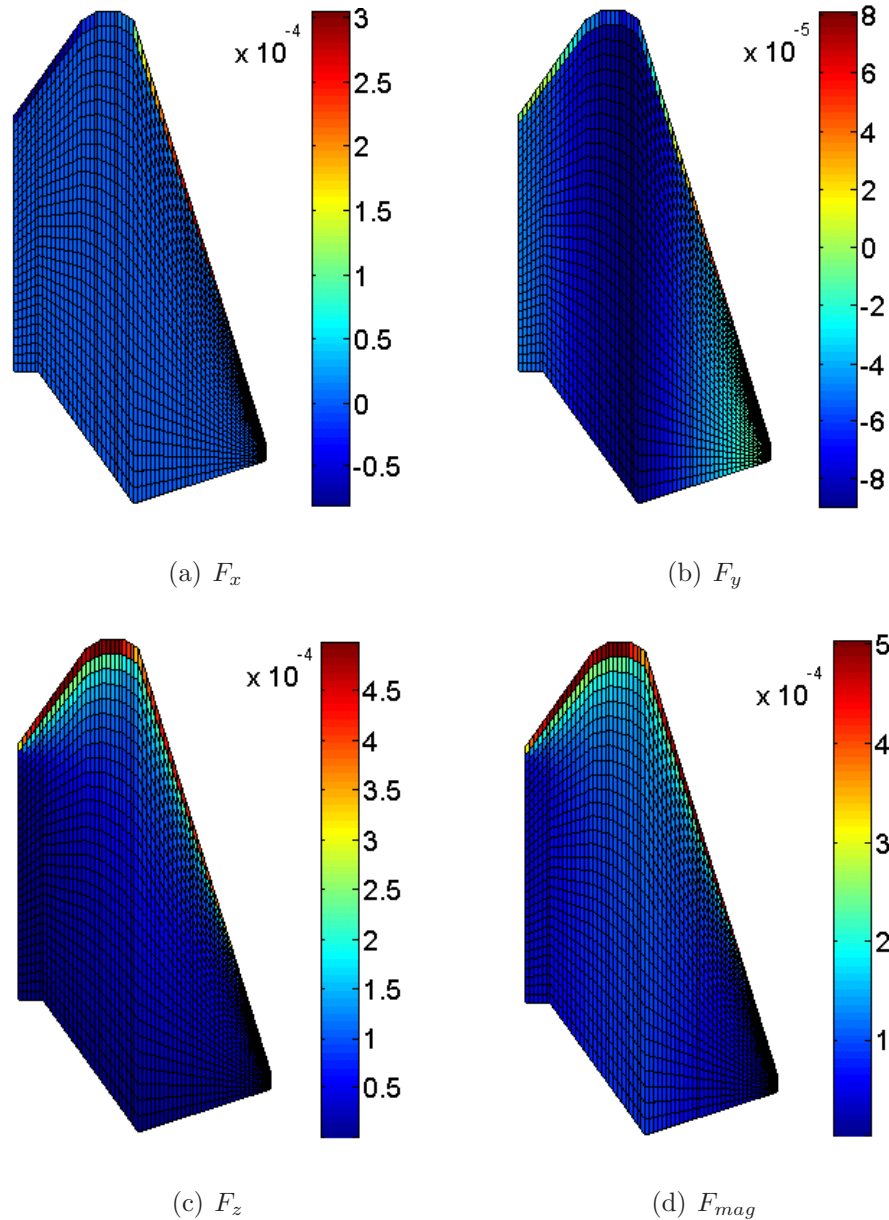


Figure 4.3: Force [N] for Birdwing in Dive Configuration at Point 2

The loads for the dive configuration are on the order of  $10^{-4}$ , whereas the loads for the other cases are on the order of  $10^{-3}$ , thus the dive configuration is probably not a good design point since ultimately the loads at each configuration will have to be withstood by a single structure.

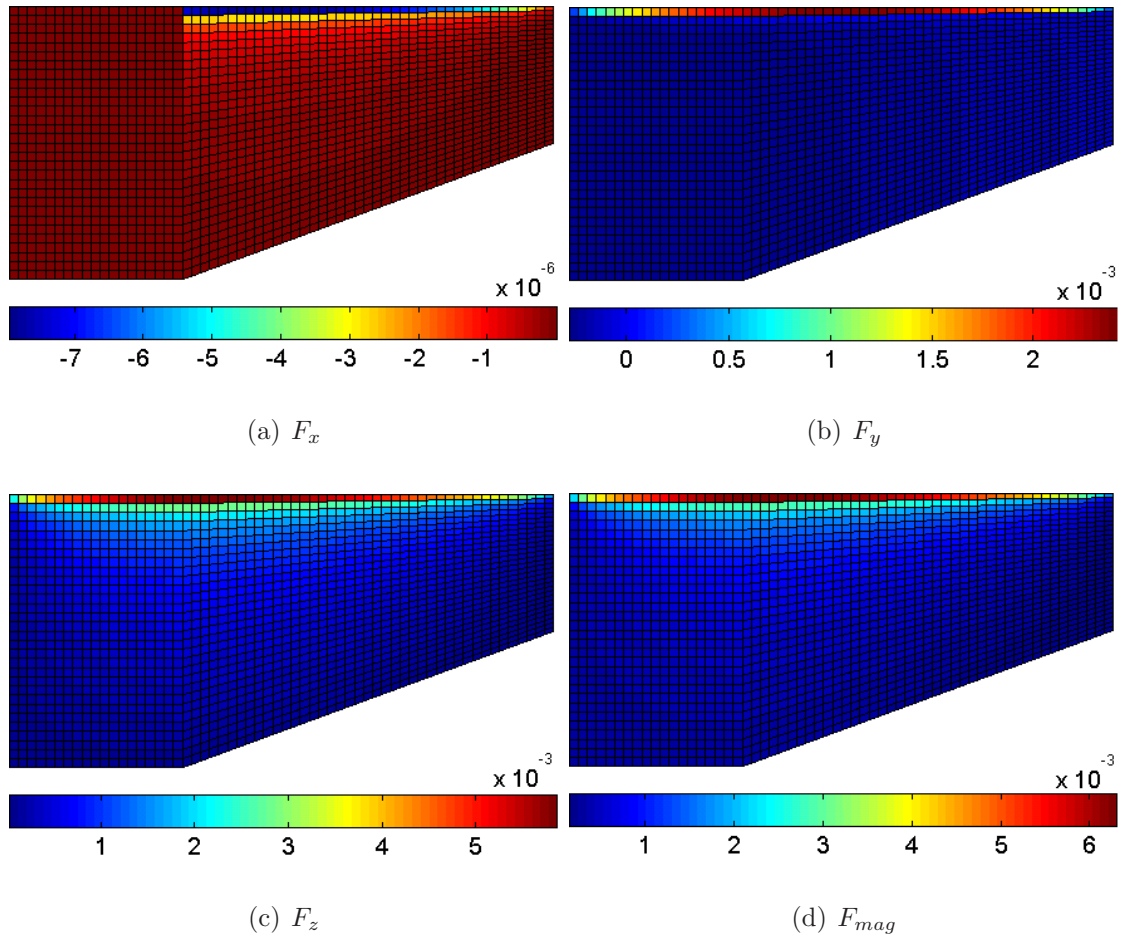


Figure 4.4: Force [N] for Birdwing in Zero Sweep Configuration at Point 3

Clearly the loads on each of the configurations is leading edge dominated. Thus a good baseline comparison for resulting structures could be constructed by simply distributing a load over the leading edge of a similar geometry. Also, resultant magnitude contours of each point in the maneuver—except the last—match the contours of the normal component  $F_z$ , indicating that the bending loads are the dominant loads. The flaring of the wings at the end of the maneuver should be bending dominated as

well, but Point 4 that was selected occurs when only a fraction of the peak velocity remains. Hence the corresponding drag loads are low, and the bending dominance is not captured. In all cases (except perhaps in the case of the dive), the trailing edge of the tip of the wing experiences essentially no load and suggests that corresponding structures may clip the lower tip of the wing.

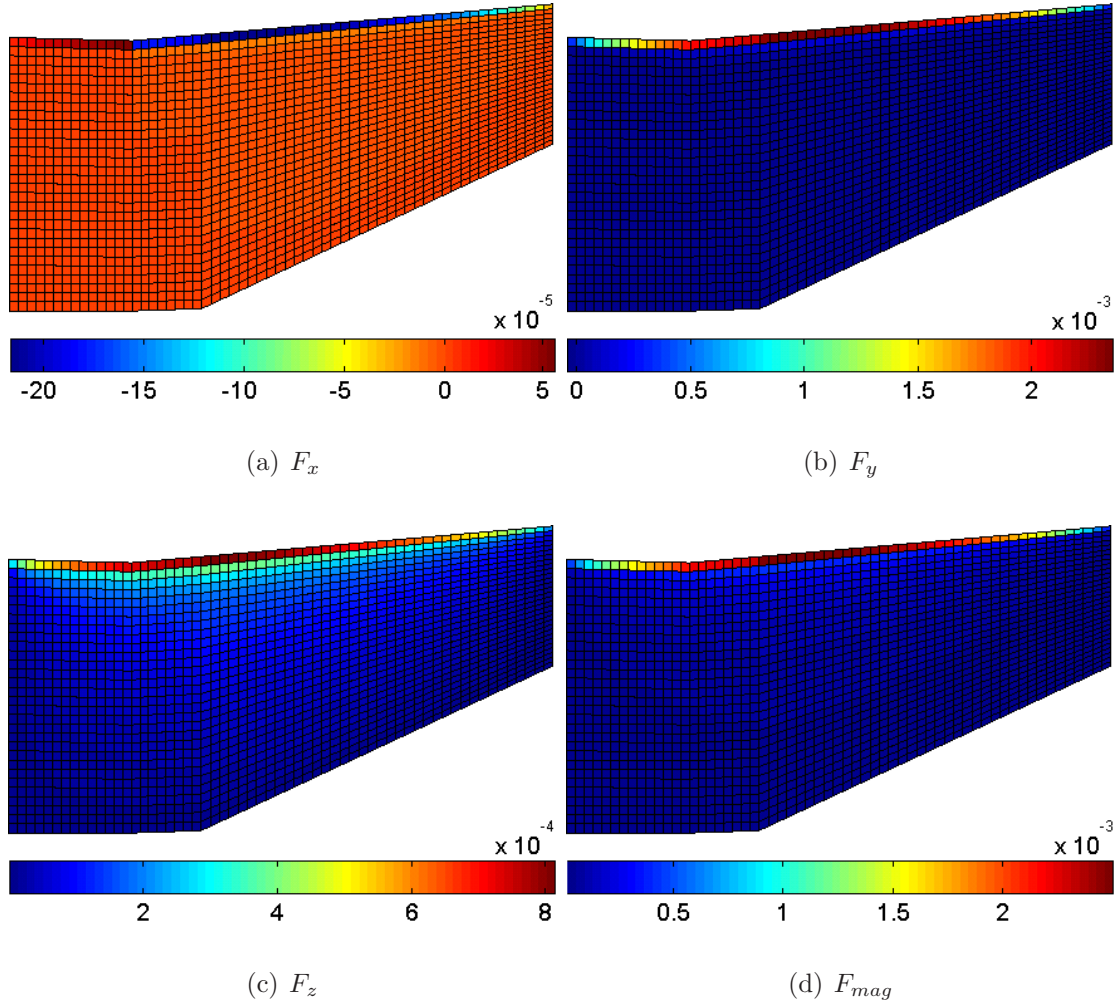


Figure 4.5: Force [N] for Birdwing in Forward Swept Configuration at Point 4

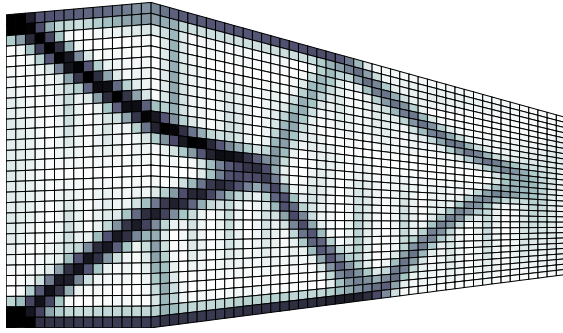
Though the magnitudes of the body forces of the birdwing in the perching trajectory is very low, the relative magnitudes are of greater importance to the optimization process. The majority of the relative magnitudes of lift and drag seem acceptable, but the most unreasonably captured aerodynamic extreme is the drag at the end of

the maneuver which should be a bending-dominated load. The beginning of the maneuver is also dominated by bending loads, but due to the lift and not the drag. If the optimization process determines any hybrid membrane-bending structures, it is anticipated that the Point 3 flight condition would produce this effect.

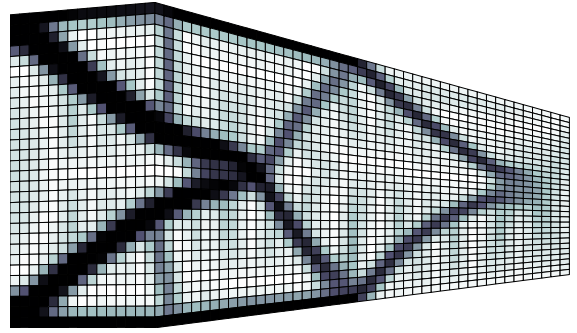
## 4.2 *Structural Layouts*

For each point along the perching maneuver, three sets of loading scenarios were performed. The first set consisted of only the membrane forces ( $F_x$  and  $F_y$ ), the second set considered the bending forces ( $F_z$ ) only, and the third set was the combined membrane-bending loading scenarios. Each set consisted of four cases with volume fractions ranging from 0.2 to 0.5. Unless convergence issues arose, each case had a penalization of 3, a filter radius of 1.5, and was run with a move limit of 0.2 and a damping of 0.5. All the runs were minimized until the solution converged within a 0.01 maximum thickness change.

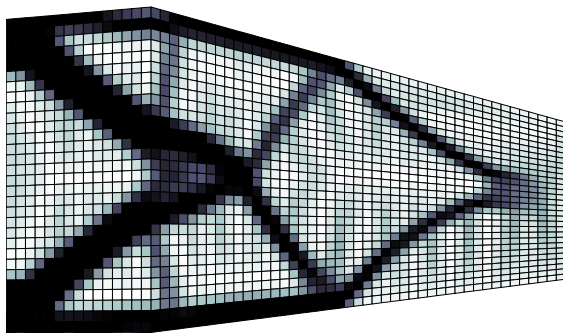
The results of the compliance minimization for the birdwing at Point 1 in the perching maneuver is shown in Figures 4.6–4.8. Considering first the purely membrane solution of Figure 4.6, the predominate structural features are the two members extending from the root of the wing in the form of a wishbone, and at the tip of the wing, a second wishbone forms. Even though the predominate membrane forces occur along the leading edge, the leading edge is not built up across the whole span like it is for the topologies of Section 4.1. This is due to the force along the remainder of the wing pulling the wing in the opposing negative chordwise direction. The “wishbone” is a natural structure to resist a spreading motion (the converse motion of squeezing a pair of pliers). It can be thought of as antipodal to two joined buttresses. Had all of the force in the chordwise direction been directed towards the trailing edge, the structures would be expected to look similar to Figure 3.34, where the leading edge is fully supported. The second main members are those along the leading and trailing edges. As evidence that the two wishbones and the leading and trailing edge members comprise the main structure, each time material is added to the wing domain,



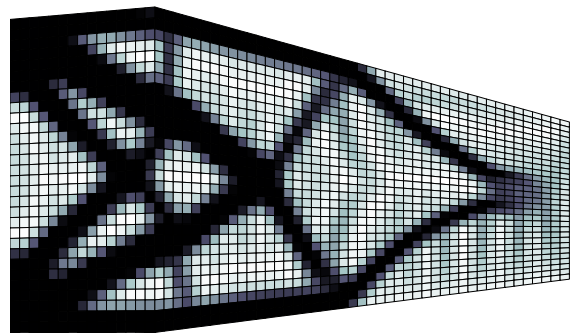
(a)  $Vol_f = 0.2$



(b)  $Vol_f = 0.3$



(c)  $Vol_f = 0.4$



(d)  $Vol_f = 0.5$

Figure 4.6: Membrane Structure for Birdwing at Point 1

these members are thickened. The wishbone structure and the intermediate members form a scissor-like structure, and consequently the voided areas are predominately quadrilateral in shape. This is in contrast to the usual triangular voids in trusses.

Another striking feature of the structure is the straight protrusions of intermediate thickness. These battens or rods point towards the center of a voided area, and towards the unsupported perimeter of the wing domain. Thus they are comparable to spokes on a wheel. This feature is a result of a distributed load covering the entire surface of the wing, and is absent in an over-simplified point load model. At a fifty percent volume fraction, the topology finally adds new members in the form of another smaller wishbone inscribed within the main wishbone.

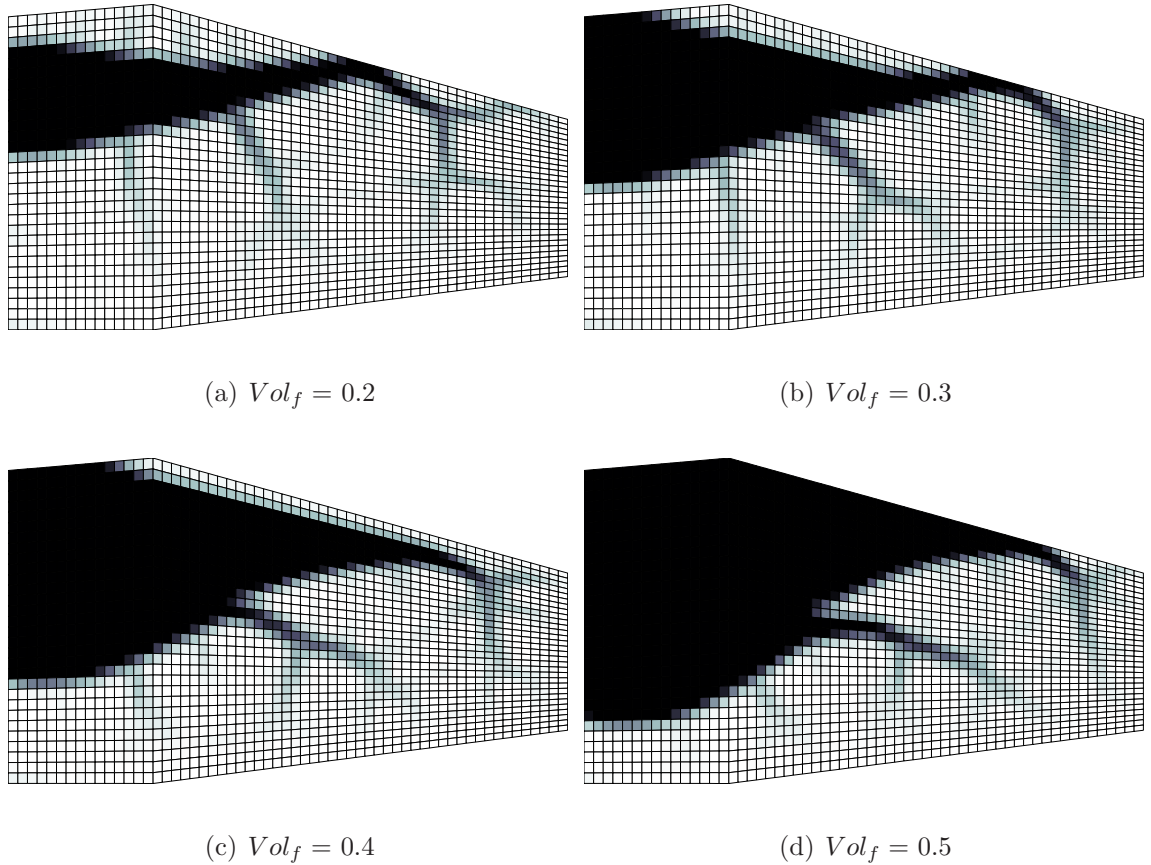


Figure 4.7: Bending Structure for Birdwing at Point 1

The bending structures of Figure 4.7 reveal that the basic form of the bending-resistant structure is not a truss, but rather a conglomeration in the form of a beam. The importance of exploring a range of volume fractions for a given wing shape and loading scenario is evident in the progression from Figure 4.7(a) to Figure 4.7(d). The impression of the fifty percent volume fraction case is that the bending structure primarily supports the leading edge, where the bending forces are highest. But Figure 4.7(a) shows this is clearly not the case. The most basic or crucial structure is a beam that is situated near the quarter chord of the wing. For the flat plate birdwing, the quarter chord is also approximately the center of pressure and the aerodynamic center of the wing, where the aerodynamic moment is zero. As the volume fraction increases, material is added along the sides of the original beam of Figure 4.7(a). The



developing shape of the beam is similar to the shape of a wishbone (i.e. like A-arms in a suspension system), or in a more crude sense, the shape of a delta.

Like the rods or battens in the membrane structure, the bending structure also develops a secondary feature captured by the intermediate thicknesses. Rather than straight rods with a regular arrangement, branches stem from the central beam structure in arbitrary directions, and then further divide into smaller networks that attempt to cover as extensive a region as possible. This branches resemble veins in the wings of insects, and as the high density of nerves in bats attest, are a very effective method of stiffening a membrane skin.

The combined membrane and bending structure for the first point in the maneuver is shown in Figure 4.8, revealing an identical structure to the membrane case. However, when the viscous drag is removed, the result is that of Figure 4.9, which

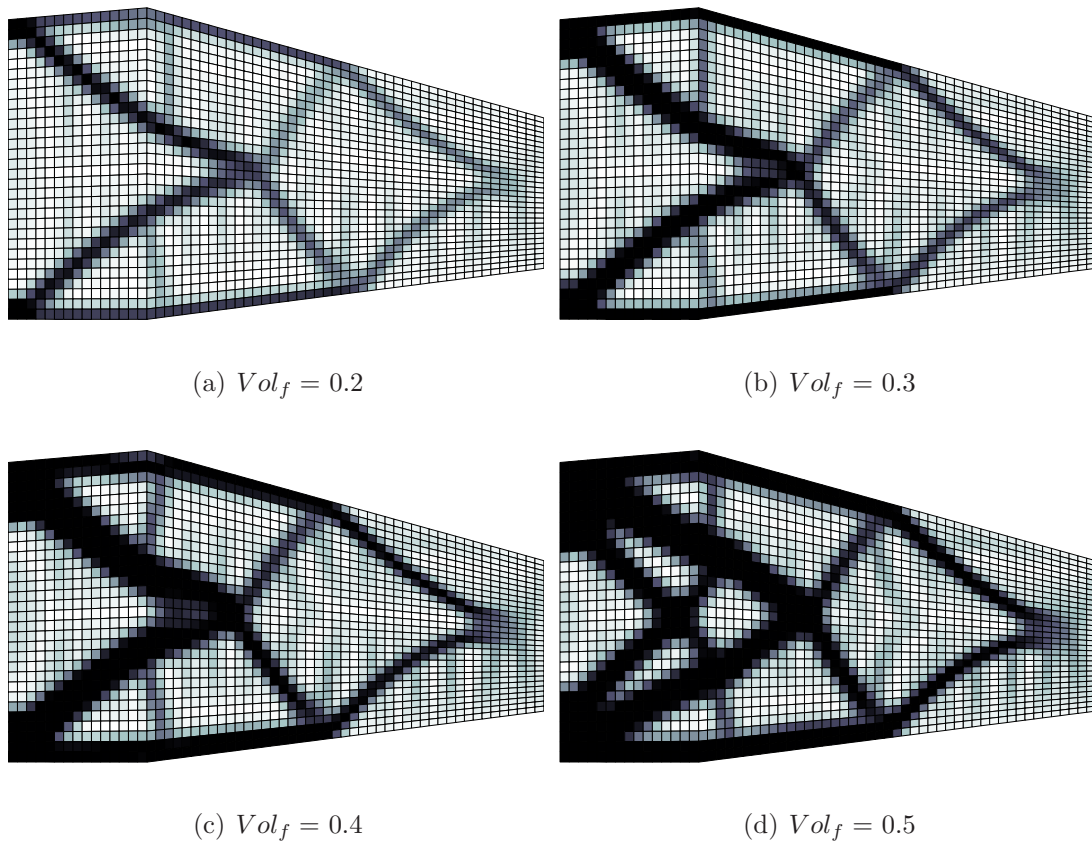


Figure 4.8: Combined Structure for Birdwing at Point 1

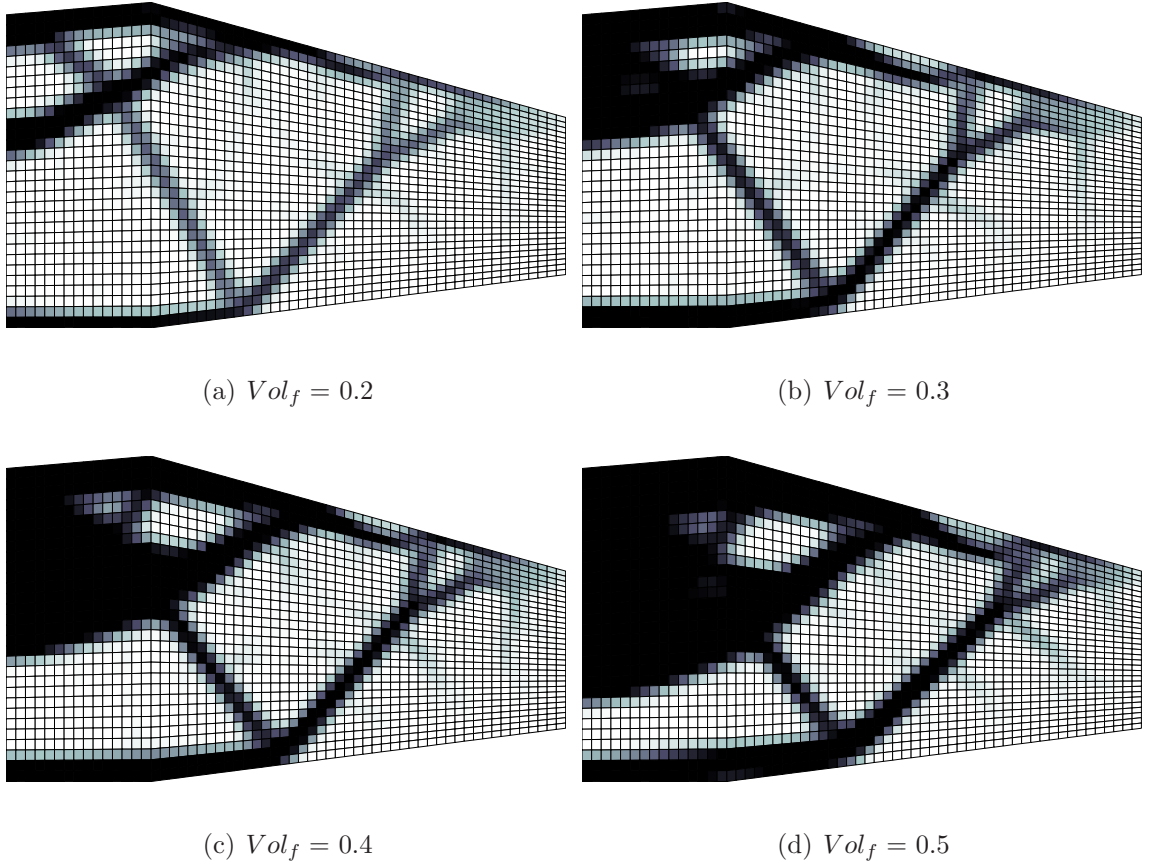
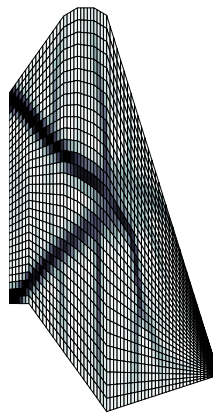


Figure 4.9: Combined Structure for Birdwing at Point 1 without Viscous Drag

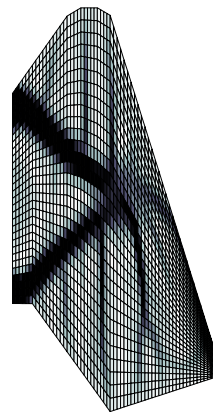
reveals a hybrid solution of the two isolated membrane and bending cases. This is not immediately evident in Figure 4.9(a), but the higher volume fraction solutions all have a delta-shaped “beam” on the front half of the wing. All of the solutions have supporting truss structure along the trailing edge of the wing that then connects to the leading edge near the tip of the wing. As material is added, the general topology remains unchanged, however the beam portion of the structure shifts in the chordwise direction, such that at fifty percent volume fraction, it is centered mid-chord. Both the scissor pattern from the membrane solution and the beam extending most of the way in the spanwise direction are attenuated, and the emerging structure better resembles a perimetric truss structure with spokes protruding inward to a hub covering the upper left corner of the wing. The vein-like stiffeners are absent, and only a few rods are present. While the hybrid solution is interesting, in actuality a completely

membrane structure supports the combined loading according to the compliance results. The larger lifting loads and smaller drag loads at the relatively small angle of attack of Point 1 would suggest a more bending-dominated solution, but the effect of the distributed viscous drag load drives the solution towards a membrane structure.

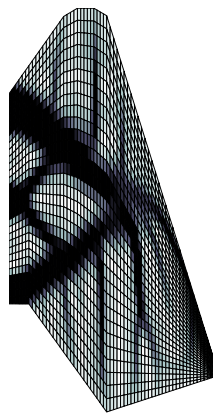
As previously mentioned, the diving portion of the perching maneuver is perhaps not a good design point when the magnitude of the loads experienced throughout the entire maneuver is considered. However, the wing shape is quite distinct from the other three configurations and thus yields unique structures worthy of investigation. The resulting structures are presented in Figures 4.10–4.12.



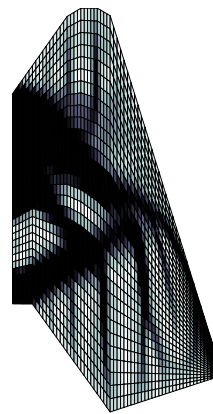
(a)  $Vol_f = 0.2$



(b)  $Vol_f = 0.3$



(c)  $Vol_f = 0.4$

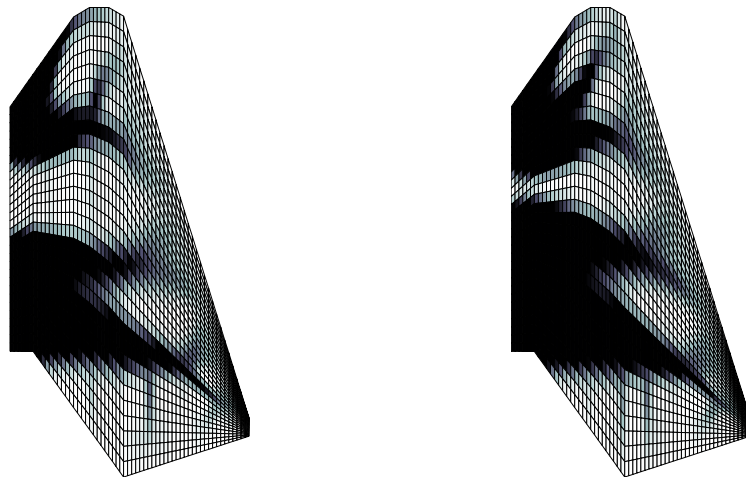


(d)  $Vol_f = 0.5$

Figure 4.10: Membrane Structure for Birdwing at Point 2

If obscured by other truss members in the back swept wing configuration, there is no overlooking the prominent wishbone of Figure 4.10. In fact, it is the only truss structure present in Figures 4.10(a) and 4.10(b). In Figure 4.10(c), a smaller wishbone facing opposite of the main wishbone is inscribed within the main wishbone again producing a quadrilateral void and a scissor-like structure. Aside from this small amount of “truss” structure, the remainder of the structure consists of very long and definitive battens, in comparison with those in the back swept wing structure. In fact, the battens are not of intermediate value, but are maximum thickness. As shown in Figure 4.3(b), the maximum axial load occurs at the leading edge near the tip of the wing, and accordingly a batten extends to the tip of the wing. The region is not just darkened because of the overly refined mesh. In light of the poor finite element mesh of the dive wing shape, an instructive structure develops.

The compliance minimization of the bending structures shown in Figure 4.11 was troublesome, and only the forty and fifty percent volume fraction runs yielded results that were not explicitly erroneous. The twenty and thirty percent volume fraction cases were never close to converging, even for a number of different damping and move limit settings. Instead, they repeatedly violated the volume constraint, and



(a)  $Vol_f = 0.4$

(b)  $Vol_f = 0.5$

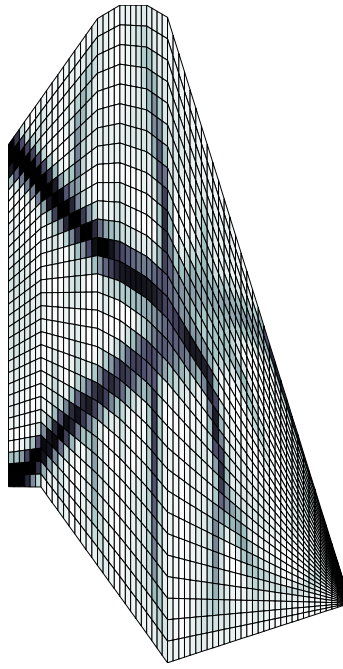
Figure 4.11: Bending Structure for Birdwing at Point 2

the maximum thickness change diverged as the iterations progressed. The maximum thickness change between iterations of the two solutions that were obtained oscillated in very high increments and eventually did drop beneath the 0.01 threshold. However, such unsteadiness in the tracking data is indicative of results that can hardly be said to have converged. In light of the poor solution, Figure 4.11 reveals two separate beam structures at the leading and trailing edge of the wing. The proportion and location of the “stiffeners” at the end of the beam extending towards the tip of the wing, when viewed as a whole, perhaps resemble feather-like protrusions.

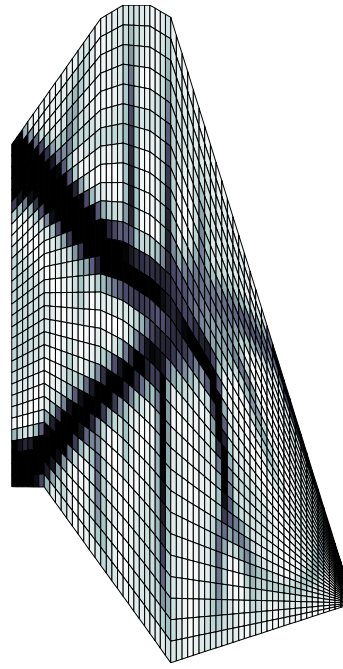
As in the combined loading case of Point 1, the distributed viscous drag again leads to combined structures identical to the isolated membrane structures, and is shown in Figure 4.12. When the viscous drag is again removed, the combined loading case for Point 2 results in Figure 4.13. In this case, Point 2 elicits a clearly bending-dominated structure. Even though the lifting load is significantly diminished, the induced drag loads are extremely low and hence show virtually no influence in the structure, save a few faint battens. Instead the structure is mainly comprised of relatively equally sized and spaced branches extending from the root chord to the tip chord, like tines on a fork. The structure bypasses the trailing edge entirely.

As the birdwing in the perching maneuver begins its ascent, it is transitioning from bending loads due to lift to bending loads due to drag. At this transitional stage, axial and normal body forces are the same order of magnitude, and, more so than at the other points, a hybrid membrane-bending solution is anticipated. The results of the compliance minimization for Point 3 are shown in Figures 4.14–4.16.

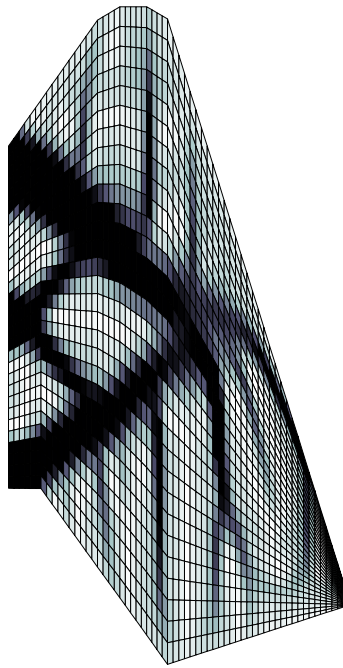
The layout and features of the membrane solution in Figure 4.14 are very similar to that of Figure 4.6, with one major distinction. The trailing edge is better supported than the leading edge. This could in part be due to the forward sweep of the trailing edge (unlike the relatively symmetrical planform of the swept back configuration). However, Figure 4.14(a) is particularly perplexing, as the leading edge is very faintly supported by a very long leg of a wishbone. One explanation may lie in the fact that



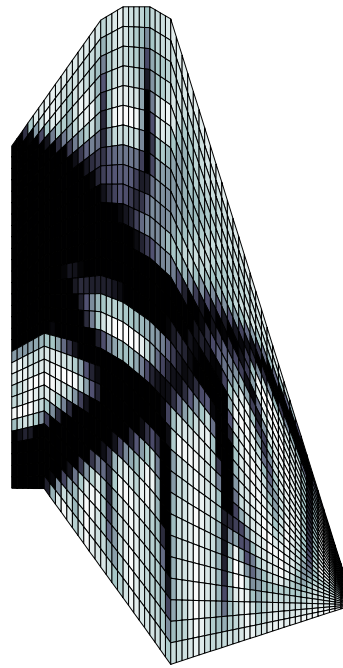
(a)  $Vol_f = 0.2$



(b)  $Vol_f = 0.3$

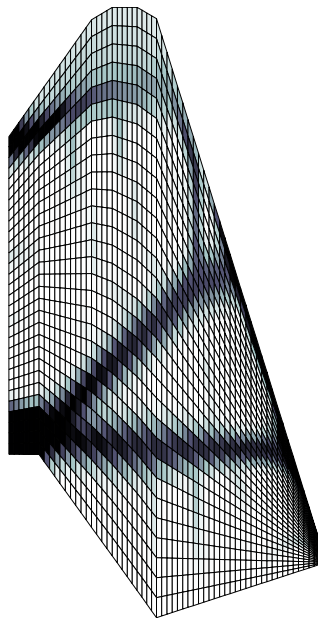


(c)  $Vol_f = 0.4$

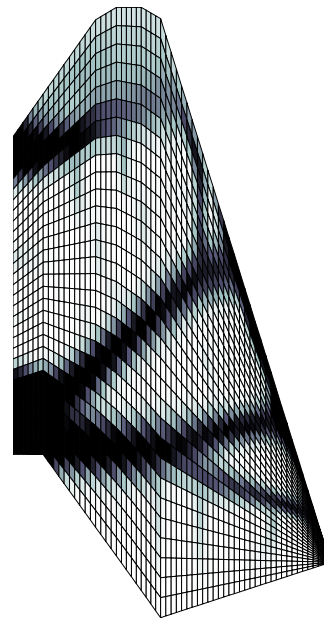


(d)  $Vol_f = 0.5$

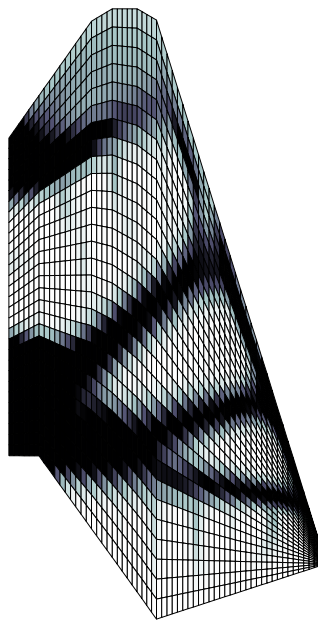
Figure 4.12: Combined Structure for Birdwing at Point 2



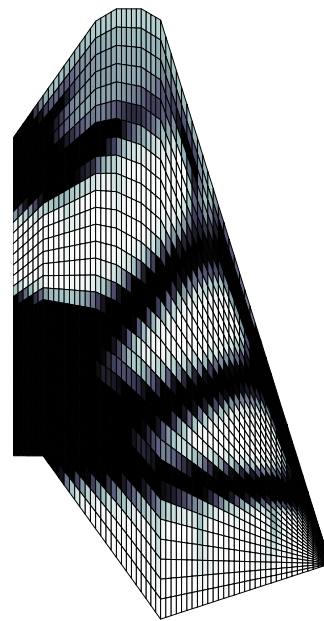
(a)  $Vol_f = 0.2$



(b)  $Vol_f = 0.3$



(c)  $Vol_f = 0.4$



(d)  $Vol_f = 0.5$

Figure 4.13: Combined Structure for Birdwing at Point 2 without Viscous Drag

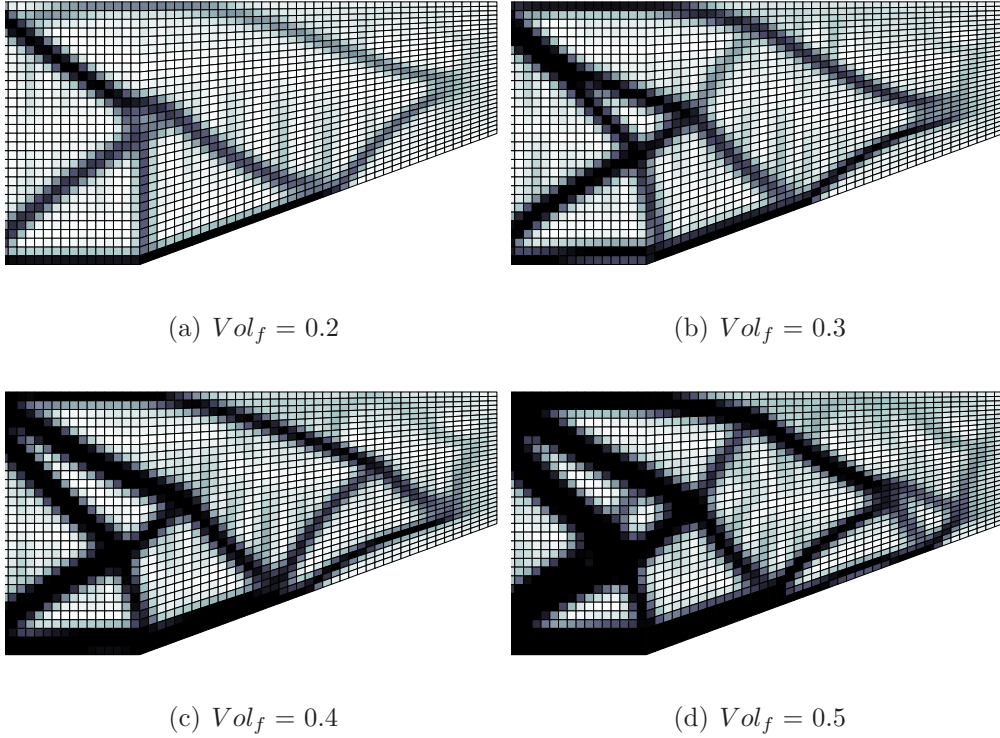


Figure 4.14: Membrane Structure for Birdwing at Point 3

there is less positive axial force than that of the back swept configuration, and hence the trailing half of the wing is being pushed relatively more in the negative axial direction compared to how much the leading edge is being pushed in the positive axial direction. At any rate, the disparity between the support along the leading and trailing edges vanishes with higher volume fractions.

The bending structures are very similar to those of the back swept configuration at Point 1. A single beam extends from the root chord most of the way to the tip of the wing. Also, the same type of stiffeners appear. However, the beam in this instance is clearly building up the leading edge, and is not centered near the quarter chord.

Figure 4.16 does not reveal a hybridization of membrane and bending elements as anticipated. The solution again favors a membrane structure, both with and without the presence of the viscous drag. However, in the absence of viscous drag (Figure 4.17) a thicker leading edge emerges, and mass starts to cluster around the upper



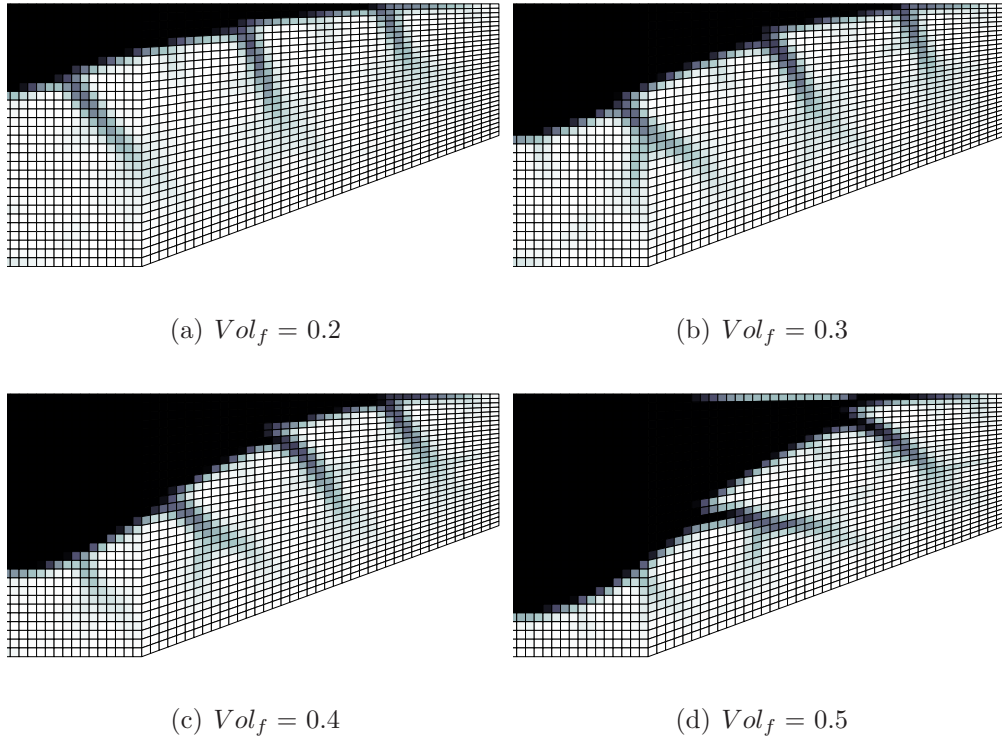
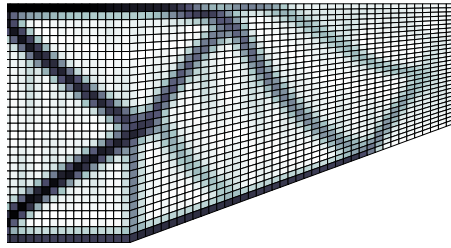


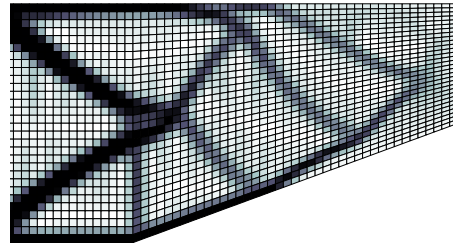
Figure 4.15: Bending Structure for Birdwing at Point 3

left corner of the wing, as it does slightly in Figure 4.16(d). When this happens, the cluster assumes the role of a hub, where separate chains of truss members revolve around the hub and are connected to the hub by “spokes”. The structure essentially forms a polar grid, and consequently little material is left to form battens. The fact that both the induced drag and viscous drag are at peak values accounts for greater membrane influence and dominance over bending.

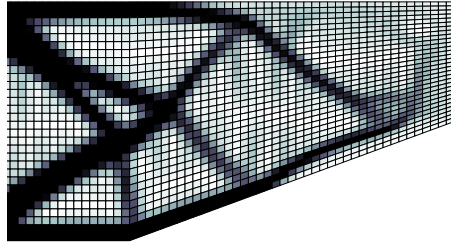
The forward swept flaring configuration of the wing is shown in Figures 4.18–4.20 for the final point in the perching maneuver. As mentioned in the previous section, the intent of Point 4 was to capture the high braking drag loads that ultimately deflect the wings out-of-plane, but the point was selected when only twenty percent of the velocity remained. Therefore the drag that should have produced a highly bending-dominated structure was lacking, and instead, the selected Point 4 is actually the only point where the axial forces (in this case, due to lift) exceed the normal body forces.



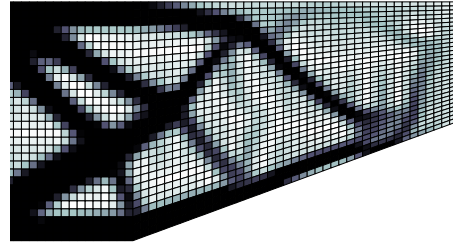
(a)  $Vol_f = 0.2$



(b)  $Vol_f = 0.3$

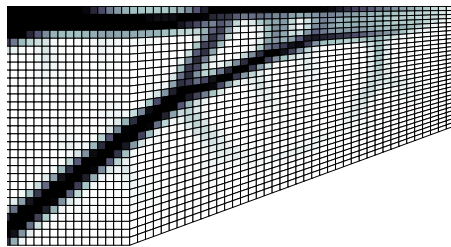


(c)  $Vol_f = 0.4$

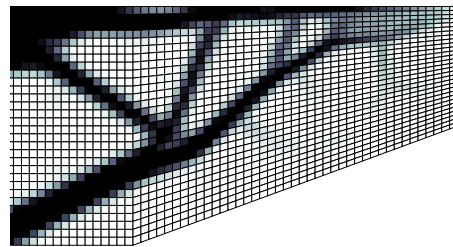


(d)  $Vol_f = 0.5$

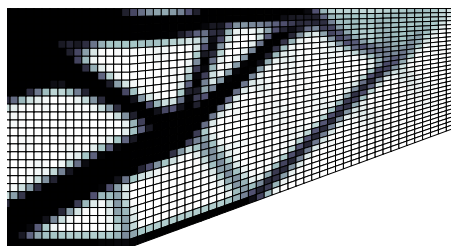
Figure 4.16: Combined Structure for Birdwing at Point 3



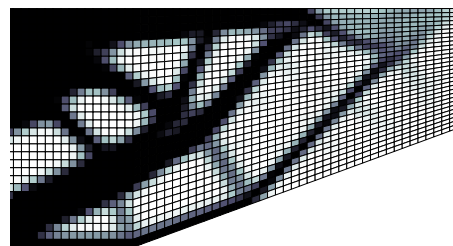
(a)  $Vol_f = 0.2$



(b)  $Vol_f = 0.3$



(c)  $Vol_f = 0.4$



(d)  $Vol_f = 0.5$

Figure 4.17: Combined Structure for Birdwing at Point 3 without Viscous Drag

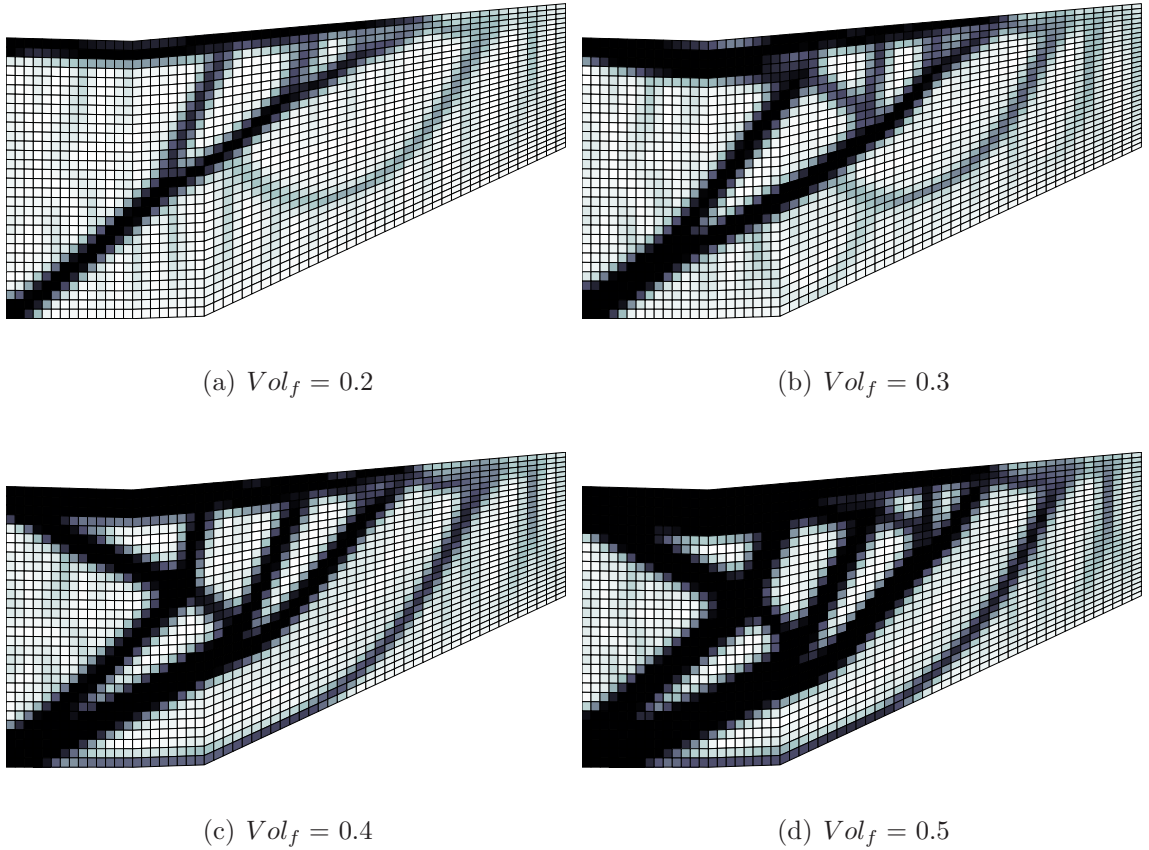


Figure 4.18: Membrane Structure for Birdwing at Point 4

In the membrane solution at Point 4, the fundamental wishbone structure, while not completely absent, does not best describe the layout. Rather, three main fingers (for volume fractions of 0.4 and 0.5) and one thinner finger emanating from the trailing edge support the leading edge. The three fingers have the appearance of three arms branching out on a candelabra. They could alternatively be likened to the three fingers found in bat wings. Many battens are again present. Additionally, some intermediate thicknesses are hanged like tinsel, covering the bottom half of the wing, in the twenty and thirty percent volume fractions.

The bending topologies do not present any new features and are very similar to those occurring at Point 1 and Point 3.

In the absence of significant drag loading, the combined structure is completely membrane-dominated. The combined structure is again virtually indistinguishable

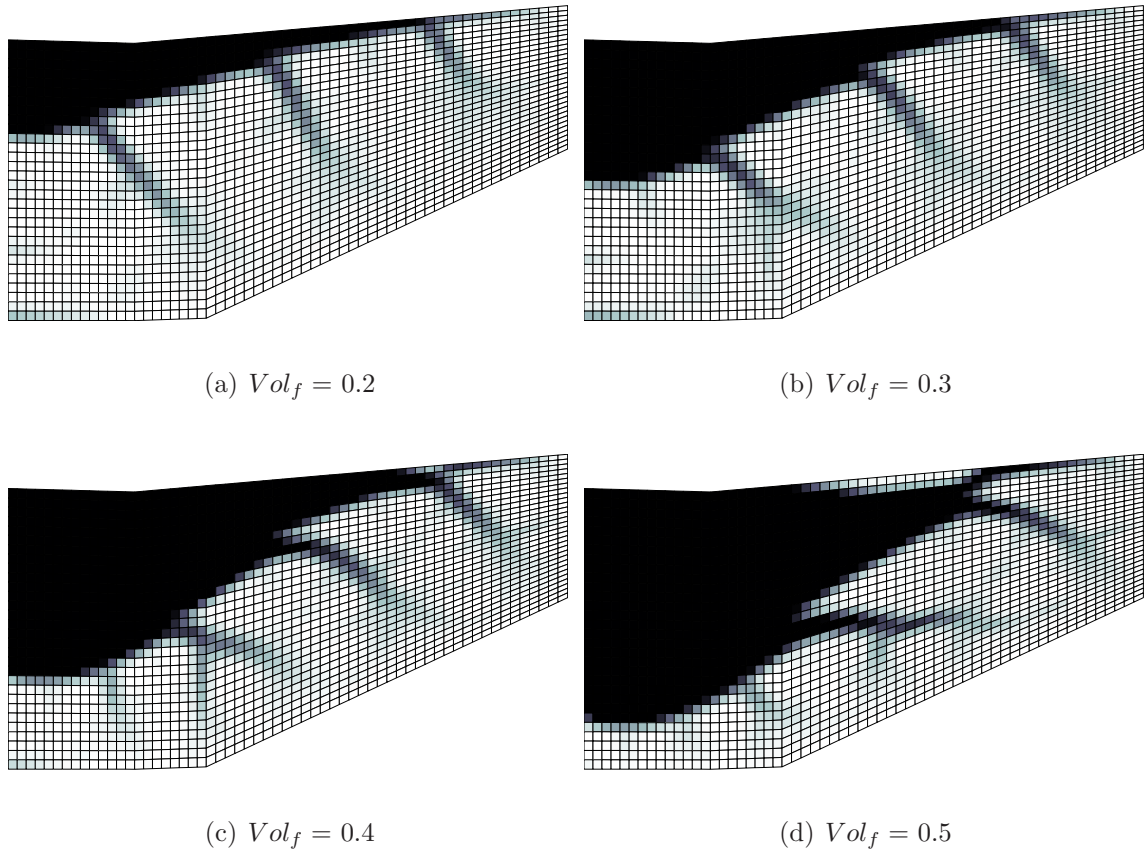


Figure 4.19: Bending Structure for Birdwing at Point 4

from the membrane case, and thus the combined structure with no viscous drag is shown in Figure 4.20. In the absence of viscous drag, a slightly different membrane structure emerges. Though there is no “hub” resulting from a beam, the members are again arranged in the fashion of a web. This is fundamentally distinct from a typical truss structure which uses the inherent stiffness in triangulation. The solution is very clean and virtually no battens or stiffeners are present.

In this section, both the structural layout due to membrane loads and bending loads were studied independently, such that in the actual combined loading, one could identify whether the structure assumed an arrangement in the membrane or bending fashion. In Chapter 3, Figures 3.18–3.20 demonstrated the extent to which bending loads can produce significantly more stress in a structure than membrane loads of the same magnitude do. Yet, the combined loading optimization continually yielded

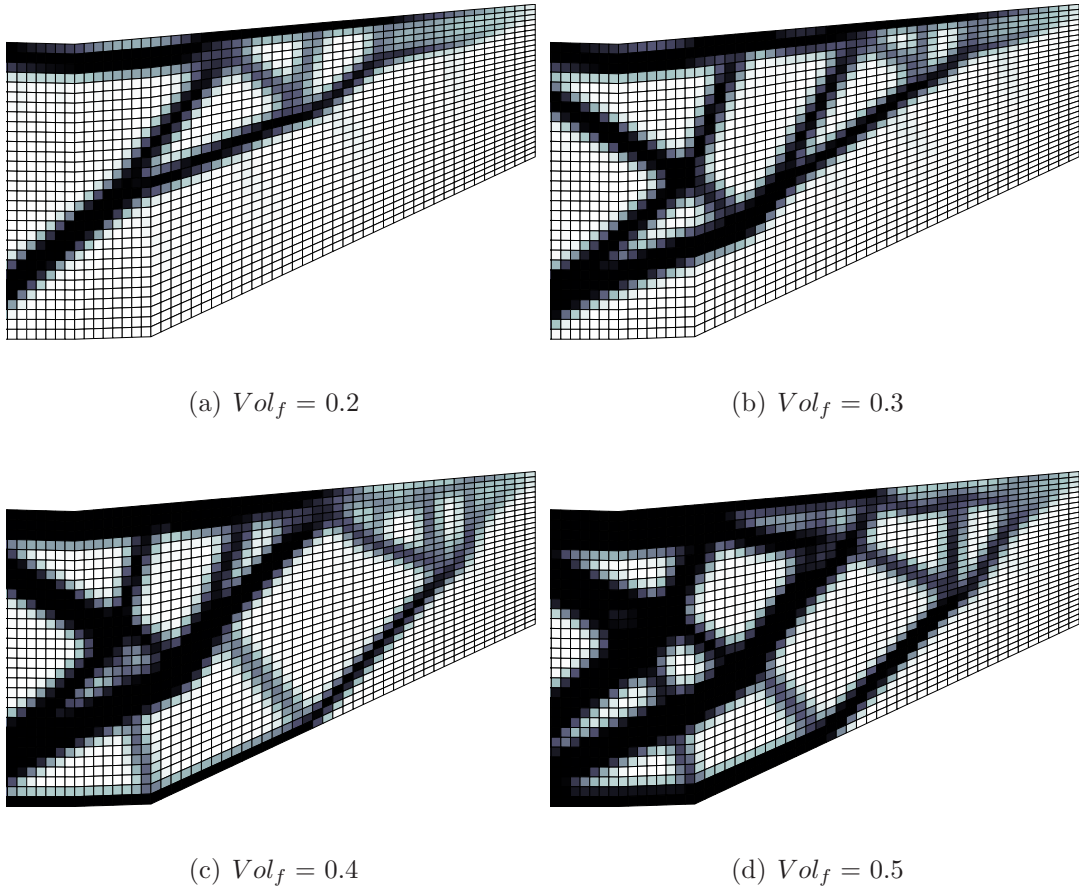


Figure 4.20: Combined Structure for Birdwing at Point 4 without Viscous Drag

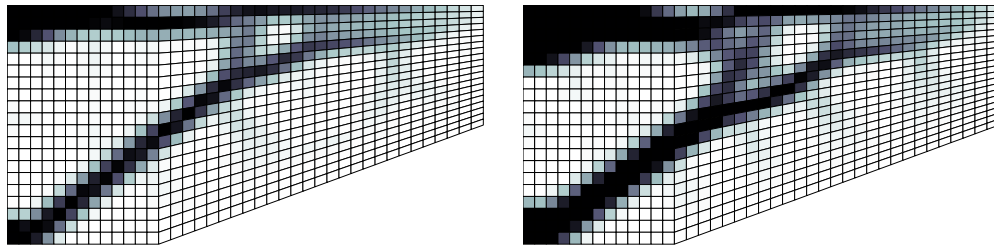
membrane layouts. When the viscous drag was removed, the effect was a structure that was a hybrid between the truss-like membrane structures and the beam-like bending structures. Thus, the viscous drag forces drove the solutions. Of course, the normal loads are still supported, albeit they were greater in magnitude than the axial loads. So the implication is that the membrane structure with discrete elements is also adequate to support out-of-plane loads. Thus the discrete members function like spars in resisting bending, but, rather than being arranged parallel to the leading edge and evenly spaced, they are aligned in a manner that also resists in-plane stretching. The implication of using a structure with discrete members for wing design is that the mechanism designed to achieve a planform shape change should be suitable to support out-of-plane loads provided that the members have bending stiffness. In

this case, the actual layout of the members would be determined by a mechanism design and not by compliance minimization. However, if compliance minimization is used to suggest structural configurations, the general layout of Points 1 and 3 (which are very similar to each other) would probably be the best design point, since the MAV would spend the majority of its flight time in those configurations, and because these points experience the highest loads. Whether a shape-changing mechanism or a compliance minimization is used to design wing structure, the topologies explored here demonstrate that the leading edge must be predominately supported. Having a larger member extend across the leading edge may integrate well with a mechanism design where the leading edge is simply swept during a planform area change.

### ***4.3 Grid Independence***

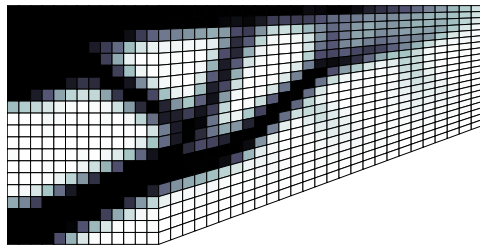
In Section 1.2, it was mentioned that the discretization scheme for a distributed parameter system will affect the solution, and thus grid quality may be a concern. To briefly address this concern, a quick grid-independence study is undertaken here. In a first attempt, the mesh of the  $0^\circ$  sweep wing configuration was refined by a factor of 1.5 and then 1.25, but both refinements failed in Tornado. It may be that the corresponding cells were physically too small for successful computation by the vortex lattice method. However, a coarse  $40 \times 20$  mesh was created and the compliance optimization was performed for the combined loading case without viscous drag, shown in Figure 4.17 for the  $60 \times 30$  mesh. The results are shown in Figure 4.21.

For the twenty percent volume fraction, little to no difference is observed. The thirty percent volume fraction case produced the same basic layout, but the coarse mesh lacked one of the “spoke-like” members present in the finer mesh. In the forty percent volume fraction case, the entire outer radius of truss members is missing, and is not present until the fifty percent volume fraction case. Though the overall layouts are of the same essence, a finer mesh allows material to divide into smaller members, which is a similar effect as simply increasing the volume fraction. Convergence data is also provided in Figure 4.22.

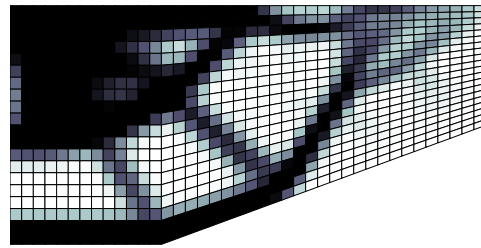


(a)  $Vol_f = 0.2$

(b)  $Vol_f = 0.3$



(c)  $Vol_f = 0.4$



(d)  $Vol_f = 0.5$

Figure 4.21: Combined Structure for Birdwing at Point 3 with Coarse Mesh

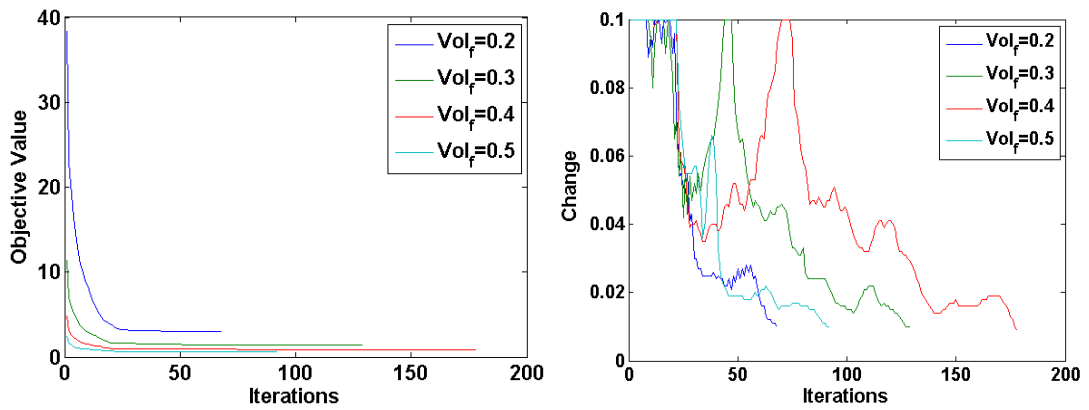


Figure 4.22: Convergence Data for Figure 4.21

## V. Conclusions

### 5.1 Summary

In 2003, DARPA began the MAS program with the aim of developing morphing aircraft structures that can substantially change shape to adapt to changing mission environments. Morphing wings with planform capabilities of telescoping, chord extension, and variable sweep will enable a multi-role platform far superior to both that of conventional fixed-wing aircraft, and even to aircraft with the capacity to modify their airfoil shape. Achieving radical shape change will require design that integrates innovative technology, including advanced flexible skin materials, efficient actuators and power systems, and mechanisms and structures with the facility to execute shape change.

Because morphing wing design is a multidisciplinary challenge, new design methods and processes must be synthesized that consider the wholistic problem. The developing field of topology optimization promises to be an effective tool that can simultaneously optimize the arrangement and connectivity of the load-bearing structure, target shape-matching mechanism, and flexible skin elements, for any number of design objectives. To this end, the literature reviewed in Chapter 2 highlighted a number of efforts undertaken that decompose a two-dimensional wing into a layer of structure, a layer of mechanism, and a layer representative of flexible skin. This model successfully demonstrated the ability to design a structure capable of achieving multiple wing configurations with an efficient distribution of actuators. Other studies developed design processes and topologies for flexible skin. In one of these efforts, a wing skin was designed by discretizing a wing planform into a number of patches with its own unique material requirements, and then engineering a material for each patch with a distinct microstructure. However, lacking throughout the different design processes reviewed was the modeling of realistic, three-dimensional loading conditions an air vehicle might experience during a morphing maneuver.

Therefore, two objectives were established to advance the previous efforts. The first objective was to develop a six degree-of-freedom finite element model comprised



of continuous elements. This model would then be capable of supporting out-of-plane loading and could be used to represent the flexible skin of the wing. However, a six degree-of-freedom model with continuous elements would also be able to represent the entire structure of the wing. It was important to implement the model in MATLAB to be able to easily integrate it with optimization routines. The second objective was to consider realistic aerodynamic loading that MAV wings might experience through a changing mission environment. The perching maneuver common to birds was selected as an interesting miniature of a multi-role mission in which a bird undergoes several planform transformations.

By constructing a compliance minimization optimization objective (equivalent to a stiffness maximization objective), conceptual wing structure layouts of a MAV at various points throughout a perching maneuver were investigated. Both the structure corresponding to the isolated cases of membrane and bending loading, as well as the combined loading, were optimized. The results revealed some distinctive features of wing structure when only membrane loads or when only bending loads were considered. The structures formed by membrane loads were comprised of many discrete truss-like members that generally supported the leading edge of the wing. Rather than predominantly relying on triangulation for inherent strength, the members were typically arranged in cobweb formations with quadrilateral voids. Straight rods or battens protruded from the truss members pointing towards the center of a voided region or towards the outskirts of the wing. Contrary to the membrane structures, the bending loads elicited a principal beam-like structure situated anywhere between the leading edge and the quarter-chord, and extending from the root chord to the tip of the wing. With additional material, the shape of the beam evolved into a wishbone or delta conglomeration. Contrary to the battens that formed in the membrane case, chutes with endings that branched in arbitrary directions emanated from the central beam. These chutes act like stiffeners similar to the neural structures found in the flat wings of insects. The full three-dimensional loading typically favored the membrane structural formation over that of the bending. However, in the absence of the viscous

drag distributed over the surface of the wing, a structure formed that is a hybrid of the typical membrane and bending structures. Thus the truss-like structure corresponding to the combined loading is thought to be driven by the distributed viscous forces. However, the discrete members must also behave like spars that can resist the bending loads, which ultimately contribute the most to the total magnitude of the forces on the wing.

## **5.2 Recommendations**

Though the resulting structures studied in Chapter 4 revealed distinctive features that parallel many formations with inherent strength found in nature, the methodology developed in Chapter 3 can certainly undergo improvements. First of all, the discretization scheme of the geometry for the finite element model was chosen to be structured meshing. Though the structured mesh was much easier to implement and also allowed a common mesh to be used for both the finite element method and the aerodynamic vortex lattice method, an unstructured mesh is more versatile in accurately modeling complex geometries. The inability of the structured mesh to model highly non-rectangular geometries was observed with the dive configuration, and merits greater consideration of using an unstructured mesh. However, if an unstructured mesh is to be employed, the advantage of easily discretizing a geometry with triangles is lost, since the element geometries derived were quadrilaterals.

The finite element itself could, of course, be selected differently. For example, rather than using Kirchhoff plate theory, Mindlin-Reissner plate theory could be used to couple the in-plane and out-of-plane effects. This would result in a five degree-of-freedom element that would avoid the need for superimposing two separate membrane and bending elements together. Such an element should also avoid the small displacement assumption of the combined membrane-bending element. Also, higher-order, nonlinear shape functions could be used, rather than the simple bilinear shape functions used here. Higher-order shape functions should increase the accuracy of the finite element model, particularly for cambered geometry.

The MATLAB implementations were very simple and the routines ran very efficiently. Both the membrane and bending models were validated with classical solutions, and their accuracy was verified. The superimposed membrane and bending element, while not validated with an analytical solution, did match the individual solutions of the membrane and bending models. The three dimensional six degree-of-freedom model, however, still lacks validation and successful integration with the SIMP model of the compliance optimization. If a full three dimensional analysis is not easily obtainable for comparison and validation of the model, the three dimensional FE model can be validated with a commercial finite element code, such as NASTRAN.

The aerodynamic force estimations can afford much improvement, though the extent to which improvement would influence the optimized thickness distributions may not be substantial. Effort should be spent to ensure that the outputted forces from Tornado do in fact act through the centroid of each panel. Post-stall aerodynamic models can also be employed to improve the aerodynamic calculations at the end of the maneuver. Also, a more ideal selection of Point 4 in the perching maneuver would capture the high angle of attack, and high angle-of-attack and pitch rates, while the aircraft still has most of its velocity. This should in turn capture the high bending loads that the drag exerts on the aircraft as it begins to slow down into a vertical landing.

In general, the Optimality Criteria method performed very efficiently, usually completing a minimization in less than five minutes with a couple seconds at most between iterations. The Method of Moving Asymptotes (MMA) is often employed for large scale optimization problems and may reduce computation time. If larger problems are undertaken, such as would be produced from mesh refinement, the MMA solver may be worthy of investigation. However, for the problem sizes explored in Chapter 4, the OC method executed very well.

### **5.3 *Future Work***

Though minor improvements can be made to the overall method used to determine wing structure layouts, the study of wing structures was of a purely conceptual nature. Thus, further elaborating the compliance minimization model will not lead to detailed design in the future. Rather, the work reviewed in Chapter 2 can be revisited with the finite element model, such that full three-dimensional loading can be applied to the simultaneous, multidisciplinary wing design, and to the flexible skin design process. The membrane structures observed in Chapter 4 may be similar to the arrangement of the kinematic members needed to achieve shape change.

With the insight gleaned from structural layouts, some experimentation is probably in order to test the stiffness of the structures. For instance, a study could be done to determine how much rigidity battens add to a structure in two-dimensional stress, or how much rigidity veins or branches add to a beam undergoing bending stress. Also, the inherent strength of a wishbone structure to resist being spread apart could be explored. A final pattern observed in the results had a layout similar to a cobweb. To aid the investigation, these structural formations could be more directly correlated to the wings of birds and insects.

*Appendix A. Derivation of General Quadrilateral Plate Bending  
Element Stiffness Matrix*

In order to discretize any general wing planform shape, given that all sides are straight lines and the tip does not have zero length, a general quadrilateral plate element formulation with a transformation mapping is necessary to model out-of-plane bending loads. In order to formulate the stiffness matrix of such an element, the moment/curvature relations will be developed for a plate element following the Kirchhoff assumptions and resulting in the constitutive relation

$$\{M\} = [D]\{\kappa\} \quad (\text{A.1})$$

Next, a general quadrilateral element with 12 degrees of freedom will be selected and its displacement functions will be defined. Then, the curvature/displacement matrix will be developed such that

$$\{\kappa\} = [B]\{d\} \quad (\text{A.2})$$

Thus the moments will be related to the displacements through substitution of Eq. (A.2) into Eq. (A.1) so that

$$\{M\} = [D][B]\{d\} \quad (\text{A.3})$$

From the principle of minimum potential energy, the stiffness matrix becomes

$$[k] = \iint_A [B]^T [D] [B] dx dy \quad (\text{A.4})$$

Finally, a method for evaluating the stiffness matrix using Gaussian quadrature will be established. In addition to developing the theory, Section A.6 describes the process for actually computing the stiffness matrix.

### *A.1 Definition of Moment/Curvature Relationship*

For the sake of simplicity, this derivation follows classical Kirchhoff plate theory (2D extension of Euler-Bernoulli beam theory). When minimizing a structure's compliance, which is the intended purpose of this model, the actual magnitudes of nodal displacements are not as important as the relative magnitude of nodal displacements all throughout the structure. Therefore additional accuracy in deformation is not needed. Proceeding then with a Kirchhoff plate element, the following assumptions are made:

1. Transverse shear strains are neglected.
2. Normal strain is neglected.
3. Normal stress is considered negligible.
4. Membrane forces are neglected.

Following Kirchhoff assumptions, a point on the plate undergoes displacements in the  $x$  and  $y$  directions due to small rotations  $\alpha$  and  $\beta$ , respectively:

$$u = -z\alpha = -z\frac{\partial w}{\partial x} \quad ; \quad v = -z\beta = -z\frac{\partial w}{\partial y} \quad (\text{A.5})$$

As expected, the displacements are proportional to how far the point is from the middle of the plate ( $z = 0$ ). The curvatures of the plate are defined by the rates of change of the angular displacements ( $\alpha$  and  $\beta$ ) of the normal lines:

$$\kappa_x = -\frac{\partial^2 w}{\partial x^2} \quad ; \quad \kappa_y = -\frac{\partial^2 w}{\partial y^2} \quad ; \quad \kappa_{xy} = -2\frac{\partial^2 w}{\partial x \partial y} \quad (\text{A.6})$$

From elasticity theory, the in-plane strains are defined as

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad ; \quad \varepsilon_y = \frac{\partial v}{\partial y} \quad ; \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad (\text{A.7})$$

Operating on Eq. (A.5) with Eq. (A.7) gives the in-plane strain/displacement relations

$$\varepsilon_x = -z \frac{\partial^2 w}{\partial x^2} \quad ; \quad \varepsilon_y = -z \frac{\partial^2 w}{\partial y^2} \quad ; \quad \gamma_{xy} = -2z \frac{\partial^2 w}{\partial x \partial y} \quad (\text{A.8})$$

which can also be expressed (using Eq. (A.6)) as

$$\varepsilon_x = z\kappa_x \quad ; \quad \varepsilon_y = z\kappa_y \quad ; \quad \gamma_{xy} = z\kappa_{xy} \quad (\text{A.9})$$

As can be seen by Eq. (A.9), the strains vary through the thickness of the plate, and so the stresses also must; but the curvatures do not. By neglecting the normal stress in the  $z$  direction, the plane stress equations for an isotropic material relate membrane stresses to membrane strains in the following manner:

$$\sigma_x = \frac{E}{1 - \nu^2} (\varepsilon_x + \nu \varepsilon_y) \quad (\text{A.10a})$$

$$\sigma_y = \frac{E}{1 - \nu^2} (\varepsilon_y + \nu \varepsilon_x) \quad (\text{A.10b})$$

$$\tau_{xy} = \frac{E}{2(1 + \nu)} \gamma_{xy} \quad (\text{A.10c})$$

Substitution of the strains (Eqs. (A.9)) into the stresses (Eqs. (A.10)) gives

$$\sigma_x = \frac{zE}{1 - \nu^2} (\kappa_x + \nu \kappa_y) \quad (\text{A.11a})$$

$$\sigma_y = \frac{zE}{1 - \nu^2} (\kappa_y + \nu \kappa_x) \quad (\text{A.11b})$$

$$\tau_{xy} = \frac{zE}{2(1 + \nu)} \kappa_{xy} \quad (\text{A.11c})$$

In order to obtain a quantity related to stress that does not vary through the thickness, the bending moments are computed by integrating the stresses multiplied by the moment arm  $z$  over the thickness of the plate. This yields the moment/curvature

relations:

$$M_x = \int_{-t/2}^{t/2} z\sigma_x dz = \frac{E}{1-\nu^2}(\kappa_x + \nu\kappa_y) \int_{-t/2}^{t/2} z^2 dz = \frac{Et^3}{12(1-\nu^2)}(\kappa_x + \nu\kappa_y) \quad (\text{A.12a})$$

$$M_y = \int_{-t/2}^{t/2} z\sigma_y dz = \frac{E}{1-\nu^2}(\kappa_y + \nu\kappa_x) \int_{-t/2}^{t/2} z^2 dz = \frac{Et^3}{12(1-\nu^2)}(\kappa_y + \nu\kappa_x) \quad (\text{A.12b})$$

$$M_{xy} = \int_{-t/2}^{t/2} z\tau_{xy} dz = \frac{E}{2(1+\nu)}\kappa_{xy} \int_{-t/2}^{t/2} z^2 dz = \frac{Et^3}{12(1-\nu^2)} \frac{1-\nu}{2} \kappa_{xy} \quad (\text{A.12c})$$

Defining the bending rigidity of the plate as

$$D = \frac{Et^3}{12(1-\nu^2)} \quad (\text{A.13})$$

the moment/curvature relations are simply expressed as

$$M_x = D(\kappa_x + \nu\kappa_y) \quad (\text{A.14a})$$

$$M_y = D(\kappa_y + \nu\kappa_x) \quad (\text{A.14b})$$

$$M_{xy} = \frac{1}{2}D(1-\nu)\kappa_{xy} \quad (\text{A.14c})$$

In matrix form, the constitutive equation is

$$\{M\} = \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = D \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} = [D]\{\kappa\} \quad (\text{A.15})$$

In light of the normal finite element relations of strains to stresses, the bending moments are analogous to stresses and the curvatures are analogous to the strains. With Eq. (A.15), Eq. (A.1) has been established.

## ***A.2 Element Selection and Displacement Definitions***

As indicated by Eq. (A.5), the plate element is free to translate out-of-plane, as well as to rotate out-of-plane. There will therefore be three degrees of freedom at each



node—a transverse displacement  $w$  and two rotations  $\theta_x$  and  $\theta_y$ . The element must also be transformed to a square computational domain. This mapping is depicted below in Figure A.1, where representative loads are shown at node  $i$ . The natural

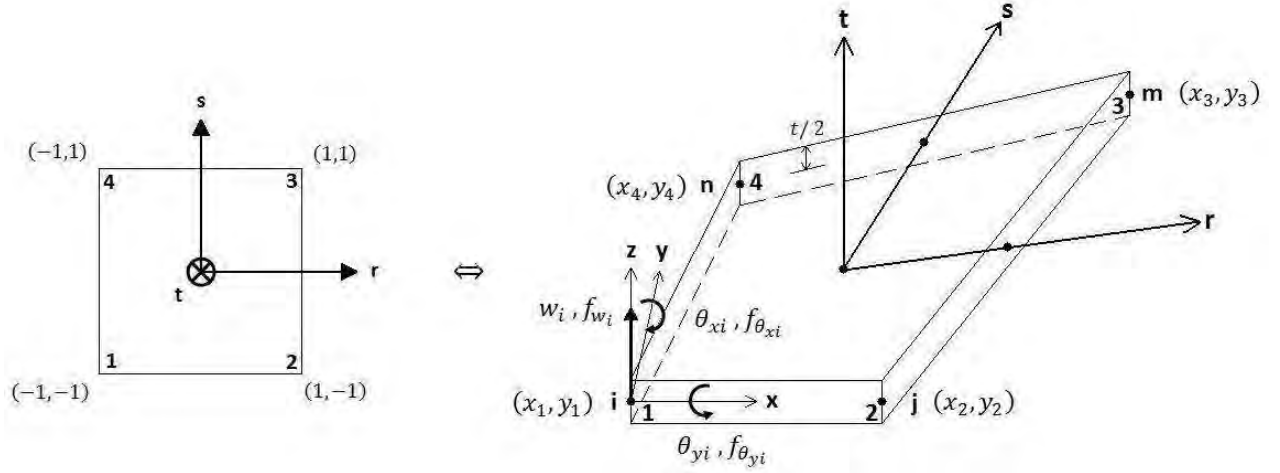


Figure A.1: Mapping of General Quadrilateral Element to Computational Domain

coordinate axes  $r$  and  $s$  need not be parallel to the  $x$  and  $y$  axes or orthogonal to each other. The edges of the element are bounded by  $\pm 1$ . The element has a total of 12 degrees of freedom—three at each of the 4 nodes.

$$\{d\} = \begin{Bmatrix} d_i \\ d_j \\ d_m \\ d_n \end{Bmatrix} ; \quad \{d_i\} = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{Bmatrix} \quad (\text{A.16})$$

Writing out the 12 displacements gives

$$\{d\} = \left\{ w_i \quad \theta_{xi} \quad \theta_{yi} \quad w_j \quad \theta_{xj} \quad \theta_{yj} \quad w_m \quad \theta_{xm} \quad \theta_{ym} \quad w_n \quad \theta_{xn} \quad \theta_{yn} \right\}^T \quad (\text{A.17})$$

Before defining a displacement function  $w(x, y)$ , a mapping between the natural and physical coordinates of the quadrilateral defined by its four corners must be set up. For a first-order scheme with eight knowns (8 nodal coordinates), bilinear shapes are

assumed for both  $x$  and  $y$ :

$$x = A_1 + A_2r + A_3s + A_4rs \quad (\text{A.18a})$$

$$y = A_5 + A_6r + A_7s + A_8rs \quad (\text{A.18b})$$

The  $A$  coefficients are then determined by evaluating  $x$  and  $y$  at the four nodes  $(-1,-1)$ ,  $(1,-1)$ ,  $(1,1)$ , and  $(-1,1)$ , and then simultaneously solving for the  $A$ 's.

$$\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{Bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{Bmatrix} \quad (\text{A.19a})$$

$$\begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{Bmatrix} A_5 \\ A_6 \\ A_7 \\ A_8 \end{Bmatrix} \quad (\text{A.19b})$$

Inverting the 4 x 4 matrices leads to the solution of the coefficients:

$$\begin{Bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{Bmatrix} = \frac{1}{4} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_1 & x_2 & x_3 & -x_4 \\ -x_1 & -x_2 & x_3 & x_4 \\ x_1 & -x_2 & x_3 & -x_4 \end{bmatrix} \quad (\text{A.20a})$$

$$\begin{Bmatrix} A_5 \\ A_6 \\ A_7 \\ A_8 \end{Bmatrix} = \frac{1}{4} \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \\ -y_1 & y_2 & y_3 & -y_4 \\ -y_1 & -y_2 & y_3 & y_4 \\ y_1 & -y_2 & y_3 & -y_4 \end{bmatrix} \quad (\text{A.20b})$$

Substituting these expressions for the  $A$ 's back into Eq. (A.18) and regrouping leads to the following mapping:

$$x = \frac{1}{4}[(1-r)(1-s)x_1 + (1+r)(1-s)x_2 + (1+r)(1+s)x_3 + (1-r)(1+s)x_4] \quad (\text{A.21a})$$

$$y = \frac{1}{4}[(1-r)(1-s)y_1 + (1+r)(1-s)y_2 + (1+r)(1+s)y_3 + (1-r)(1+s)y_4] \quad (\text{A.21b})$$

These coordinates can be expressed in matrix form in the following manner:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{Bmatrix} \quad (\text{A.22})$$

where the shape functions  $N_i$  relating the nodal coordinates  $(x_i, y_i)$  to any  $(x, y)$  position along the plate are expressed by the natural coordinates  $(r, s)$ :

$$N_1 = \frac{1}{4}(1-r)(1-s) \quad ; \quad N_2 = \frac{1}{4}(1+r)(1-s) \quad (\text{A.23a})$$

$$N_3 = \frac{1}{4}(1+r)(1+s) \quad ; \quad N_4 = \frac{1}{4}(1-r)(1+s) \quad (\text{A.23b})$$

Since they will be needed later, the shape function partial derivatives with respect to the natural coordinates are

$$\frac{\partial N_1}{\partial r} = \frac{1}{4}(-1+s); \quad \frac{\partial N_2}{\partial r} = \frac{1}{4}(1-s); \quad \frac{\partial N_3}{\partial r} = \frac{1}{4}(1+s); \quad \frac{\partial N_4}{\partial r} = \frac{1}{4}(-1-s) \quad (\text{A.24a})$$

$$\frac{\partial N_1}{\partial s} = \frac{1}{4}(-1+r); \quad \frac{\partial N_2}{\partial s} = \frac{1}{4}(-1-r); \quad \frac{\partial N_3}{\partial s} = \frac{1}{4}(1+r); \quad \frac{\partial N_4}{\partial s} = \frac{1}{4}(1-r) \quad (\text{A.24b})$$

The partial derivatives of the  $x$  and  $y$  coordinates with respect to the natural coordinates are then

$$\begin{Bmatrix} x_r \\ x_s \end{Bmatrix} = \frac{1}{4} \begin{bmatrix} -1+s & 1-s & 1+s & -1-s \\ -1+r & -1-r & 1+r & 1-r \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} \quad (\text{A.25a})$$

$$\begin{Bmatrix} y_r \\ y_s \end{Bmatrix} = \frac{1}{4} \begin{bmatrix} -1+s & 1-s & 1+s & -1-s \\ -1+r & -1-r & 1+r & 1-r \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix} \quad (\text{A.25b})$$

Given that there are 12 degrees of freedom, a 12-term polynomial is selected for the displacement function  $w(x, y)$  that allows for rigid body motion and constant strain:

$$\begin{aligned} w = & a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 + a_7x^3 + a_8x^2y \\ & + a_9xy^2 + a_{10}y^3 + a_{11}x^3y + a_{12}xy^3 \end{aligned} \quad (\text{A.26})$$

The rotations related to the transverse displacements, used in Eqs. (A.5), are

$$\theta_x = \frac{\partial w}{\partial y} \quad ; \quad \theta_y = -\frac{\partial w}{\partial x} \quad (\text{A.27})$$

Operating on Eqs. (A.26) with Eqs. (A.27), the expressions for the rotation degrees of freedom are

$$\theta_x = a_3 + a_5x + 2a_6y + a_8x^2 + 2a_9xy + 3a_{10}y^2 + a_{11}x^3 + 3a_{12}xy^2 \quad (\text{A.28a})$$

$$\theta_y = -a_2 - 2a_4x - a_5y - 3a_7x^2 - 2a_8xy - a_9y^2 - 3a_{11}x^2y - a_{12}y^3 \quad (\text{A.28b})$$

In matrix form, the displacements are expressed as

$$\begin{pmatrix} w \\ \theta_x \\ \theta_y \end{pmatrix} = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 & x^3 & x^2y & xy^2 & y^3 & x^3y & xy^3 \\ 0 & 0 & 1 & 0 & x & 2y & 0 & x^2 & 2xy & 3y^2 & x^3 & 3xy^2 \\ 0 & -1 & 0 & -2x & -y & 0 & -3x^2 & -2xy & -y^2 & 0 & -3x^2y & -y^3 \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \end{pmatrix} \quad (\text{A.29})$$

or more succinctly by the notation

$$\{\psi\} = [P]\{a\} \quad (\text{A.30})$$

To determine the  $a$  coefficients,  $\{\psi\}$  is evaluated at the 4 nodal points:

$$\begin{pmatrix} w_1 \\ \theta_{x1} \\ \theta_{y1} \\ w_2 \\ \theta_{x2} \\ \theta_{y2} \\ w_3 \\ \theta_{x3} \\ \theta_{y3} \\ w_4 \\ \theta_{x4} \\ \theta_{y4} \end{pmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 & x_1^3 & x_1^2y_1 & x_1y_1^2 & y_1^3 & x_1^3y_1 & x_1y_1^3 \\ 0 & 0 & 1 & 0 & x_1 & 2y_1 & 0 & x_1^2 & 2x_1y_1 & 3y_1^2 & x_1^3 & 3x_1y_1^2 \\ 0 & -1 & 0 & -2x_1 & -y_1 & 0 & -3x_1^2 & -2x_1y_1 & -y_1^2 & 0 & -3x_1^2y_1 & -y_1^3 \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 & x_2^3 & x_2^2y_2 & x_2y_2^2 & y_2^3 & x_2^3y_2 & x_2y_2^3 \\ 0 & 0 & 1 & 0 & x_2 & 2y_2 & 0 & x_2^2 & 2x_2y_2 & 3y_2^2 & x_2^3 & 3x_2y_2^2 \\ 0 & -1 & 0 & -2x_2 & -y_2 & 0 & -3x_2^2 & -2x_2y_2 & -y_2^2 & 0 & -3x_2^2y_2 & -y_2^3 \\ 1 & x_3 & y_3 & x_3^2 & x_3y_3 & y_3^2 & x_3^3 & x_3^2y_3 & x_3y_3^2 & y_3^3 & x_3^3y_3 & x_3y_3^3 \\ 0 & 0 & 1 & 0 & x_3 & 2y_3 & 0 & x_3^2 & 2x_3y_3 & 3y_3^2 & x_3^3 & 3x_3y_3^2 \\ 0 & -1 & 0 & -2x_3 & -y_3 & 0 & -3x_3^2 & -2x_3y_3 & -y_3^2 & 0 & -3x_3^2y_3 & -y_3^3 \\ 1 & x_4 & y_4 & x_4^2 & x_4y_4 & y_4^2 & x_4^3 & x_4^2y_4 & x_4y_4^2 & y_4^3 & x_4^3y_4 & x_4y_4^3 \\ 0 & 0 & 1 & 0 & x_4 & 2y_4 & 0 & x_4^2 & 2x_4y_4 & 3y_4^2 & x_4^3 & 3x_4y_4^2 \\ 0 & -1 & 0 & -2x_4 & -y_4 & 0 & -3x_4^2 & -2x_4y_4 & -y_4^2 & 0 & -3x_4^2y_4 & -y_4^3 \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \end{pmatrix} \quad (\text{A.31})$$

This matrix equation is denoted by

$$\{d\} = [C]\{a\} \quad (\text{A.32})$$

The inverse of  $[C]$  is used to solve the equation for the  $a$ 's. These coefficients can then be substituted back into Eq. (A.30) to obtain the following:

$$\{\psi\} = [P]\{a\} = [P][C]^{-1}\{d\} = [N]\{d\}, \quad \text{where } [N] = [P][C]^{-1} \quad (\text{A.33})$$

### A.3 Definition of Curvature/Displacement Relationship

The next step is to build the curvature/displacement relationship. The curvatures (Eq. (A.6)) can be resolved into a [3 x 3] matrix, denoted as  $[D']$ , that operates on the displacement matrix  $\{\psi\}$ :

$$\{\kappa\} = \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} = \begin{Bmatrix} -\frac{\partial^2 w}{\partial x^2} \\ -\frac{\partial^2 w}{\partial y^2} \\ -\frac{\partial^2 w}{\partial x \partial y} - \frac{\partial^2 w}{\partial y \partial x} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & \frac{\partial(\cdot)}{\partial x} \\ 0 & -\frac{\partial(\cdot)}{\partial y} & 0 \\ 0 & -\frac{\partial(\cdot)}{\partial x} & \frac{\partial(\cdot)}{\partial y} \end{bmatrix} \begin{Bmatrix} w \\ \frac{\partial w}{\partial x} \\ \frac{\partial w}{\partial y} \end{Bmatrix} \quad (\text{A.34})$$

Since  $\{\psi\}$  is actually  $\{\psi(x(r, s), y(r, s))\}$  the chain rule will need to be employed in order to obtain the partial derivatives of  $\{\psi\}$  with respect to  $x$  and  $y$ :

$$\frac{\partial(\cdot)}{\partial r} = \frac{\partial(\cdot)}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial(\cdot)}{\partial y} \frac{\partial y}{\partial r} \quad ; \quad \frac{\partial(\cdot)}{\partial s} = \frac{\partial(\cdot)}{\partial x} \frac{\partial x}{\partial s} + \frac{\partial(\cdot)}{\partial y} \frac{\partial y}{\partial s} \quad (\text{A.35})$$

Using Cramer's rule to simultaneously solve for the partial derivatives with respect to  $x$  and  $y$  leads to the Jacobian matrix  $[J]$ :

$$\frac{\partial(\cdot)}{\partial x} = \frac{1}{|J|} \begin{vmatrix} \frac{\partial(\cdot)}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial(\cdot)}{\partial s} & \frac{\partial y}{\partial s} \end{vmatrix} \quad ; \quad \frac{\partial(\cdot)}{\partial y} = \frac{1}{|J|} \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial(\cdot)}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial(\cdot)}{\partial s} \end{vmatrix} \quad (\text{A.36})$$

$$\text{where} \quad [J] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \quad (\text{A.37})$$

Expanding the determinants, the expressions (Eq. (A.36)) reduce to:

$$\frac{\partial(\cdot)}{\partial x} = \frac{1}{|J|} \left[ \frac{\partial(\cdot)}{\partial r} y_s - \frac{\partial(\cdot)}{\partial s} y_r \right] \quad ; \quad \frac{\partial(\cdot)}{\partial y} = \frac{1}{|J|} \left[ \frac{\partial(\cdot)}{\partial s} x_r - \frac{\partial(\cdot)}{\partial r} x_s \right] \quad (\text{A.38})$$

$$\text{where} \quad |J| = x_r y_s - x_s y_r \quad (\text{A.39})$$

Substituting these relations into Eq. (A.34) yields

$$\{\kappa\} = \frac{1}{|J|} \begin{bmatrix} 0 & 0 & \frac{\partial(\cdot)}{\partial r}y_s - \frac{\partial(\cdot)}{\partial s}y_r \\ 0 & -\frac{\partial(\cdot)}{\partial s}x_r + \frac{\partial(\cdot)}{\partial r}x_s & 0 \\ 0 & -\frac{\partial(\cdot)}{\partial r}y_s + \frac{\partial(\cdot)}{\partial s}y_r & \frac{\partial(\cdot)}{\partial s}x_r - \frac{\partial(\cdot)}{\partial r}x_s \end{bmatrix} \begin{Bmatrix} w \\ \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} \end{Bmatrix} \quad (\text{A.40})$$

or in compact matrix form:  $\{\kappa\} = [D']\{\psi\}$ . Substituting Eq. (A.33) in for  $\{\psi\}$  reveals that  $[D']$  actually operates on  $[P]$ :

$$\{\kappa\} = [D'][P][C]^{-1}\{d\} \quad (\text{A.41})$$

Since  $[P]$  (expressed in Eq. (A.29)) is comprised of  $x$  and  $y$  coordinates and  $[D']$  computes the partial derivatives with respect to the natural coordinates, the chain rule is employed to perform the operation. The first column of  $[D']$  is  $[0]$  such that the derivatives only need to be determined for rows 2 and 3 of  $[P]$  (i.e.  $P_{2j}$  and  $P_{3j}$ ).

$$\frac{\partial P_{2j}}{\partial r} = \begin{bmatrix} 0 & 0 & 0 & 0 & x_r & 2y_r & 0 & 2xx_r & 2(x_r y + xy_r) \\ & & & & 6yy_r & 3x^2x_r & 3(x_r y^2 + 2xyy_r) \end{bmatrix} \quad (\text{A.42})$$

$$\frac{\partial P_{3j}}{\partial r} = - \begin{bmatrix} 0 & 0 & 0 & 2x_r & y_r & 0 & 6xx_r & 2(x_r y + xy_r) & 2yy_r \\ & & & 0 & 3(2xx_r y + x^2y_r) & 3y^2y_r \end{bmatrix} \quad (\text{A.43})$$

$$\frac{\partial P_{2j}}{\partial s} = \begin{bmatrix} 0 & 0 & 0 & 0 & x_s & 2y_s & 0 & 2xx_s & 2(x_s y + xy_s) \\ & & & & 6yy_s & 3x^2x_s & 3(x_s y^2 + 2xyy_s) \end{bmatrix} \quad (\text{A.44})$$

$$\frac{\partial P_{3j}}{\partial s} = - \begin{bmatrix} 0 & 0 & 0 & 2x_s & y_s & 0 & 6xx_s & 2(x_s y + xy_s) & 2yy_s \\ & & & 0 & 3(2xx_s y + x^2y_s) & 3y^2y_s \end{bmatrix} \quad (\text{A.45})$$

After computing these partial derivatives, the product or operation of  $[D']$  onto  $[P]$  results in  $[Q]$ , whose columns are computed as follows:

$$[Q] = [D'][P] = \frac{1}{|J|} \begin{bmatrix} [Q_1] & \dots & [Q_{12}] \end{bmatrix} \quad (\text{A.46})$$

where

$$[Q_i] = \begin{bmatrix} y_s(P_{3i})_r - y_r(P_{3i})_s \\ -x_r(P_{2i})_s + x_s(P_{2i})_r \\ -y_s(P_{2i})_r + y_r(P_{2i})_s + x_r(P_{3i})_s - x_s(P_{3i})_r \end{bmatrix} \quad (\text{A.47})$$

With  $[Q]$ , the curvature/displacement matrix  $[B]$  can now be computed from Eq. (A.41):

$$\begin{aligned} \{\kappa\} &= [D'][P][C]^{-1}\{d\} = [Q][C]^{-1}\{d\} = [B]\{d\} \\ \text{where } [B] &= [Q][C]^{-1} \end{aligned} \quad (\text{A.48})$$

Thus the curvature/displacement equation (Eq. (A.2)) has been established, and by Eq. (A.3), the moments are now directly related to the displacements:

$$\{M\} = [D][B]\{d\} \quad (\text{A.49})$$

#### ***A.4 Stiffness Matrix Formulation***

To derive the stiffness matrix, the principle of minimum potential energy is typically employed. The total potential energy of the plate is the integral over the volume of the product of the stress and strain terms:

$$U = \frac{1}{2} \iiint (\sigma_x \varepsilon_x + \sigma_y \varepsilon_y + \tau_{xy} \gamma_{xy}) dV \quad (\text{A.50})$$

Substituting Eqs. (A.9) and (A.12) into Eq. (A.50) yields

$$U = \frac{1}{2} \iint (M_x \kappa_x + M_y \kappa_y + M_{xy} \kappa_{xy}) dA \quad (\text{A.51})$$



After applying the principle of minimum potential energy, the stiffness matrix is then computed via a double integral over the area of the element:

$$[k] = \iint [B]^T [D] [B] dx dy \quad (\text{A.52})$$

A general derivation of this form of the stiffness matrix using the principle of minimum potential energy is given in [23, pgs. 317-319], but is omitted here. Since  $[B]$  is a function of  $r$  and  $s$ , the integration must be performed with respect to  $r$  and  $s$ . The transformation of the variables, as well as the region from  $x$  and  $y$  to  $r$  and  $s$ , involves the determinant of the Jacobian matrix:

$$\iint_A f(x, y) dx dy = \iint_A f(r, s) |J| dr ds \quad (\text{A.53})$$

Thus  $[k]$  becomes

$$[k] = \int_{-1}^1 \int_{-1}^1 [B]^T [D] [B] |J| dr ds \quad (\text{A.54})$$

The order of the stiffness matrix is  $[12 \times 12]$ . Since  $[B]$  is a rather complicated expression, the integration determining the element stiffness matrix is best carried out numerically. This derivation uses the Gaussian quadrature method to evaluate the integration.

### ***A.5 Gaussian Quadrature Method of Integration***

A more complete explanation of the Gaussian quadrature method is given in Section 10.4 of [23]. The basic idea of the method is to approximate an integral by summing some weighted sampling points within the interval of integration:

$$I = \int_a^b f(r) dr = \sum_{i=1}^n W_i f_i \quad (\text{A.55})$$

This basic notion can be extended in multiple dimensions. The integration of concern here consists of two integrals over the interval  $[-1, 1]$  (Figure A.2). Using Gaussian

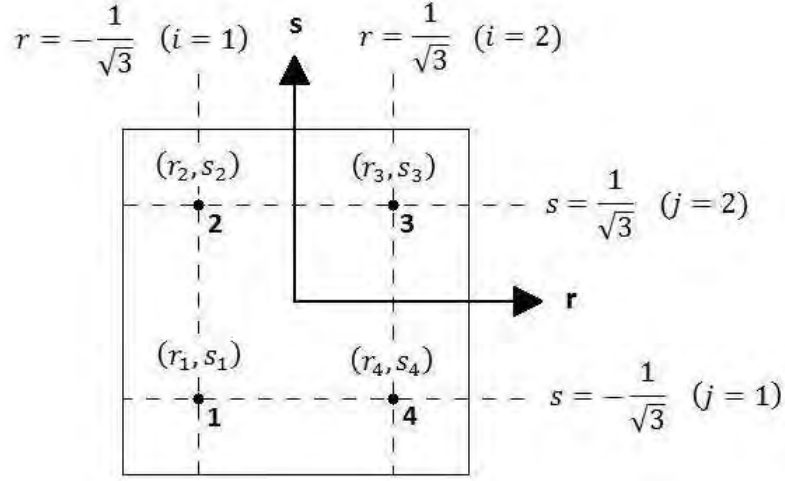


Figure A.2: 2 x 2 Gaussian Quadrature Evaluation of Stiffness Integral

quadrature results in a double summation:

$$\begin{aligned}
 I &= \int_{-1}^1 \int_{-1}^1 f(r, s) dr ds = \int_{-1}^1 \left[ \sum_{i=1}^n W_i f(r_i, s) \right] ds \\
 &= \sum_{j=1}^m W_j \left[ \sum_{i=1}^n W_i f(r_i, s_j) \right] = \sum_{i=1}^n \sum_{j=1}^m W_i W_j f(r_i, s_j) \quad (\text{A.56})
 \end{aligned}$$

A 2 x 2 Gaussian rule is considered sufficient accuracy for the purposes of this model. If more accuracy is desired, more points can be sampled. With  $n = m = 2$ , the double summation can be written out as follows:

$$I = W_1 W_1 f(r_1, s_1) + W_1 W_2 f(r_1, s_2) + W_2 W_1 f(r_2, s_1) + W_2 W_2 f(r_2, s_2) \quad (\text{A.57})$$

The associated weights determined by a 2 x 2 Gaussian rule are all equal to 1, and the point locations are  $r_i, s_i = \pm 1/\sqrt{3}$  for integration over the interval  $[-1, 1]$ . Evaluating

the stiffness integrand (Eq. (A.52)) at these points results in the following:

$$\begin{aligned}
[k] = & \left[ B \left( -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right) \right]^T [D] \left[ B \left( -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right) \right] \left| J \left( -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right) \right| \\
& + \left[ B \left( -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right) \right]^T [D] \left[ B \left( -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right) \right] \left| J \left( -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right) \right| \\
& + \left[ B \left( \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right) \right]^T [D] \left[ B \left( \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right) \right] \left| J \left( \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right) \right| \\
& + \left[ B \left( \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right) \right]^T [D] \left[ B \left( \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right) \right] \left| J \left( \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right) \right| \quad (\text{A.58})
\end{aligned}$$

### A.6 Computation of Stiffness Matrix

A summary of the steps necessary to compute the stiffness matrix is enumerated here.

1. Obtain the four nodal coordinates and the three constitutive properties Young's modulus  $E$ , Poisson's ratio  $\nu$ , and the element thickness  $t$ . These are the only inputs needed to determine the element stiffness.
2. Build the constitutive matrix  $[D]$  using Eqs. (A.13) and (A.15).
3. Compute the  $[C]$  matrix from Eq. (A.31).
4. Set-up a summation scheme to sum the remaining steps over the Gaussian points given by Eq. (A.58) (steps 5-9).
5. Calculate the bilinear shape functions (Eqs. (A.23)), the corresponding physical coordinates  $x$  and  $y$  (Eqs. (A.22)), and the partial derivatives of the physical coordinates with respect to the natural coordinates (Eqs. (A.25)).
6. Calculate the partial derivatives of the  $[P]$  matrix using Eq. (A.42) through Eq. (A.45).
7. Build the  $[Q]$  matrix using Eqs. (A.47) and (A.46). The determinant  $|J|$  can be calculated from Eq. (A.39).
8. Calculate the  $[B]$  matrix from Eq. (A.48).
9. Finally,  $[k]$  is calculated by summing the terms in Eq. (A.58).

## Appendix B. Finite Element MATLAB Codes

Listing B.1: Membrane Finite Element Model

```
1 %% FINITE ELEMENT MEMBRANE ANALYSIS
function FEm(params, varargin)

%Setup Parameters
if isempty(varargin)==1
6   config=1;
elseif isempty(varargin)==0
   config=varargin{1};
end
params=params{config};
11 if isfield(params, 'points')==0
    mh=params.mesh;
elseif isfield(params, 'points')==1
    mh=params.points;
end
16 nelxp=params.n;
[~,n]=size(nelxp);
nelx=zeros(n);
for i=1:n
    nelx(i)=nelxp{i}-1;
21 end
nelx=sum(nelx);
nely=params.m{1}-1;
nel=nelx*nely;

26 %Evaluate Element Stiffness Matrices
edof=zeros(8,nel);KE=zeros(8*nel,8);
X=zeros(4,nel);Y=zeros(4,nel);
for el=1:nel
    m=floor((el-1)/nely);
31   n=rem(el-1,nely)+1;
    edof(:,el)=[2*m*(nely+1)+2*n+1;2*m*(nely+1)+2*n+2;
                2*(m+1)*(nely+1)+2*n+1;2*(m+1)*(nely+1)+2*n+2;
                2*(m+1)*(nely+1)+2*n-1;2*(m+1)*(nely+1)+2*n;
                2*m*(nely+1)+2*n-1;2*m*(nely+1)+2*n];
36   mhi= [m*(nely+1)+n+1;(m+1)*(nely+1)+n+1;
          (m+1)*(nely+1)+n;m*(nely+1)+n];
    X(:,el)=[mh(mhi(1),1);mh(mhi(2),1);mh(mhi(3),1);mh(mhi(4),1)];
    Y(:,el)=[mh(mhi(1),2);mh(mhi(2),2);mh(mhi(3),2);mh(mhi(4),2)];
    [KE(8*(el-1)+1:8*el,1:8),B0(3*(el-1)+1:3*el,1:8),D]=...
41   Ke(X(:,el),Y(:,el));
end

%Calculate Global Displacements
[U]=FE(nelx,nely,edof,KE);
46 %Visualize Results
u=U(1:2:2*(nelx+1)*(nely+1),1);
v=U(2:2:2*(nelx+1)*(nely+1),1);
Xa=mh(:,1)+u;
```

```

51 Ya=mh(:,2)+v;
   X=zeros(4,nel); Y=zeros(4,nel);
   color=zeros(4,nel);
   for el=1:nel
       m=floor((el-1)/nely);
56   n=rem(el-1,nely)+1;
       mhi=[m*(nely+1)+n;(m+1)*(nely+1)+n;
           (m+1)*(nely+1)+n+1;m*(nely+1)+n+1];
       X(:,el)=[Xa(mhi(1),1);Xa(mhi(2),1);Xa(mhi(3),1);Xa(mhi(4),1)];
       Y(:,el)=[Ya(mhi(1),1);Ya(mhi(2),1);Ya(mhi(3),1);Ya(mhi(4),1)];
61   d=U(edof(:,el),1);
       sig=(D*B0(3*(el-1)+1:3*el,1:8)*d);
       prin1=(sig(1)+sig(2))/2+sqrt(((sig(1)-sig(2))/2)^2+sig(3)^2);
       prin2=(sig(1)+sig(2))/2-sqrt(((sig(1)-sig(2))/2)^2+sig(3)^2);
       sig(4)=sqrt(((prin1-prin2)^2+prin1^2+prin2^2)/2);
66   color(:,el)=sig;
       end
       figure;patch(X,Y,color(1,:));
       title('\sigma_x');axis equal;grid;colorbar
       figure;patch(X,Y,color(2,:));
71   title('\sigma_y');axis equal;grid;colorbar
       figure;patch(X,Y,color(3,:));
       title('\tau_{xy}');axis equal;grid;colorbar
       figure;patch(X,Y,color(4,:));
       title('Von Mises Stress');axis equal;grid;colorbar
76   end

%% SOLVE DISPLACEMENTS
81 function [U]=FE(nelx,nely,edof,KE)

%%Compile Global Stiffness Matrix
K=zeros(2*(nelx+1)*(nely+1),2*(nelx+1)*(nely+1));
F=zeros(2*(nely+1)*(nelx+1),1);U=zeros(2*(nely+1)*(nelx+1),1);
86 for el=1:nelx*nely
    K(edof(:,el),edof(:,el))=...
        K(edof(:,el),edof(:,el))+KE(8*(el-1)+1:8*el,1:8);
    end

91 %Define Loads and Supports
F(100,1)=-1000; %INPUT: Apply external forces
fixeddofs=1:2*(nely+1); %INPUT: Constrain degrees of freedom
alldofs=1:2*(nely+1)*(nelx+1);
freedofs=setdiff(alldofs,fixeddofs);
96 U(freedofs,1)=K(freedofs,freedofs)\F(freedofs,1);
U(fixeddofs,:)=0;

    end

101 %% ELEMENT STIFFNESS MATRIX

```

```

function [KE,B0,D]=Ke(X,Y)

%Constitutive Matrix
106 E=200e9; %INPUT: Young's modulus
    nu=0.3; %INPUT: Poisson's ratio
    th=1; %INPUT: thickness of element
    D=E*th/(1-nu^2)*[1 nu 0;nu 1 0;0 0 (1-nu)/2];

111 %Gaussian Quadrature Integration
    R=[0 -sqrt(1/3) -sqrt(1/3) sqrt(1/3) sqrt(1/3)];
    S=[0 -sqrt(1/3) sqrt(1/3) -sqrt(1/3) sqrt(1/3)];
    k=zeros(8,8);
    for i=1:5
116     r=R(i); s=S(i);
        xr=(1/4)*[-1+s 1-s 1+s -1-s]*X;
        xs=(1/4)*[-1+r -1-r 1+r 1-r]*X;
        yr=(1/4)*[-1+s 1-s 1+s -1-s]*Y;
        ys=(1/4)*[-1+r -1-r 1+r 1-r]*Y;
121     detJ=xr*ys-xs*yr;
        N1r=(1/4)*(s-1); N1s=(1/4)*(r-1);
        N2r=(1/4)*(1-s); N2s=(1/4)*(-r-1);
        N3r=(1/4)*(1+s); N3s=(1/4)*(1+r);
        N4r=(1/4)*(-1-s); N4s=(1/4)*(1-r);
126     B1=[ys*N1r-yr*N1s 0;0 xr*N1s-xs*N1r;
           xr*N1s-xs*N1r ys*N1r-yr*N1s];
        B2=[ys*N2r-yr*N2s 0;0 xr*N2s-xs*N2r;
           xr*N2s-xs*N2r ys*N2r-yr*N2s];
        B3=[ys*N3r-yr*N3s 0;0 xr*N3s-xs*N3r;
           xr*N3s-xs*N3r ys*N3r-yr*N3s];
131     B4=[ys*N4r-yr*N4s 0;0 xr*N4s-xs*N4r;
           xr*N4s-xs*N4r ys*N4r-yr*N4s];
        B=(1/detJ)*[B1 B2 B3 B4];
        if i==1
136         B0=B;
        else
            k=k+B'*D*B*detJ;
        end
    end
141 KE=k;

end

```

Listing B.2: Bending Finite Element Model

```

%% FINITE ELEMENT BENDING ANALYSIS
2 function FEb(params,varargin)

%Setup Parameters
if isempty(varargin)==1
    config=1;
7 elseif isempty(varargin)==0
    config=varargin{1};
end

```

```

params=params{config};
if isfield(params,'points')==0
12     mh=params.mesh;
elseif isfield(params,'points')==1
     mh=params.points;
end
nelxp=params.n;
17 [~,n]=size(nelxp);
nelx=zeros(n);
for i=1:n
     nelx(i)=nelxp{i}-1;
end
22 nelx=sum(nelx);
nely=params.m{1}-1;
nel=nelx*nely;

%Elemental Stiffness Matrices
27 edof=zeros(12,nel);KE=zeros(12*nel,12);
Xc=zeros(4,nel);Yc=zeros(4,nel);
for el=1:nel
     m=floor((el-1)/nely);
     n=rem(el-1,nely)+1;
32     edofi=[3*m*(nely+1)+3*n+1 3*(m+1)*(nely+1)+3*n+1 ...
            3*(m+1)*(nely+1)+3*n-2 3*m*(nely+1)+3*n-2];
     edof(:,el)=[edofi(1);edofi(1)+1;edofi(1)+2;
                edofi(2);edofi(2)+1;edofi(2)+2;
                edofi(3);edofi(3)+1;edofi(3)+2;
37                edofi(4);edofi(4)+1;edofi(4)+2];
     mhi=[m*(nely+1)+n+1;(m+1)*(nely+1)+n+1;
          (m+1)*(nely+1)+n;m*(nely+1)+n];
     Xc(:,el)=[mh(mhi(1),1);mh(mhi(2),1);
              mh(mhi(3),1);mh(mhi(4),1)];
42     Yc(:,el)=[mh(mhi(1),2);mh(mhi(2),2);
              mh(mhi(3),2);mh(mhi(4),2)];
     [KE(12*(el-1)+1:12*el,1:12),B0(3*(el-1)+1:3*el,1:12),D]=...
     Ke(Xc(:,el),Yc(:,el));
end
47 %Calculate Global Displacements
[U]=FE(nelx,nely,edof,KE);

%Visualize Results
52 Zc=zeros(4,nel);
color=zeros(4,el);
for el=1:nel
     Zc(:,el)=U(edof([1 4 7 10],el),1);
     d=U(edof(:,el),1);
57     sig=D*B0(3*(el-1)+1:3*el,1:12)*d;
     prin1=(sig(1)+sig(2))/2+sqrt(((sig(1)-sig(2))/2)^2+sig(3)^2);
     prin2=(sig(1)+sig(2))/2-sqrt(((sig(1)-sig(2))/2)^2+sig(3)^2);
     sig(4)=sqrt(((prin1-prin2)^2+prin1^2+prin2^2)/2);
     color(1:4,el)=sig;

```

```

62 end
figure;patch(Xc,Yc,Zc,color(1,:));
title('\sigma_x');axis equal;grid;colorbar
figure;patch(Xc,Yc,Zc,color(2,:));
title('\sigma_y');axis equal;grid;colorbar
67 figure;patch(Xc,Yc,Zc,color(3,:));
title('\tau_{xy}');axis equal;grid;colorbar
figure;patch(Xc,Yc,Zc,color(4,:));
title('\sigma_{vm}');axis equal;grid;colorbar

72 end

%% SOLVE DISPLACEMENTS
function [U]=FE(nelx,nely,edof,KE)
77
%Compile Global Stiffness Matrix
K=zeros(3*(nelx+1)*(nely+1),3*(nelx+1)*(nely+1));
F=zeros(3*(nely+1)*(nelx+1),1);U=zeros(3*(nely+1)*(nelx+1),1);
for el=1:nelx*nely
82     K(edof(:,el),edof(:,el))=...
        K(edof(:,el),edof(:,el))+KE(12*(el-1)+1:12*el,1:12);
end

%Define Loads and Supports
87 F(100,1)=-1000; %INPUT: Apply external forces
fixeddofs=1:3*(nely+1); %INPUT: Constrain degrees of freedom
alldofs=1:3*(nely+1)*(nelx+1);
freedofs=setdiff(alldofs,fixeddofs);
U(freedofs,:)=K(freedofs,freedofs)\F(freedofs,:);
92 U(fixeddofs,:)=0;

end

97 %% ELEMENT STIFFNESS MATRIX
function [KE,B0,Dz]=Ke(X,Y)

%Constitutive Matrix
E=200e9; %INPUT: Young's modulus
102 nu=0.3; %INPUT: Poisson's ratio
th=1; %INPUT: thickness of element
z=th/2;
D=(E*th^3/(12*(1-nu^2)))*[1 nu 0;nu 1 0;0 0 (1-nu)/2];
Dz=z*E/(1-nu^2)*[1 nu 0;nu 1 0;0 0 (1-nu)/2];
107
%Gaussian Quadrature Integration
x1=X(1); x2=X(2); x3=X(3); x4=X(4);
y1=Y(1); y2=Y(2); y3=Y(3); y4=Y(4);
C=[1 x1 y1 x1^2 x1*y1 y1^2 x1^3 x1^2*y1 x1*y1^2 y1^3 x1^3*y1 x1*y1...
^3
112 0 0 1 0 x1 2*y1 0 x1^2 2*x1*y1 3*y1^2 x1^3 3*x1*y1^2

```



```

0 -1 0 -2*x1 -y1 0 -3*x1^2 -2*x1*y1 -y1^2 0 -3*x1^2*y1 -y1^3
1 x2 y2 x2^2 x2*y2 y2^2 x2^3 x2^2*y2 x2*y2^2 y2^3 x2^3*y2 x2*y2...
^3
0 0 1 0 x2 2*y2 0 x2^2 2*x2*y2 3*y2^2 x2^3 3*x2*y2^2
0 -1 0 -2*x2 -y2 0 -3*x2^2 -2*x2*y2 -y2^2 0 -3*x2^2*y2 -y2^3
117 1 x3 y3 x3^2 x3*y3 y3^2 x3^3 x3^2*y3 x3*y3^2 y3^3 x3^3*y3 x3*y3...
^3
0 0 1 0 x3 2*y3 0 x3^2 2*x3*y3 3*y3^2 x3^3 3*x3*y3^2
0 -1 0 -2*x3 -y3 0 -3*x3^2 -2*x3*y3 -y3^2 0 -3*x3^2*y3 -y3^3
1 x4 y4 x4^2 x4*y4 y4^2 x4^3 x4^2*y4 x4*y4^2 y4^3 x4^3*y4 x4*y4...
^3
0 0 1 0 x4 2*y4 0 x4^2 2*x4*y4 3*y4^2 x4^3 3*x4*y4^2
122 0 -1 0 -2*x4 -y4 0 -3*x4^2 -2*x4*y4 -y4^2 0 -3*x4^2*y4 -y4^3];
R=[0 -sqrt(1/3) -sqrt(1/3) sqrt(1/3) sqrt(1/3)];
S=[0 -sqrt(1/3) sqrt(1/3) -sqrt(1/3) sqrt(1/3)];
k=zeros(12,12);
for i=1:5
127 r=R(i); s=S(i);
N1=(1/4)*(1-r)*(1-s);
N2=(1/4)*(1+r)*(1-s);
N3=(1/4)*(1+r)*(1+s);
N4=(1/4)*(1-r)*(1+s);
132 N=[N1 N2 N3 N4];
x=N*X; y=N*Y;
xr=(1/4)*[-1+s 1-s 1+s -1-s]*X;
xs=(1/4)*[-1+r -1-r 1+r 1-r]*X;
yr=(1/4)*[-1+s 1-s 1+s -1-s]*Y;
137 ys=(1/4)*[-1+r -1-r 1+r 1-r]*Y;
detJ=xr*ys-xs*yr;
P21r=0; P21s=0;
P22r=0; P22s=0;
P23r=0; P23s=0;
142 P24r=0; P24s=0;
P25r=xr; P25s=xs;
P26r=2*yr; P26s=2*ys;
P27r=0; P27s=0;
P28r=2*x*xr; P28s=2*x*xs;
147 P29r=2*(xr*y+x*yr); P29s=2*(xs*y+x*ys);
P210r=6*y*yr; P210s=6*y*ys;
P211r=3*x^2*xr; P211s=3*x^2*xs;
P212r=3*(xr*y^2+2*x*y*yr); P212s=3*(xs*y^2+2*x*y*ys);
P31r=0; P31s=0;
152 P32r=0; P32s=0;
P33r=0; P33s=0;
P34r=-2*xr; P34s=-2*xs;
P35r=-yr; P35s=-ys;
P36r=0; P36s=0;
157 P37r=-6*x*xr; P37s=-6*x*xs;
P38r=-2*(xr*y+x*yr); P38s=-2*(xs*y+x*ys);
P39r=-2*y*yr; P39s=-2*y*ys;
P310r=0; P310s=0;
P311r=-3*(2*x*xr*y+x^2*yr); P311s=-3*(2*x*xs*y+x^2*ys);

```

```

162 P312r=-3*y^2*yr;          P312s=-3*y^2*ys;
    Q1=[ys*P31r-yr*P31s  ys*P32r-yr*P32s  ys*P33r-yr*P33s
        -xr*P21s+xs*P21r  -xr*P22s+xs*P22s  -xr*P23s+xs*P23r
        -ys*P21r+yr*P21s+xr*P31s-xr*P31r  -ys*P22r+yr*P22s...
        +xr*P32s-xr*P32r  -ys*P23r+yr*P23s+xr*P33s-xr*P33r];
167 Q2=[ys*P34r-yr*P34s  ys*P35r-yr*P35s  ys*P36r-yr*P36s
        -xr*P24s+xs*P24r  -xr*P25s+xs*P25r  -xr*P26s+xs*P26r
        -ys*P24r+yr*P24s+xr*P34s-xr*P34r  -ys*P25r+yr*P25s...
        +xr*P35s-xr*P35r  -ys*P26r+yr*P26s+xr*P36s-xr*P36r];
172 Q3=[ys*P37r-yr*P37s  ys*P38r-yr*P38s  ys*P39r-yr*P39s
        -xr*P27s+xs*P27r  -xr*P28s+xs*P28r  -xr*P29s+xs*P29r
        -ys*P27r+yr*P27s+xr*P37s-xr*P37r  -ys*P28r+yr*P28s...
        +xr*P38s-xr*P38r  -ys*P29r+yr*P29s+xr*P39s-xr*P39r];
    Q4=[ys*P310r-yr*P310s  ys*P311r-yr*P311s  ys*P312r-yr*P312s
        -xr*P210s+xs*P210r  -xr*P211s+xs*P211r  -xr*P212s+xs*P212r
177 -ys*P210r+yr*P210s+xr*P310s-xr*P310r...
        -ys*P211r+yr*P211s+xr*P311s-xr*P311r...
        -ys*P212r+yr*P212s+xr*P312s-xr*P312r];
    Q=(1/detJ)*[Q1 Q2 Q3 Q4];
    B=Q/C;
182 if i==1
        B0=B;
    else
        k=k+B'*D*B*detJ;
    end
187 end
    KE=k;

end

```

Listing B.3: Combined Membrane-Bending Finite Element Model

```

%% FINITE ELEMENT MEMBRANE & BENDING SUPERPOSITION ANALYSIS
function FEmb(params,varargin)

%Setup Parameters
5 if isempty(varargin)==1
    config=1;
elseif isempty(varargin)==0
    config=varargin{1};
end
10 params=params{config};
    mh=params.mesh;
    [~,n]=size(params.n);
    nelx=0;
    for i=1:n
15     nelx=nelx+params.n{1}-1;
    end
    nely=params.m{1}-1;
    nel=nelx*nely;

20 %Elemental Stiffness Matrices
    edof=zeros(24,nel);KE=zeros(24*nel,24);

```

```

X=zeros(4,nel);Y=zeros(4,nel);
for el=1:nel
    m=floor((el-1)/nely);
25    n=rem(el-1,nely)+1;
    e=[6*m*(nely+1)+6*n+1 6*(m+1)*(nely+1)+6*n+1 ...
        6*(m+1)*(nely+1)+6*n-5 6*m*(nely+1)+6*n-5];
    edof(:,el)=[e(1);e(1)+1;e(1)+2;e(1)+3;e(1)+4;e(1)+5;
30                e(2);e(2)+1;e(2)+2;e(2)+3;e(2)+4;e(2)+5;
                e(3);e(3)+1;e(3)+2;e(3)+3;e(3)+4;e(3)+5;
                e(4);e(4)+1;e(4)+2;e(4)+3;e(4)+4;e(4)+5];
    mhi=[m*(nely+1)+n+1;(m+1)*(nely+1)+n+1;
        (m+1)*(nely+1)+n;m*(nely+1)+n];
    X(:,el)=[mh(mhi(1),1);mh(mhi(2),1);mh(mhi(3),1);mh(mhi(4),1)];
35    Y(:,el)=[mh(mhi(1),2);mh(mhi(2),2);mh(mhi(3),2);mh(mhi(4),2)];
    [KE(24*(el-1)+1:24*el,1:24),B0m(3*(el-1)+1:3*el,1:8),...
        B0b(3*(el-1)+1:3*el,1:12),Dm,Db]...
        =Ke(X(:,el),Y(:,el));
end
40    %Calculate Global Displacements
    [U]=FE(nelx,nely,edof,KE);

    %Visualize Results
45    u=U(1:6:6*(nelx+1)*(nely+1),1);
    v=U(2:6:6*(nelx+1)*(nely+1),1);
    Xa=mh(:,1)+u; Ya=mh(:,2)+v;
    Z=zeros(4,nel); color=zeros(nel,1);
    for el=1:nel
50        m=floor((el-1)/nely);
        n=rem(el-1,nely)+1;
        mhi=[m*(nely+1)+n+1;(m+1)*(nely+1)+n+1;
            (m+1)*(nely+1)+n;m*(nely+1)+n];
        X(:,el)=[Xa(mhi(1),1);Xa(mhi(2),1);Xa(mhi(3),1);Xa(mhi(4),1)];
55        Y(:,el)=[Ya(mhi(1),1);Ya(mhi(2),1);Ya(mhi(3),1);Ya(mhi(4),1)];
        Z(:,el)=U(edof([3 9 15 21],el),1);
        d=U(edof(:,el));
        dm=d([1:2 7:8 13:14 19:20]);
        db=d([3:5 9:11 15:17 21:23]);
60        sigm=(Dm*B0m(3*(el-1)+1:3*el,1:8)*dm);
        sigb=(Db*B0b(3*(el-1)+1:3*el,1:12)*db);
        sig=sigm+sigb;
        prin1=(sig(1)+sig(2))/2+sqrt(((sig(1)-sig(2))/2)^2+sig(3)^2);
        prin2=(sig(1)+sig(2))/2-sqrt(((sig(1)-sig(2))/2)^2+sig(3)^2);
65        sig(4)=sqrt(((prin1-prin2)^2+prin1^2+prin2^2)/2);
        color(1:4,el)=sig;
    end
    figure;patch(X,Y,Z,color(1,:));
    title('\sigma_x');axis equal;grid;colorbar
70    figure;patch(X,Y,Z,color(2,:));
    title('\sigma_y');axis equal;grid;colorbar
    figure;patch(X,Y,Z,color(3,:));
    title('\tau_{xy}');axis equal;grid;colorbar

```

```

figure;patch(X,Y,Z,color(4,:));
75 title('\sigma_{vm}');axis equal;grid;colorbar

end

80 %% SOLVE DISPLACEMENTS
function [U]=FE(nelx,nely,edof,KE)

%Compile Stiffness Matrix
K=zeros(6*(nelx+1)*(nely+1),6*(nelx+1)*(nely+1));
85 F=zeros(6*(nely+1)*(nelx+1),2);U=zeros(6*(nely+1)*(nelx+1),2);
for el=1:nelx*nely
    K(edof(:,el),edof(:,el))=...
        K(edof(:,el),edof(:,el))+KE(24*(el-1)+1:24*el,1:24);
end
90
%Define Loads and Supports
F(1000,1)=-1000; %INPUT: Apply external forces
fixeddofs=1:100; %INPUT: Constrain degrees of freedom
alldofs=1:6*(nely+1)*(nelx+1);
95 freedofs=setdiff(alldofs,fixeddofs);
U(freedofs,:)=K(freedofs,freedofs)\F(freedofs,:);
U(fixeddofs,:)=0;

end
100

%% ELEMENT STIFFNESS MATRIX
function [KE,B0m,B0b,Dm,Dbz]=Ke(X,Y)

105 %Constitutive Matrix
E=200e9; %INPUT: Young's modulus
nu=0.3; %INPUT: Poisson's ratio
th=0.1; %INPUT: element thickness
z=th/2;
110 D=[1 nu 0;nu 1 0;0 0 (1-nu)/2];
Dm=(E*th/(1-nu^2))*D;
Db=E*th^3/(12*(1-nu^2))*D;
Dbz=z*E/(1-nu^2)*D;

115 %Formulation of Bending [C] Matrix
x1=X(1); x2=X(2); x3=X(3); x4=X(4);
y1=Y(1); y2=Y(2); y3=Y(3); y4=Y(4);
C=[1 x1 y1 x1^2 x1*y1 y1^2 x1^3 x1^2*y1 x1*y1^2 y1^3 x1^3*y1 x1*y1...
    ^3
    0 0 1 0 x1 2*y1 0 x1^2 2*x1*y1 3*y1^2 x1^3 3*x1*y1^2
120 0 -1 0 -2*x1 -y1 0 -3*x1^2 -2*x1*y1 -y1^2 0 -3*x1^2*y1 -y1^3
    1 x2 y2 x2^2 x2*y2 y2^2 x2^3 x2^2*y2 x2*y2^2 y2^3 x2^3*y2 x2*y2...
    ^3
    0 0 1 0 x2 2*y2 0 x2^2 2*x2*y2 3*y2^2 x2^3 3*x2*y2^2
    0 -1 0 -2*x2 -y2 0 -3*x2^2 -2*x2*y2 -y2^2 0 -3*x2^2*y2 -y2^3

```

```

1 x3 y3 x3^2 x3*y3 y3^2 x3^3 x3^2*y3 x3*y3^2 y3^3 x3^3*y3 x3*y3...
  ^3
125 0 0 1 0 x3 2*y3 0 x3^2 2*x3*y3 3*y3^2 x3^3 3*x3*y3^2
0 -1 0 -2*x3 -y3 0 -3*x3^2 -2*x3*y3 -y3^2 0 -3*x3^2*y3 -y3^3
1 x4 y4 x4^2 x4*y4 y4^2 x4^3 x4^2*y4 x4*y4^2 y4^3 x4^3*y4 x4*y4...
  ^3
0 0 1 0 x4 2*y4 0 x4^2 2*x4*y4 3*y4^2 x4^3 3*x4*y4^2
0 -1 0 -2*x4 -y4 0 -3*x4^2 -2*x4*y4 -y4^2 0 -3*x4^2*y4 -y4^3];
130 %Membrane & Bending Gaussian Quadrature Integration
R=[0 -sqrt(1/3) -sqrt(1/3) sqrt(1/3) sqrt(1/3)];
S=[0 -sqrt(1/3) sqrt(1/3) -sqrt(1/3) sqrt(1/3)];
km=zeros(8,8); kb=zeros(12,12);
135 for i=1:5
    r=R(i); s=S(i);

    %Geometry Shape Functions; x-y Coordinates and Derivatives; det(...
      J)
140 N1=(1/4)*(1-r)*(1-s);
N2=(1/4)*(1+r)*(1-s);
N3=(1/4)*(1+r)*(1+s);
N4=(1/4)*(1-r)*(1+s);
N=[N1 N2 N3 N4];
x=N*X; y=N*Y;
145 xr=(1/4)*[s-1 1-s 1+s -1-s]*X;
xs=(1/4)*[r-1 -1-r 1+r 1-r]*X;
ys=(1/4)*[r-1 -1-r 1+r 1-r]*Y;
yr=(1/4)*[s-1 1-s 1+s -1-s]*Y;
detJ=xr*ys-xs*yr;
150 %Membrane Formulation
N1r=(1/4)*(s-1); N1s=(1/4)*(r-1);
N2r=(1/4)*(1-s); N2s=(1/4)*(-r-1);
N3r=(1/4)*(1+s); N3s=(1/4)*(1+r);
155 N4r=(1/4)*(-1-s); N4s=(1/4)*(1-r);
B1=[ys*N1r-yr*N1s 0;0 xr*N1s-xs*N1r;
    xr*N1s-xs*N1r ys*N1r-yr*N1s];
B2=[ys*N2r-yr*N2s 0;0 xr*N2s-xs*N2r;
    xr*N2s-xs*N2r ys*N2r-yr*N2s];
160 B3=[ys*N3r-yr*N3s 0;0 xr*N3s-xs*N3r;
    xr*N3s-xs*N3r ys*N3r-yr*N3s];
B4=[ys*N4r-yr*N4s 0;0 xr*N4s-xs*N4r;
    xr*N4s-xs*N4r ys*N4r-yr*N4s];
Bm=(1/detJ)*[B1 B2 B3 B4];
165 if i==1
    B0m=Bm;
else
    km=km+Bm'*Dm*Bm*detJ;
end
170 %Bending Formulation
P21r=0; P21s=0;

```

```

P22r=0; P22s=0;
P23r=0; P23s=0;
175 P24r=0; P24s=0;
P25r=xr; P25s=xs;
P26r=2*yr; P26s=2*ys;
P27r=0; P27s=0;
P28r=2*x*xr; P28s=2*x*xs;
180 P29r=2*(xr*y+x*yr); P29s=2*(xs*y+x*ys);
P210r=6*y*yr; P210s=6*y*ys;
P211r=3*x^2*xr; P211s=3*x^2*xs;
P212r=3*(xr*y^2+2*x*y*yr); P212s=3*(xs*y^2+2*x*y*ys);
185 P31r=0; P31s=0;
P32r=0; P32s=0;
P33r=0; P33s=0;
P34r=-2*xr; P34s=-2*xs;
P35r=-yr; P35s=-ys;
P36r=0; P36s=0;
190 P37r=-6*x*xr; P37s=-6*x*xs;
P38r=-2*(xr*y+x*yr); P38s=-2*(xs*y+x*ys);
P39r=-2*y*yr; P39s=-2*y*ys;
P310r=0; P310s=0;
P311r=-3*(2*x*xr*y+x^2*yr); P311s=-3*(2*x*xs*y+x^2*ys);
195 P312r=-3*y^2*yr; P312s=-3*y^2*ys;
Q1=[ys*P31r-yr*P31s ys*P32r-yr*P32s ys*P33r-yr*P33s
-xr*P21s+xs*P21r -xr*P22s+xs*P22s -xr*P23s+xs*P23r
-ys*P21r+yr*P21s+xr*P31s-xs*P31r -ys*P22r+yr*P22s...
+xr*P32s-xs*P32r -ys*P23r+yr*P23s+xr*P33s-xs*P33r];
200 Q2=[ys*P34r-yr*P34s ys*P35r-yr*P35s ys*P36r-yr*P36s
-xr*P24s+xs*P24r -xr*P25s+xs*P25r -xr*P26s+xs*P26r
-ys*P24r+yr*P24s+xr*P34s-xs*P34r -ys*P25r+yr*P25s...
+xr*P35s-xs*P35r -ys*P26r+yr*P26s+xr*P36s-xs*P36r];
205 Q3=[ys*P37r-yr*P37s ys*P38r-yr*P38s ys*P39r-yr*P39s
-xr*P27s+xs*P27r -xr*P28s+xs*P28r -xr*P29s+xs*P29r
-ys*P27r+yr*P27s+xr*P37s-xs*P37r -ys*P28r+yr*P28s...
+xr*P38s-xs*P38r -ys*P29r+yr*P29s+xr*P39s-xs*P39r];
Q4=[ys*P310r-yr*P310s ys*P311r-yr*P311s ys*P312r-yr*P312s
-xr*P210s+xs*P210r -xr*P211s+xs*P211r -xr*P212s+xs*P212r
210 -ys*P210r+yr*P210s+xr*P310s-xs*P310r...
-ys*P211r+yr*P211s+xr*P311s-xs*P311r...
-ys*P212r+yr*P212s+xr*P312s-xs*P312r];
Q=(1/detJ)*[Q1 Q2 Q3 Q4];
Bb=Q/C;
215 if i==1
B0b=Bb;
else
kb=kb+Bb'*Db*Bb*detJ;
end
220 end

%Superposition of Membrane and Bending Stiffness Matrices
k11m=km(1:2,1:2);k12m=km(1:2,3:4);
k13m=km(1:2,5:6);k14m=km(1:2,7:8);

```

```

225 k21m=km(3:4,1:2);k22m=km(3:4,3:4);
    k23m=km(3:4,5:6);k24m=km(3:4,7:8);
    k31m=km(5:6,1:2);k32m=km(5:6,3:4);
    k33m=km(5:6,5:6);k34m=km(5:6,7:8);
    k41m=km(7:8,1:2);k42m=km(7:8,3:4);
230 k43m=km(7:8,5:6);k44m=km(7:8,7:8);
    k11b=kb(1:3,1:3);k12b=kb(1:3,4:6);
    k13b=kb(1:3,7:9);k14b=kb(1:3,10:12);
    k21b=kb(4:6,1:3);k22b=kb(4:6,4:6);
    k23b=kb(4:6,7:9);k24b=kb(4:6,10:12);
235 k31b=kb(7:9,1:3);k32b=kb(7:9,4:6);
    k33b=kb(7:9,7:9);k34b=kb(7:9,10:12);
    k41b=kb(10:12,1:3);k42b=kb(10:12,4:6);
    k43b=kb(10:12,7:9);k44b=kb(10:12,10:12);
    A=areaP(X,Y);k=0.3*E*th*A;k5=-0.5*k;
240 z21=zeros(2,1);z23=zeros(2,3);z31=zeros(3,1);
    KE=[k11m z23 z21 k12m z23 z21 k13m z23 z21 k14m z23 z21
        z23' k11b z31 z23' k12b z31 z23' k13b z31 z23' k14b z31
        z21' z31' k z21' z31' k5 z21' z31' k5 z21' z31' k5
        k21m z23 z21 k22m z23 z21 k23m z23 z21 k24m z23 z21
245 z23' k21b z31 z23' k22b z31 z23' k23b z31 z23' k24b z31
        z21' z31' k5 z21' z31' k z21' z31' k5 z21' z31' k5
        k31m z23 z21 k32m z23 z21 k33m z23 z21 k34m z23 z21
        z23' k31b z31 z23' k32b z31 z23' k33b z31 z23' k34b z31
        z21' z31' k5 z21' z31' k5 z21' z31' k z21' z31' k5
250 k41m z23 z21 k42m z23 z21 k43m z23 z21 k44m z23 z21
        z23' k41b z31 z23' k42b z31 z23' k43b z31 z23' k44b z31
        z21' z31' k5 z21' z31' k5 z21' z31' k5 z21' z31' k ];

end
255

%% AREA OF A POLYGON
function [A]=areaP(x,y)

260 x(end+1)=x(1);y(end+1)=y(1);sum=0;
    for i=1:length(x)-1
        sum=sum+det([x(i) x(i+1);y(i) y(i+1)]);
    end
    A=0.5*sum;
265
end

```

Listing B.4: 3D Membrane-Bending Finite Element Model

```

%% FINITE ELEMENT MEMBRANE & BENDING ANALYSIS FOR 3-D GEOMETRIES
function FEmb3D(params)
3
    %Setup Parameters
    params=params{1};
    mh=params.mesh;
    [~,n]=size(params.n);
8 nelx=0;

```

```

for i=1:n
    nelx=nelx+params.n{1}-1;
end
nely=params.m{1}-1;
13 nel=nelx*nely;

%Elemental Stiffness Matrices
edof=zeros(24,nel);KE=zeros(24*nel,24);T=zeros(24*nel,24);
X=zeros(4,nel);Y=zeros(4,nel);Z=zeros(4,nel);
18 for el=1:nel
    m=floor((el-1)/nely);
    n=rem(el-1,nely)+1;
    e=[6*m*(nely+1)+6*n+1 6*(m+1)*(nely+1)+6*n+1 ...
        6*(m+1)*(nely+1)+6*n-5 6*m*(nely+1)+6*n-5];
23 edof(:,el) = [e(1);e(1)+1;e(1)+2;e(1)+3;e(1)+4;e(1)+5;
                e(2);e(2)+1;e(2)+2;e(2)+3;e(2)+4;e(2)+5;
                e(3);e(3)+1;e(3)+2;e(3)+3;e(3)+4;e(3)+5;
                e(4);e(4)+1;e(4)+2;e(4)+3;e(4)+4;e(4)+5];
    mhi=[m*(nely+1)+n+1;(m+1)*(nely+1)+n+1;
28         (m+1)*(nely+1)+n;m*(nely+1)+n];
    X(:,el)=[mh(mhi(1),1);mh(mhi(2),1);mh(mhi(3),1);mh(mhi(4),1)];
    Y(:,el)=[mh(mhi(1),2);mh(mhi(2),2);mh(mhi(3),2);mh(mhi(4),2)];
    Z(:,el)=[mh(mhi(1),3);mh(mhi(2),3);mh(mhi(3),3);mh(mhi(4),3)];
    [KE(24*(el-1)+1:24*el,1:24),T(24*(el-1)+1:24*el,1:24),...
33     B0m(3*(el-1)+1:3*el,1:8),B0b(3*(el-1)+1:3*el,1:12),...
     Dm,Db]...
     =Ke(X(:,el),Y(:,el),Z(:,el));

end

38 %Calculate Global Displacements
[U]=FE(nelx,nely,edof,KE);

%Visualize Results
u=U(1:6:6*(nelx+1)*(nely+1),1);
43 v=U(2:6:6*(nelx+1)*(nely+1),1);
w=U(3:6:6*(nelx+1)*(nely+1),1);
Xa=mh(:,1)+u; Ya=mh(:,2)+v; Za=mh(:,3)+w;
color=zeros(nel,1);
for el=1:nel
48     m=floor((el-1)/nely);
        n=rem(el-1,nely)+1;
        mhi=[m*(nely+1)+n+1;(m+1)*(nely+1)+n+1;
              (m+1)*(nely+1)+n;m*(nely+1)+n];
        X(:,el)=[Xa(mhi(1),1);Xa(mhi(2),1);Xa(mhi(3),1);Xa(mhi(4),1)];
53     Y(:,el)=[Ya(mhi(1),1);Ya(mhi(2),1);Ya(mhi(3),1);Ya(mhi(4),1)];
        Z(:,el)=[Za(mhi(1),1);Za(mhi(2),1);Za(mhi(3),1);Za(mhi(4),1)];
        d=U(edof(:,el));
        dm=d([1:2 7:8 13:14 19:20]);
        db=d([3:5 9:11 15:17 21:23]);
58     sigm=(Dm*B0m(3*(el-1)+1:3*el,1:8)*dm);
        sigb=(Db*B0b(3*(el-1)+1:3*el,1:12)*db);
        sig=sigm+sigb;

```



```

        prin1=(sig(1)+sig(2))/2+sqrt(((sig(1)-sig(2))/2)^2+sig(3)^2);
        prin2=(sig(1)+sig(2))/2-sqrt(((sig(1)-sig(2))/2)^2+sig(3)^2);
63      sig(4)=sqrt(((prin1-prin2)^2+prin1^2+prin2^2)/2);
        color(1:4,e1)=sig;
    end
    figure;patch(X,Y,Z,color(1,:));
    title('\sigma_x');axis equal;grid;colorbar
68  figure;patch(X,Y,Z,color(2,:));
    title('\sigma_y');axis equal;grid;colorbar
    figure;patch(X,Y,Z,color(3,:));
    title('\tau_{xy}');axis equal;grid;colorbar
    figure;patch(X,Y,Z,color(4,:));
73  title('\sigma_{vm}');axis equal;grid;colorbar

    end

78  %% SOLVE DISPLACEMENTS
    function [U]=FE(nelx,nely,edof,KE)

    %Compile Stiffness Matrix
    K=zeros(6*(nelx+1)*(nely+1),6*(nelx+1)*(nely+1));
83  F=zeros(6*(nely+1)*(nelx+1),1);U=zeros(6*(nely+1)*(nelx+1),1);
    for e1=1:nelx*nely
        K(edof(:,e1),edof(:,e1))=...
            K(edof(:,e1),edof(:,e1)) + KE(24*(e1-1)+1:24*e1,1:24);
    end

88  % Define Loads and Supports
    F(1000,1)=-1000; %INPUT: Apply external forces
    fixeddofs=1:100; %INPUT: Constrain degrees of freedom
    alldofs=1:6*(nely+1)*(nelx+1);
93  freedofs=setdiff(alldofs,fixeddofs);
    U(freedofs,:)=K(freedofs,freedofs)\F(freedofs,:);
    U(fixeddofs,:)=0;

    end

98  %% ELEMENT STIFFNESS MATRIX
    function [KE,T,B0m,B0b,Dm,Dbz]=Ke(X,Y,Z)

103 %Constitutive Matrix
    E=200e9; %INPUT: Young's modulus
    nu=0.3; %INPUT: Poisson's ratio
    th=0.1; %INPUT: element thickness
    z=th/2;
108 D=[1 nu 0;nu 1 0;0 0 (1-nu)/2];
    Dm=(E*th/(1-nu^2))*D;
    Db=E*th^3/(12*(1-nu^2))*D;
    Dbz=z*E/(1-nu^2)*D;

```

```

113 %Local Shape and Transformation
    [T,xf,yf]=trans(X,Y,Z);

    %Formulation of Bending [C] Matrix
    x1=X(1); x2=X(2); x3=X(3); x4=X(4);
118 y1=Y(1); y2=Y(2); y3=Y(3); y4=Y(4);
    C=[1 x1 y1 x1^2 x1*y1 y1^2 x1^3 x1^2*y1 x1*y1^2 y1^3 x1^3*y1 x1*y1...
        ^3
        0 0 1 0 x1 2*y1 0 x1^2 2*x1*y1 3*y1^2 x1^3 3*x1*y1^2
        0 -1 0 -2*x1 -y1 0 -3*x1^2 -2*x1*y1 -y1^2 0 -3*x1^2*y1 -y1^3
        1 x2 y2 x2^2 x2*y2 y2^2 x2^3 x2^2*y2 x2*y2^2 y2^3 x2^3*y2 x2*y2...
        ^3
123 0 0 1 0 x2 2*y2 0 x2^2 2*x2*y2 3*y2^2 x2^3 3*x2*y2^2
        0 -1 0 -2*x2 -y2 0 -3*x2^2 -2*x2*y2 -y2^2 0 -3*x2^2*y2 -y2^3
        1 x3 y3 x3^2 x3*y3 y3^2 x3^3 x3^2*y3 x3*y3^2 y3^3 x3^3*y3 x3*y3...
        ^3
        0 0 1 0 x3 2*y3 0 x3^2 2*x3*y3 3*y3^2 x3^3 3*x3*y3^2
        0 -1 0 -2*x3 -y3 0 -3*x3^2 -2*x3*y3 -y3^2 0 -3*x3^2*y3 -y3^3
128 1 x4 y4 x4^2 x4*y4 y4^2 x4^3 x4^2*y4 x4*y4^2 y4^3 x4^3*y4 x4*y4...
        ^3
        0 0 1 0 x4 2*y4 0 x4^2 2*x4*y4 3*y4^2 x4^3 3*x4*y4^2
        0 -1 0 -2*x4 -y4 0 -3*x4^2 -2*x4*y4 -y4^2 0 -3*x4^2*y4 -y4^3];

    %Membrane & Bending Gaussian Quadrature Integration
133 R=[0 -sqrt(1/3) -sqrt(1/3) sqrt(1/3) sqrt(1/3)];
    S=[0 -sqrt(1/3) sqrt(1/3) -sqrt(1/3) sqrt(1/3)];
    km=zeros(8,8); kb=zeros(12,12);
    for i=1:5
        r=R(i); s=S(i);
138 %Bilinear Shape Functions; x-y Coordinates and Derivatives; det(...)
        J)
        N1=(1/4)*(1-r)*(1-s);
        N2=(1/4)*(1+r)*(1-s);
        N3=(1/4)*(1+r)*(1+s);
143 N4=(1/4)*(1-r)*(1+s);
        N=[N1 N2 N3 N4];
        x=N*X; y=N*Y;
        xr=(1/4)*[s-1 1-s 1+s -1-s]*X;
        xs=(1/4)*[r-1 -1-r 1+r 1-r]*X;
148 ys=(1/4)*[r-1 -1-r 1+r 1-r]*Y;
        yr=(1/4)*[s-1 1-s 1+s -1-s]*Y;
        detJ=xr*ys-xs*yr;

    %Membrane Formulation
153 N1r=(1/4)*(s-1); N1s=(1/4)*(r-1);
        N2r=(1/4)*(1-s); N2s=(1/4)*(-r-1);
        N3r=(1/4)*(1+s); N3s=(1/4)*(1+r);
        N4r=(1/4)*(-1-s); N4s=(1/4)*(1-r);
        B1=[ys*N1r-yr*N1s 0;0 xr*N1s-xs*N1r;
158 xr*N1s-xs*N1r ys*N1r-yr*N1s];
        B2=[ys*N2r-yr*N2s 0;0 xr*N2s-xs*N2r;

```

```

        xr*N2s-xs*N2r ys*N2r-yr*N2s];
B3=[ys*N3r-yr*N3s 0;0 xr*N3s-xs*N3r;
    xr*N3s-xs*N3r ys*N3r-yr*N3s];
163 B4=[ys*N4r-yr*N4s 0;0 xr*N4s-xs*N4r;
    xr*N4s-xs*N4r ys*N4r-yr*N4s];
Bm=(1/detJ)*[B1 B2 B3 B4];
if i==1
    B0m=Bm;
168 else
    km=km+Bm'*Dm*Bm*detJ;
end

%Bending Formulation
173 P21r=0; P21s=0;
    P22r=0; P22s=0;
    P23r=0; P23s=0;
    P24r=0; P24s=0;
    P25r=xr; P25s=xs;
178 P26r=2*yr; P26s=2*ys;
    P27r=0; P27s=0;
    P28r=2*x*xr; P28s=2*x*xs;
    P29r=2*(xr*y+x*yr); P29s=2*(xs*y+x*ys);
    P210r=6*y*yr; P210s=6*y*ys;
183 P211r=3*x^2*xr; P211s=3*x^2*xs;
    P212r=3*(xr*y^2+2*x*y*yr); P212s=3*(xs*y^2+2*x*y*ys);
    P31r=0; P31s=0;
    P32r=0; P32s=0;
    P33r=0; P33s=0;
188 P34r=-2*xr; P34s=-2*xs;
    P35r=-yr; P35s=-ys;
    P36r=0; P36s=0;
    P37r=-6*x*xr; P37s=-6*x*xs;
    P38r=-2*(xr*y+x*yr); P38s=-2*(xs*y+x*ys);
193 P39r=-2*y*yr; P39s=-2*y*ys;
    P310r=0; P310s=0;
    P311r=-3*(2*x*xr*y+x^2*yr); P311s=-3*(2*x*xs*y+x^2*ys);
    P312r=-3*y^2*yr; P312s=-3*y^2*ys;
198 Q1=[ys*P31r-yr*P31s ys*P32r-yr*P32s ys*P33r-yr*P33s
    -xr*P21s+xs*P21r -xr*P22s+xs*P22s -xr*P23s+xs*P23r
    -ys*P21r+yr*P21s+xr*P31s -xs*P31r -ys*P22r+yr*P22s...
    +xr*P32s-xs*P32r -ys*P23r+yr*P23s+xr*P33s-xs*P33r];
Q2=[ys*P34r-yr*P34s ys*P35r-yr*P35s ys*P36r-yr*P36s
    -xr*P24s+xs*P24r -xr*P25s+xs*P25r -xr*P26s+xs*P26r
203 -ys*P24r+yr*P24s+xr*P34s -xs*P34r -ys*P25r+yr*P25s...
    +xr*P35s-xs*P35r -ys*P26r+yr*P26s+xr*P36s-xs*P36r];
Q3=[ys*P37r-yr*P37s ys*P38r-yr*P38s ys*P39r-yr*P39s
    -xr*P27s+xs*P27r -xr*P28s+xs*P28r -xr*P29s+xs*P29r
    -ys*P27r+yr*P27s+xr*P37s -xs*P37r -ys*P28r+yr*P28s...
208 +xr*P38s-xs*P38r -ys*P29r+yr*P29s+xr*P39s-xs*P39r];
Q4=[ys*P310r-yr*P310s ys*P311r-yr*P311s ys*P312r-yr*P312s
    -xr*P210s+xs*P210r -xr*P211s+xs*P211r -xr*P212s+xs*P212r
    -ys*P210r+yr*P210s+xr*P310s -xs*P310r...

```

```

        -ys*P211r+yr*P211s+xr*P311s-xs*P311r...
213     -ys*P212r+yr*P212s+xr*P312s-xs*P312r];
    Q=(1/detJ)*[Q1 Q2 Q3 Q4];
    Bb=Q/C;
    if i==1
        B0b=Bb;
218     else
        kb=kb+Bb'*Db*Bb*detJ;
    end
end

223 %Superposition of Membrane and Bending Stiffness Matrices
k11m=km(1:2,1:2);k12m=km(1:2,3:4);
k13m=km(1:2,5:6);k14m=km(1:2,7:8);
k21m=km(3:4,1:2);k22m=km(3:4,3:4);
k23m=km(3:4,5:6);k24m=km(3:4,7:8);
228 k31m=km(5:6,1:2);k32m=km(5:6,3:4);
k33m=km(5:6,5:6);k34m=km(5:6,7:8);
k41m=km(7:8,1:2);k42m=km(7:8,3:4);
k43m=km(7:8,5:6);k44m=km(7:8,7:8);
k11b=kb(1:3,1:3);k12b=kb(1:3,4:6);
233 k13b=kb(1:3,7:9);k14b=kb(1:3,10:12);
k21b=kb(4:6,1:3);k22b=kb(4:6,4:6);
k23b=kb(4:6,7:9);k24b=kb(4:6,10:12);
k31b=kb(7:9,1:3);k32b=kb(7:9,4:6);
k33b=kb(7:9,7:9);k34b=kb(7:9,10:12);
238 k41b=kb(10:12,1:3);k42b=kb(10:12,4:6);
k43b=kb(10:12,7:9);k44b=kb(10:12,10:12);
A=areaP(xf,yf);k=0.3*E*th*A;k5=-0.5*k;
z21=zeros(2,1); z23=zeros(2,3);z31=zeros(3,1);
Ke=[k11m z23 z21 k12m z23 z21 k13m z23 z21 k14m z23 z21
243 z23' k11b z31 z23' k12b z31 z23' k13b z31 z23' k14b z31
z21' z31' k z21' z31' k5 z21' z31' k5 z21' z31' k5
k21m z23 z21 k22m z23 z21 k23m z23 z21 k24m z23 z21
z23' k21b z31 z23' k22b z31 z23' k23b z31 z23' k24b z31
z21' z31' k5 z21' z31' k z21' z31' k5 z21' z31' k5
248 k31m z23 z21 k32m z23 z21 k33m z23 z21 k34m z23 z21
z23' k31b z31 z23' k32b z31 z23' k33b z31 z23' k34b z31
z21' z31' k5 z21' z31' k5 z21' z31' k z21' z31' k5
k41m z23 z21 k42m z23 z21 k43m z23 z21 k44m z23 z21
z23' k41b z31 z23' k42b z31 z23' k43b z31 z23' k44b z31
253 z21' z31' k5 z21' z31' k5 z21' z31' k5 z21' z31' k ];
KE=(Ke/T)*T;

end

258
%% TRANSFORMATION MATRIX
function [T,xf,yf]=trans(X,Y,Z)
p1=[X(1) Y(1) Z(1)];p2=[X(2) Y(2) Z(2)];
p3=[X(3) Y(3) Z(3)];p4=[X(4) Y(4) Z(4)];
263 v21=p2-p1; v32=p3-p2; v41=p4-p1;

```

```

d21=norm(v21,2); d32=norm(v32,2); d41=norm(v41,2);
ang1=acos(dot(v21,v32)/(d21*d32));
ang2=acos(dot(v21,v41)/(d21*d41));
xf=[0 d21 d21+d32*cos(ang1) d41*cos(ang2)];
268 yf=[0 0 d32*sin(ang1) d41*sin(ang2)];
et=cross(v21,v41)/norm(cross(v21,v41),2);
nx=et(1); ny=et(2); nz=et(3);
r=[0 1 0]; s=[0 0 1];
N1=(1/4)*(1-r).*(1-s);
273 N2=(1/4)*(1+r).*(1-s);
N3=(1/4)*(1+r).*(1+s);
N4=(1/4)*(1-r).*(1+s);
N=[N1' N2' N3' N4'];
x=N*X; y=N*Y;
278 z=Z(1)-(1/nz)*(nx*(x-X(1))+ny*(y-Y(1)));
r=[x(2)-x(1) y(2)-y(1) z(2)-z(1)];
s=[x(3)-x(1) y(3)-y(1) z(3)-z(1)];
er=r/norm(r,2); es=s/norm(s,2);
ex=[1 0 0]; ey=[0 1 0]; ez=[0 0 1];
283 l1=cosV(ex,er); l2=cosV(ex,es); l3=cosV(ex,et);
m1=cosV(ey,er); m2=cosV(ey,es); m3=cosV(ey,et);
n1=cosV(ez,er); n2=cosV(ez,es); n3=cosV(ez,et);
Tn=[l1 l2 l3 0 0 0
     m1 m2 m3 0 0 0
     n1 n2 n3 0 0 0
     0 0 0 l1 l2 l3
     0 0 0 m1 m2 m3
     0 0 0 n1 n2 n3];
T=blkdiag(Tn,Tn,Tn,Tn);
293
end

%% COSINE BETWEEN VECTORS
298 function [ang]=cosV(e1,e2)
ang=dot(e1,e2)/(norm(e1,2)*norm(e2,2));

end

303
%% AREA OF A POLYGON
function [A]=areaP(x,y)

x(end+1)=x(1);y(end+1)=y(1);sum=0;
308 for i=1:length(x)-1
sum=sum+det([x(i) x(i+1);y(i) y(i+1)]);
end
A=0.5*sum;

313 end

```

## Appendix C. SIMP MATLAB Codes

Listing C.1: Membrane SIMP Model

```
1 %% SIMP METHOD FOR MEMBRANE ELEMENT
function [el,nel,X,Y,x]=SIMPm(params,tor,varargin)

%Setup Parameters
if isempty(varargin)==1
6   config=1;
elseif isempty(varargin)==0
   config=varargin{1};
end
params=params{config};
11 if isfield(params,'points')==0
    mh=params.mesh;
elseif isfield(params,'points')==1
    mh=params.points;
end
16 nelxp=params.n;
[~,n]=size(nelxp);
nelx=zeros(n);
for i=1:n
    nelx(i)=nelxp{i}-1;
21 end
nelxp=nelx;
nelx=sum(nelx);
nely=params.m{1}-1;
nel=nelx*nely;
26 %Input Parameters
volfrac=0.5; %INPUT: Volume fraction
penal=3; %INPUT: Penalty
rmin=1.5; %INPUT: Radius of filter
31 conv=0.01; %INPUT: Change convergence
xmin=0.001; %INPUT: Minimum element thickness
fdofs=union(1:2:2*(nely+1),2); %INPUT: Constrain DOFs
[F]=Faero(tor,nelx,nelxp,nely,2,1);
disp('Aerodynamic Forces Computed')
36 %Elemental Stiffness
edof=zeros(8,nel);KE=zeros(8*nel,8);
X=zeros(4,nel);Y=zeros(4,nel);
A=zeros(nel,1);
41 for el=1:nel
    m=floor((el-1)/nely);
    n=rem(el-1,nely)+1;
    edof(:,el)=[2*m*(nely+1)+2*n+1;2*m*(nely+1)+2*n+2;
        2*(m+1)*(nely+1)+2*n+1;2*(m+1)*(nely+1)+2*n+2;
46    2*(m+1)*(nely+1)+2*n-1;2*(m+1)*(nely+1)+2*n;
        2*m*(nely+1)+2*n-1;2*m*(nely+1)+2*n];
    mhi=[m*(nely+1)+n+1;(m+1)*(nely+1)+n+1;
        (m+1)*(nely+1)+n;m*(nely+1)+n];
    X(:,el)=[mh(mhi(1),1);mh(mhi(2),1);mh(mhi(3),1);mh(mhi(4),1)];
```

```

51     Y(:,el)=[mh(mhi(1),2);mh(mhi(2),2);mh(mhi(3),2);mh(mhi(4),2)];
        A(el)=areaP(X(:,el),Y(:,el));
        [KE(8*(el-1)+1:8*el,1:8)]=Ke(X(:,el),Y(:,el));
    end
    disp('Elemental Stiffness Matrices Computed')
56     %Start Iteration
        vol=volfrac*sum(A);
        x(1:nel,1)=volfrac;
        loop=0;change=1;
61     while change>conv
            loop=loop+1;
            xold=x;

            %Calculate Global Displacements
66     [U]=FE(nelx,nely,edof,KE,x,penal,F,fdofs);

            %Objective Function and Sensitivity Analysis
            c=0;dc=zeros(nel,1);
            for el=1:nel
71         Ue=U(edof(:,el),1);
            c=c+x(el)^penal*Ue'*KE(8*(el-1)+1:8*el,1:8)'*Ue;
            dc(el)=-penal*x(el)^(penal-1)*Ue'*KE(8*(el-1)+1:8*el,1:8)'.
                Ue;
            end

76     %Filtering of Sensitivities
            [dc]=check(nelx,nely,rmin,x,dc);

            %Design Update by the Optimality Criteria
            [x]=OC(x,xmin,vol,dc,A);
81

            %Print Results
            change=max(abs(x-xold));
            disp([' It.: ' sprintf('%4i',loop) ...
                ' Obj.: ' sprintf('%10.4f',c) ...
86         ' Vol.: ' sprintf('%6.3f',(x'*A)/sum(A)) ...
                ' ch.: ' sprintf('%6.3f',change )])
            end

            %Plot Densities
91 h=figure;set(h,'color',[1 1 1]);colormap(bone);patch(X,Y,-x');
            axis equal;axis tight;axis off

            end

96     %% OPTIMALITY CRITERIA UPDATE
        function [xnew]=OC(x,xmin,vol,dc,A)
            l1=0;l2=100000;move=0.2;eta=0.5;
            while (l2-l1>1e-4)
101         lmid=0.5*(l2+l1);

```

```

        xnew=max(xmin,max(x-move,min(1.,min(x+move,x.*(-dc./(lmid*A)).^...
            eta)))));
        if xnew'*A-vol>0;
            l1=lmid;
        else
106     l2=lmid;
        end
    end
end

111

%% MESH-INDEPENDENCY FILTER
function [dcn]=check(nelx,nely,rmin,x,dc)
xmap=reshape(x,nely,nelx);
116 dcmmap=reshape(dc,nely,nelx);
dcnmap=zeros(nely,nelx);
for i=1:nelx
    for j=1:nely
        sum=0.0;
121     for k=max(i-floor(rmin),1):min(i+floor(rmin),nelx);
            for l=max(j-floor(rmin),1):min(j+floor(rmin),nely)
                fac=rmin-sqrt((i-k)^2+(j-l)^2);
                sum=sum+max(0,fac);
                dcnmap(j,i)=dcnmap(j,i)+max(0,fac)*xmap(l,k)*dcmmap(l,k);
126     end
            end
            dcnmap(j,i)=dcnmap(j,i)/(xmap(j,i)*sum);
        end
    end
end
131 dcn=reshape(dcnmap,nelx*nely,1);

end

136 %% FE-ANALYSIS
function [U]=FE(nelx,nely,edof,KE,x,penal,F,fdofs)

%%Compile Global Stiffness Matrix
K=zeros(2*(nelx+1)*(nely+1),2*(nelx+1)*(nely+1));
141 U=zeros(2*(nely+1)*(nelx+1),1);
for el=1:nelx*nely
    K(edof(:,el),edof(:,el))=...
        K(edof(:,el),edof(:,el))+x(el)^penal*KE(8*(el-1)+1:8*el,1:8)...
        ;
end
146
%%Calculate Displacement
fixeddofs=fdofs;
alldofs=1:2*(nely+1)*(nelx+1);
freedofs=setdiff(alldofs,fixeddofs);
151 F(fixeddofs,:)=0;

```



```

U(freedofs,:) = K(freedofs, freedofs) \ F(freedofs, :);
U(fixeddofs, :) = 0;

end
156

%% ELEMENT STIFFNESS MATRIX
function [KE, B0, D] = Ke(X, Y)

161 %Constitutive Matrix
E = 200e9; %INPUT: Young's modulus
nu = 0.3; %INPUT: Poisson's ratio
th = 1; %INPUT: thickness of element
D = E * th / (1 - nu^2) * [1 nu 0; nu 1 0; 0 0 (1 - nu) / 2];

166 %Gaussian Quadrature Integration
R = [0 -sqrt(1/3) -sqrt(1/3) sqrt(1/3) sqrt(1/3)];
S = [0 -sqrt(1/3) sqrt(1/3) -sqrt(1/3) sqrt(1/3)];
k = zeros(8, 8);

171 for i = 1:length(R)
    r = R(i); s = S(i);
    xr = (1/4) * [-1+s 1-s 1+s -1-s] * X;
    xs = (1/4) * [-1+r -1-r 1+r 1-r] * X;
    yr = (1/4) * [-1+s 1-s 1+s -1-s] * Y;
176 ys = (1/4) * [-1+r -1-r 1+r 1-r] * Y;
    detJ = xr * ys - xs * yr;
    N1r = (1/4) * (s - 1); N1s = (1/4) * (r - 1);
    N2r = (1/4) * (1 - s); N2s = (1/4) * (-r - 1);
    N3r = (1/4) * (1 + s); N3s = (1/4) * (1 + r);
181 N4r = (1/4) * (-1 - s); N4s = (1/4) * (1 - r);
    B1 = [ys * N1r - yr * N1s 0; 0 xr * N1s - xs * N1r;
          xr * N1s - xs * N1r ys * N1r - yr * N1s];
    B2 = [ys * N2r - yr * N2s 0; 0 xr * N2s - xs * N2r;
          xr * N2s - xs * N2r ys * N2r - yr * N2s];
186 B3 = [ys * N3r - yr * N3s 0; 0 xr * N3s - xs * N3r;
          xr * N3s - xs * N3r ys * N3r - yr * N3s];
    B4 = [ys * N4r - yr * N4s 0; 0 xr * N4s - xs * N4r;
          xr * N4s - xs * N4r ys * N4r - yr * N4s];
    B = (1/detJ) * [B1 B2 B3 B4];

191 if i == 1
        B0 = B;
    else
        k = k + B' * D * B * detJ;
    end
end
196 end
KE = k;
end

```

Listing C.2: Bending SIMP Model

```

%% SIMP METHOD: Bending
2 function [el, nel, X, Y, x] = SIMPb(params, tor, varargin)

```

```

%Setup Parameters
if isempty(varargin)==1
    config=1;
7 elseif isempty(varargin)==0
    config=varargin{1};
end
params=params{config};
if isfield(params,'points')==0
12     mh=params.mesh;
elseif isfield(params,'points')==1
    mh=params.points;
end
nelxp=params.n;
17 [~,n]=size(nelxp);
nelx=zeros(n);
for i=1:n
    nelx(i)=nelxp{i}-1;
end
22 nelx=sum(nelx);
nely=params.m{1}-1;
nel=nelx*nely;

%Input Parameters
27 volfrac=0.5; %INPUT: Volume fraction
penal=3; %INPUT: Penalty
rmin=1.5; %INPUT: Radius of filter
conv=0.01; %INPUT: Change convergence
xmin=0.001; %INPUT: Minimum element thickness
32 fdofs=1:3*(nely+1); %INPUT: Constrain DOFs
[F]=Faero(tor,nelx,nelxp,nely,3,1);
disp('Aerodynamic Forces Computed')

%Elemental Stiffness Matrices
37 edof=zeros(12,nel);KE=zeros(12*nel,12);
X=zeros(4,nel);Y=zeros(4,nel);
A=zeros(nel,1);
for el=1:nel
    m=floor((el-1)/nely);
42     n=rem(el-1,nely)+1;
    edofi=[3*m*(nely+1)+3*n+1 3*(m+1)*(nely+1)+3*n+1 ...
           3*(m+1)*(nely+1)+3*n-2 3*m*(nely+1)+3*n-2];
    edof(:,el)=[edofi(1);edofi(1)+1;edofi(1)+2;
                edofi(2);edofi(2)+1;edofi(2)+2;
47                 edofi(3);edofi(3)+1;edofi(3)+2;
                edofi(4);edofi(4)+1;edofi(4)+2];
    mhi=[m*(nely+1)+n+1;(m+1)*(nely+1)+n+1;
          (m+1)*(nely+1)+n;m*(nely+1)+n];
    X(:,el)=[mh(mhi(1),1);mh(mhi(2),1);mh(mhi(3),1);mh(mhi(4),1)];
52     Y(:,el)=[mh(mhi(1),2);mh(mhi(2),2);mh(mhi(3),2);mh(mhi(4),2)];
    A(el)=areaP(X(:,el),Y(:,el));
    [KE(12*(el-1)+1:12*el,1:12)]=Ke(X(:,el),Y(:,el));
end

```

```

disp('Elemental Stiffness Matrices Computed')
57
%Start Iteration
vol=volfrac*sum(A);
x(1:nel,1)=volfrac;
loop=0;change=1;
62 while change>conv
    loop=loop+1;
    xold=x;

    %Calculate Global Displacements
67 [U]=FE(nelx,nely,edof,KE,x,penal,F,fdofs);

    %Objective Function and Sensitivity Analysis
c=0;dc=zeros(nel,1);
for el=1:nel
72 Ue=U(edof(:,el),1);
    c=c+x(el)^(penal)*Ue'*KE(12*(el-1)+1:12*el,1:12)*Ue;
    dc(el)=-penal*x(el)^(penal-1)*Ue'*KE(12*(el-1)+1:12*el...
        ,1:12)*Ue;
end

77 %Filtering of Sensitivities
[dc]=check(nelx,nely,rmin,x,dc);

%Design Update by the Optimality Criteria
[x]=OC(x,xmin,vol,dc,A);
82

%Print Results
change=max(abs(x-xold));
disp([' It.: ' sprintf('%4i',loop) ...
    ' Obj.: ' sprintf('%10.6f',c) ...
87    ' Vol.: ' sprintf('%6.3f',(x'*A)/sum(A)) ...
    ' ch.: ' sprintf('%6.3f',change) ])
end

%Plot Densities
92 h=figure;set(h,'color',[1 1 1]);colormap(bone);patch(X,Y,-x');
axis equal;axis tight;axis off

end

97
%% OPTIMALITY CRITERIA UPDATE
function [xnew]=OC(x,xmin,vol,dc,A)
l1=0;l2=100000;move=.2;eta=0.3;
while (l2-l1>1e-4)
102 lmid=0.5*(l2+l1);
    xnew=max(xmin,max(x-move,min(1.,min(x+move,x*(-dc./(lmid*A)).^...
        eta)))));
    if xnew'*A-vol>0;
        l1=lmid;

```

```

    else
107     l2=lmid;
        end
    end

    end
112

%% MESH-INDEPENDENCY FILTER
function [dcn]=check(nelx,nely,rmin,x,dc)
xmap=reshape(x,nely,nelx);
117 dcnmap=reshape(dc,nely,nelx);
    dcnmap=zeros(nely,nelx);
    for i=1:nelx
        for j=1:nely
            sum=0.0;
122     for k=max(i-floor(rmin),1):min(i+floor(rmin),nelx);
                for l=max(j-floor(rmin),1):min(j+floor(rmin),nely)
                    fac=rmin-sqrt((i-k)^2+(j-l)^2);
                    sum=sum+max(0,fac);
                    dcnmap(j,i)=dcnmap(j,i)+max(0,fac)*xmap(l,k)*dcmap(l,k);
127     end
                end
            dcnmap(j,i)=dcnmap(j,i)/(xmap(j,i)*sum);
        end
    end
132 dcn=reshape(dcnmap,nelx*nely,1);

    end

137 %% FE-ANALYSIS
function [U]=FE(nelx,nely,edof,KE,x,penal,F,fdofs)

%%Compile Global Stiffness Matrix
K=zeros(3*(nelx+1)*(nely+1),3*(nelx+1)*(nely+1));
142 U=zeros(3*(nely+1)*(nelx+1),1);
    for el=1:nelx*nely
        K(edof(:,el),edof(:,el))=...
            K(edof(:,el),edof(:,el))+x(el)^penal*KE(12*(el-1)+1:12*el...
                ,1:12);
    end
147

%%Define Loads and Supports
fixeddofs=fdofs;
alldofs=1:3*(nely+1)*(nelx+1);
freedofs=setdiff(alldofs,fixeddofs);
152 F(fixeddofs,:)=0;
    U(freedofs,:)=K(freedofs,freedofs)\F(freedofs,:);
    U(fixeddofs,:)=0;

    end

```

157

```
% ELEMENT STIFFNESS MATRIX
function [KE]=Ke(X,Y)
```

162 %Constitutive Matrix

```
E=200e9; %INPUT: Young's modulus
nu=0.3; %INPUT: Poisson's ratio
th=1; %INPUT: thickness of element
D=(E*th^3/(12*(1-nu^2)))*[1 nu 0;nu 1 0;0 0 (1-nu)/2];
```

167

```
%Gaussian Quadrature Integration
```

```
x1=X(1); x2=X(2); x3=X(3); x4=X(4);
y1=Y(1); y2=Y(2); y3=Y(3); y4=Y(4);
C=[1 x1 y1 x1^2 x1*y1 y1^2 x1^3 x1^2*y1 x1*y1^2 y1^3 x1^3*y1 x1*y1^3...
```

172

```
0 0 1 0 x1 2*y1 0 x1^2 2*x1*y1 3*y1^2 x1^3 3*x1*y1^2
0 -1 0 -2*x1 -y1 0 -3*x1^2 -2*x1*y1 -y1^2 0 -3*x1^2*y1 -y1^3
1 x2 y2 x2^2 x2*y2 y2^2 x2^3 x2^2*y2 x2*y2^2 y2^3 x2^3*y2 x2*y2^3...
```

177

```
0 0 1 0 x2 2*y2 0 x2^2 2*x2*y2 3*y2^2 x2^3 3*x2*y2^2
0 -1 0 -2*x2 -y2 0 -3*x2^2 -2*x2*y2 -y2^2 0 -3*x2^2*y2 -y2^3
1 x3 y3 x3^2 x3*y3 y3^2 x3^3 x3^2*y3 x3*y3^2 y3^3 x3^3*y3 x3*y3^3...
```

```
0 0 1 0 x3 2*y3 0 x3^2 2*x3*y3 3*y3^2 x3^3 3*x3*y3^2
0 -1 0 -2*x3 -y3 0 -3*x3^2 -2*x3*y3 -y3^2 0 -3*x3^2*y3 -y3^3
1 x4 y4 x4^2 x4*y4 y4^2 x4^3 x4^2*y4 x4*y4^2 y4^3 x4^3*y4 x4*y4^3...
```

182

```
0 0 1 0 x4 2*y4 0 x4^2 2*x4*y4 3*y4^2 x4^3 3*x4*y4^2
0 -1 0 -2*x4 -y4 0 -3*x4^2 -2*x4*y4 -y4^2 0 -3*x4^2*y4 -y4^3];
```

```
R=[-sqrt(1/3) -sqrt(1/3) sqrt(1/3) sqrt(1/3)];
S=[-sqrt(1/3) sqrt(1/3) -sqrt(1/3) sqrt(1/3)];
k=zeros(12,12);
```

187 for i=1:length(R)

```
r=R(i); s=S(i);
N1=(1/4)*(1-r)*(1-s);
N2=(1/4)*(1+r)*(1-s);
N3=(1/4)*(1+r)*(1+s);
```

192

```
N4=(1/4)*(1-r)*(1+s);
N=[N1 N2 N3 N4];
```

```
x=N*X; y=N*Y;
xr=(1/4)*[-1+s 1-s 1+s -1-s]*X;
xs=(1/4)*[-1+r -1-r 1+r 1-r]*X;
197 yr=(1/4)*[-1+s 1-s 1+s -1-s]*Y;
ys=(1/4)*[-1+r -1-r 1+r 1-r]*Y;
```

202

```
P21r=0; P21s=0;
P22r=0; P22s=0;
P23r=0; P23s=0;
P24r=0; P24s=0;
P25r=xr; P25s=xs;
P26r=2*yr; P26s=2*ys;
```

```

P27r=0; P27s=0;
P28r=2*x*xr; P28s=2*x*xs;
207 P29r=2*(xr*y+x*yr); P29s=2*(xs*y+x*ys);
P210r=6*y*yr; P210s=6*y*ys;
P211r=3*x^2*xr; P211s=3*x^2*xs;
P212r=3*(xr*y^2+2*x*y*yr); P212s=3*(xs*y^2+2*x*y*ys);
P31r=0; P31s=0;
212 P32r=0; P32s=0;
P33r=0; P33s=0;
P34r=-2*xr; P34s=-2*xs;
P35r=-yr; P35s=-ys;
P36r=0; P36s=0;
217 P37r=-6*x*xr; P37s=-6*x*xs;
P38r=-2*(xr*y+x*yr); P38s=-2*(xs*y+x*ys);
P39r=-2*y*yr; P39s=-2*y*ys;
P310r=0; P310s=0;
P311r=-3*(2*x*xr*y+x^2*yr); P311s=-3*(2*x*xs*y+x^2*ys);
222 P312r=-3*y^2*yr; P312s=-3*y^2*ys;
Q1=[ys*P31r-yr*P31s ys*P32r-yr*P32s ys*P33r-yr*P33s
-xr*P21s+xs*P21r -xr*P22s+xs*P22s -xr*P23s+xs*P23r
-ys*P21r+yr*P21s+xr*P31s-xs*P31r -ys*P22r+yr*P22s...
+xr*P32s-xs*P32r -ys*P23r+yr*P23s+xr*P33s-xs*P33r];
227 Q2=[ys*P34r-yr*P34s ys*P35r-yr*P35s ys*P36r-yr*P36s
-xr*P24s+xs*P24r -xr*P25s+xs*P25r -xr*P26s+xs*P26r
-ys*P24r+yr*P24s+xr*P34s-xs*P34r -ys*P25r+yr*P25s...
+xr*P35s-xs*P35r -ys*P26r+yr*P26s+xr*P36s-xs*P36r];
Q3=[ys*P37r-yr*P37s ys*P38r-yr*P38s ys*P39r-yr*P39s
232 -xr*P27s+xs*P27r -xr*P28s+xs*P28r -xr*P29s+xs*P29r
-ys*P27r+yr*P27s+xr*P37s-xs*P37r -ys*P28r+yr*P28s...
+xr*P38s-xs*P38r -ys*P29r+yr*P29s+xr*P39s-xs*P39r];
Q4=[ys*P310r-yr*P310s ys*P311r-yr*P311s ys*P312r-yr*P312s
237 -xr*P210s+xs*P210r -xr*P211s+xs*P211r -xr*P212s+xs*P212r
-ys*P210r+yr*P210s+xr*P310s-xs*P310r...
-ys*P211r+yr*P211s+xr*P311s-xs*P311r...
-ys*P212r+yr*P212s+xr*P312s-xs*P312r];
detJ=xr*ys-xs*yr;
Q=(1/detJ)*[Q1 Q2 Q3 Q4];
242 B=Q/C;
k=k+B'*D*B*detJ;
end
KE=k;

247 end

```

Listing C.3: Combined Membrane-Bending SIMP Model

```

%% SIMP METHOD: Membrane & Bending
function [el,nel,X,Y,x]=SIMPmb(params,tor,varargin)
3
%Setup Parameters
if isempty(varargin)==1
    config=1;
elseif isempty(varargin)==0

```

```

8     config=varargin{1};
    end
    params=params{config};
    if isfield(params,'points')==0
        mh=params.mesh;
13 elseif isfield(params,'points')==1
        mh=params.points;
    end
    nelxp=params.n;
    [~,n]=size(nelxp);
18 nelx=zeros(n);
    for i=1:n
        nelx(i)=nelxp{i}-1;
    end
    nelx=sum(nelx);
23 nely=params.m{1}-1;
    nel=nelx*nely;

    %Input Parameters
    volfrac=0.5; %INPUT: Volume fraction
28 penal=3; %INPUT: Penalty
    rmin=1.5; %INPUT: Radius of filter
    conv=0.01; %INPUT: Change convergence
    xmin=0.001; %INPUT: Minimum element thickness
    fdofs=1:6*(nely+1); %INPUT: Constrain DOFs
33 [F]=Faero(tor,nelx,nelxp,nely,6,1);
    disp('Aerodynamic Forces Computed')

    %Elemental Stiffness Matrices
    edof=zeros(24,nel); KE=zeros(24*nel,24);
38 X=zeros(4,nel); Y=zeros(4,nel);
    A=zeros(nel,1);
    for el=1:nel
        m=floor((el-1)/nely);
        n=rem(el-1,nely)+1;
43 e=[6*m*(nely+1)+6*n+1 6*(m+1)*(nely+1)+6*n+1 ...
        6*(m+1)*(nely+1)+6*n-5 6*m*(nely+1)+6*n-5];
        edof(:,el) = [e(1);e(1)+1;e(1)+2;e(1)+3;e(1)+4;e(1)+5;
            e(2);e(2)+1;e(2)+2;e(2)+3;e(2)+4;e(2)+5;
            e(3);e(3)+1;e(3)+2;e(3)+3;e(3)+4;e(3)+5;
48 e(4);e(4)+1;e(4)+2;e(4)+3;e(4)+4;e(4)+5];
        mhi= [m*(nely+1)+n+1;(m+1)*(nely+1)+n+1;
            (m+1)*(nely+1)+n;m*(nely+1)+n];
        X(:,el)=[mh(mhi(1),1);mh(mhi(2),1);mh(mhi(3),1);mh(mhi(4),1)];
        Y(:,el)=[mh(mhi(1),2);mh(mhi(2),2);mh(mhi(3),2);mh(mhi(4),2)];
53 A(el)=areaP(X(:,el),Y(:,el));
        [KE(24*(el-1)+1:24*el,1:24)]=Ke(X(:,el),Y(:,el));
    end
    disp('Elemental Stiffness Matrices Computed')

58 %Start Iteration
    vol=volfrac*sum(A);

```

```

x(1:nel,1)=volfrac;
loop=0;change=1;
while change>conv
63   loop=loop+1;
    xold=x;

    %Calculate Global Displacements
    [U]=FE(nelx,nely,edof,KE,x,penal,F,fdofs);
68
    %Objective Function and Sensitivity Analysis
    c=0;dc=zeros(nel,1);
    for el=1:nel
        Ue=U(edof(:,el),1);
73        c=c+x(el)^penal*Ue'*KE(24*(el-1)+1:24*el,1:24)*Ue;
        dc(el)=-penal*x(el)^(penal-1)*Ue'*KE(24*(el-1)+1:24*el,1:24)...
            '*Ue;
    end

    %Filtering of Sensitivities
78   [dc]=check(nelx,nely,rmin,x,dc);

    %Design Update by the Optimality Criteria
    [x]=OC(x,xmin,vol,dc,A);

83   %Print Results
    change=max(abs(x-xold));
    disp([' It.: ' sprintf('%4i',loop) ...
        ' Obj.: ' sprintf('%10.6f',c) ...
        ' Vol.: ' sprintf('%6.3f',(x'*A)/sum(A)) ...
88        ' ch.: ' sprintf('%6.3f',change) ])
    end
    toc

    %Plot Densities
93 h=figure;set(h,'color',[1 1 1]);colormap(bone);patch(X,Y,-x');
    axis equal;axis tight;axis off

    end

98   %% OPTIMALITY CRITERIA UPDATE
    function [xnew]=OC(x,xmin,vol,dc,A)
    l1=0;l2=100000;move=0.1;eta=0.5;
    while (l2-l1>1e-4)
103   lmid=0.5*(l2+l1);
        xnew=max(xmin,max(x-move,min(1.,min(x+move,x*(-dc./(lmid*A)).^...
            eta)))));
        if xnew'*A-vol>0;
            l1=lmid;
        else
108   l2=lmid;
        end
    end

```



```

    end

    end
113

    %% MESH-INDEPENDENCY FILTER
    function [dcn]=check(nelx,nely,rmin,x,dc)
    xmap=reshape(x,nely,nelx);
118 dcnmap=reshape(dc,nely,nelx);
    dcnmap=zeros(nely,nelx);
    for i=1:nelx
        for j=1:nely
            sum=0.0;
123     for k=max(i-floor(rmin),1):min(i+floor(rmin),nelx);
                for l=max(j-floor(rmin),1):min(j+floor(rmin),nely)
                    fac=rmin-sqrt((i-k)^2+(j-l)^2);
                    sum=sum+max(0,fac);
                    dcnmap(j,i)=dcnmap(j,i)+max(0,fac)*xmap(l,k)*dcmap(l,k);
128     end
                end
            dcnmap(j,i)=dcnmap(j,i)/(xmap(j,i)*sum);
        end
    end
133 dcn=reshape(dcnmap,nelx*nely,1);

    end

138 %% FE-ANALYSIS
    function [U]=FE(nelx,nely,edof,KE,x,penal,F,fdofs)

    %%Compile Stiffness Matrix
    K=zeros(6*(nelx+1)*(nely+1),6*(nelx+1)*(nely+1));
143 U=zeros(6*(nely+1)*(nelx+1),1);
    for el=1:nelx*nely
        K(edof(:,el),edof(:,el))=...
            K(edof(:,el),edof(:,el))+x(el)^penal*KE(24*(el-1)+1:24*el...
                ,1:24);
    end
148
    %% Define Loads and Supports
    fixeddofs=fdofs;
    alldofs=1:6*(nely+1)*(nelx+1);
    freedofs=setdiff(alldofs,fixeddofs);
153 U(freedofs,:)=K(freedofs,freedofs)\F(freedofs,:);
    U(fixeddofs,:)= 0;

    end

158
    %% ELEMENT STIFFNESS MATRIX
    function [KE]=Ke(X,Y)

```

```

%Constitutive Matrix
E=200e9; %INPUT: Young's modulus
163 nu=0.3; %INPUT: Poisson's ratio
th=1; %INPUT: thickness of element
Dm=(E*th/(1-nu^2))*[1 nu 0;nu 1 0;0 0 (1-nu)/2];
Db=Dm*th^2/12;

168 %Formulation of Bending [C] Matrix
x1=X(1); x2=X(2); x3=X(3); x4=X(4);
y1=Y(1); y2=Y(2); y3=Y(3); y4=Y(4);
C=[1 x1 y1 x1^2 x1*y1 y1^2 x1^3 x1^2*y1 x1*y1^2 y1^3 x1^3*y1 x1*y1...
^3
0 0 1 0 x1 2*y1 0 x1^2 2*x1*y1 3*y1^2 x1^3 3*x1*y1^2
173 0 -1 0 -2*x1 -y1 0 -3*x1^2 -2*x1*y1 -y1^2 0 -3*x1^2*y1 -y1^3
1 x2 y2 x2^2 x2*y2 y2^2 x2^3 x2^2*y2 x2*y2^2 y2^3 x2^3*y2 x2*y2...
^3
0 0 1 0 x2 2*y2 0 x2^2 2*x2*y2 3*y2^2 x2^3 3*x2*y2^2
0 -1 0 -2*x2 -y2 0 -3*x2^2 -2*x2*y2 -y2^2 0 -3*x2^2*y2 -y2^3
1 x3 y3 x3^2 x3*y3 y3^2 x3^3 x3^2*y3 x3*y3^2 y3^3 x3^3*y3 x3*y3...
^3
178 0 0 1 0 x3 2*y3 0 x3^2 2*x3*y3 3*y3^2 x3^3 3*x3*y3^2
0 -1 0 -2*x3 -y3 0 -3*x3^2 -2*x3*y3 -y3^2 0 -3*x3^2*y3 -y3^3
1 x4 y4 x4^2 x4*y4 y4^2 x4^3 x4^2*y4 x4*y4^2 y4^3 x4^3*y4 x4*y4...
^3
0 0 1 0 x4 2*y4 0 x4^2 2*x4*y4 3*y4^2 x4^3 3*x4*y4^2
0 -1 0 -2*x4 -y4 0 -3*x4^2 -2*x4*y4 -y4^2 0 -3*x4^2*y4 -y4^3];

183 %Membrane & Bending Gaussian Quadrature Integration
R=[-sqrt(1/3) -sqrt(1/3) sqrt(1/3) sqrt(1/3)];
S=[-sqrt(1/3) sqrt(1/3) -sqrt(1/3) sqrt(1/3)];
km=zeros(8,8); kb=zeros(12,12);
188 for i=1:length(R)
r=R(i); s=S(i);

%Bilinear Shape Functions;x-y Coordinates and Derivatives;det(J)
193 N1=(1/4)*(1-r)*(1-s);
N2=(1/4)*(1+r)*(1-s);
N3=(1/4)*(1+r)*(1+s);
N4=(1/4)*(1-r)*(1+s);
N=[N1 N2 N3 N4];
x=N*X; y=N*Y;
198 xr=(1/4)*[s-1 1-s 1+s -1-s]*X;
xs=(1/4)*[r-1 -1-r 1+r 1-r]*X;
ys=(1/4)*[r-1 -1-r 1+r 1-r]*Y;
yr=(1/4)*[s-1 1-s 1+s -1-s]*Y;
detJ=xr*ys-xs*yr;

203 %Membrane Formulation
N1r=(1/4)*(s-1); N1s=(1/4)*(r-1);
N2r=(1/4)*(1-s); N2s=(1/4)*(-r-1);
N3r=(1/4)*(1+s); N3s=(1/4)*(1+r);
208 N4r=(1/4)*(-1-s); N4s=(1/4)*(1-r);

```

```

B1=[ys*N1r-yr*N1s 0;0 xr*N1s-xs*N1r;
    xr*N1s-xs*N1r ys*N1r-yr*N1s];
B2=[ys*N2r-yr*N2s 0;0 xr*N2s-xs*N2r;
    xr*N2s-xs*N2r ys*N2r-yr*N2s];
213 B3=[ys*N3r-yr*N3s 0;0 xr*N3s-xs*N3r;
    xr*N3s-xs*N3r ys*N3r-yr*N3s];
B4=[ys*N4r-yr*N4s 0;0 xr*N4s-xs*N4r;
    xr*N4s-xs*N4r ys*N4r-yr*N4s];
Bm=(1/detJ)*[B1 B2 B3 B4];
218 km=km+Bm'*Dm*Bm*detJ;

%Bending Formulation
P21r=0; P21s=0;
P22r=0; P22s=0;
223 P23r=0; P23s=0;
P24r=0; P24s=0;
P25r=xr; P25s=xs;
P26r=2*yr; P26s=2*ys;
P27r=0; P27s=0;
228 P28r=2*x*xr; P28s=2*x*xs;
P29r=2*(xr*y+x*yr); P29s=2*(xs*y+x*ys);
P210r=6*y*yr; P210s=6*y*ys;
P211r=3*x^2*xr; P211s=3*x^2*xs;
P212r=3*(xr*y^2+2*x*y*yr); P212s=3*(xs*y^2+2*x*y*ys);
233 P31r=0; P31s=0;
P32r=0; P32s=0;
P33r=0; P33s=0;
P34r=-2*xr; P34s=-2*xs;
P35r=-yr; P35s=-ys;
238 P36r=0; P36s=0;
P37r=-6*x*xr; P37s=-6*x*xs;
P38r=-2*(xr*y+x*yr); P38s=-2*(xs*y+x*ys);
P39r=-2*y*yr; P39s=-2*y*ys;
P310r=0; P310s=0;
243 P311r=-3*(2*x*xr*y+x^2*yr); P311s=-3*(2*x*xs*y+x^2*ys);
P312r=-3*y^2*yr; P312s=-3*y^2*ys;
Q1=[ys*P31r-yr*P31s ys*P32r-yr*P32s ys*P33r-yr*P33s
    -xr*P21s+xs*P21r -xr*P22s+xs*P22s -xr*P23s+xs*P23r
    -ys*P21r+yr*P21s+xr*P31s-xs*P31r -ys*P22r+yr*P22s...
248 +xr*P32s-xs*P32r -ys*P23r+yr*P23s+xr*P33s-xs*P33r];
Q2=[ys*P34r-yr*P34s ys*P35r-yr*P35s ys*P36r-yr*P36s
    -xr*P24s+xs*P24r -xr*P25s+xs*P25r -xr*P26s+xs*P26r
    -ys*P24r+yr*P24s+xr*P34s-xs*P34r -ys*P25r+yr*P25s...
    +xr*P35s-xs*P35r -ys*P26r+yr*P26s+xr*P36s-xs*P36r];
253 Q3=[ys*P37r-yr*P37s ys*P38r-yr*P38s ys*P39r-yr*P39s
    -xr*P27s+xs*P27r -xr*P28s+xs*P28r -xr*P29s+xs*P29r
    -ys*P27r+yr*P27s+xr*P37s-xs*P37r -ys*P28r+yr*P28s...
    +xr*P38s-xs*P38r -ys*P29r+yr*P29s+xr*P39s-xs*P39r];
Q4=[ys*P310r-yr*P310s ys*P311r-yr*P311s ys*P312r-yr*P312s
258 -xr*P210s+xs*P210r -xr*P211s+xs*P211r -xr*P212s+xs*P212r
    -ys*P210r+yr*P210s+xr*P310s-xs*P310r...
    -ys*P211r+yr*P211s+xr*P311s-xs*P311r...

```

```

        -ys*P212r+yr*P212s+xr*P312s-xs*P312r];
    Q=(1/detJ)*[Q1 Q2 Q3 Q4];
263 Bb=Q/C;
    kb=kb+Bb'*Db*Bb*detJ;
end

%Superposition of Membrane and Bending Stiffness Matrices
268 k11m=km(1:2,1:2);k12m=km(1:2,3:4);
    k13m=km(1:2,5:6);k14m=km(1:2,7:8);
    k21m=km(3:4,1:2);k22m=km(3:4,3:4);
    k23m=km(3:4,5:6);k24m=km(3:4,7:8);
    k31m=km(5:6,1:2);k32m=km(5:6,3:4);
273 k33m=km(5:6,5:6);k34m=km(5:6,7:8);
    k41m=km(7:8,1:2);k42m=km(7:8,3:4);
    k43m=km(7:8,5:6);k44m=km(7:8,7:8);
    k11b=kb(1:3,1:3);k12b=kb(1:3,4:6);
    k13b=kb(1:3,7:9);k14b=kb(1:3,10:12);
278 k21b=kb(4:6,1:3);k22b=kb(4:6,4:6);
    k23b=kb(4:6,7:9);k24b=kb(4:6,10:12);
    k31b=kb(7:9,1:3);k32b=kb(7:9,4:6);
    k33b=kb(7:9,7:9);k34b=kb(7:9,10:12);
    k41b=kb(10:12,1:3);k42b=kb(10:12,4:6);
283 k43b=kb(10:12,7:9);k44b=kb(10:12,10:12);
    A=areaP(X,Y);
    k=0.3*E*th*A;
    k5=-0.5*k;
    z21=zeros(2,1); z23=zeros(2,3);z31=zeros(3,1);
288 KE=[k11m z23 z21 k12m z23 z21 k13m z23 z21 k14m z23 z21
        z23' k11b z31 z23' k12b z31 z23' k13b z31 z23' k14b z31
        z21' z31' k z21' z31' k5 z21' z31' k5 z21' z31' k5
        k21m z23 z21 k22m z23 z21 k23m z23 z21 k24m z23 z21
        z23' k21b z31 z23' k22b z31 z23' k23b z31 z23' k24b z31
293 z21' z31' k5 z21' z31' k z21' z31' k5 z21' z31' k5
        k31m z23 z21 k32m z23 z21 k33m z23 z21 k34m z23 z21
        z23' k31b z31 z23' k32b z31 z23' k33b z31 z23' k34b z31
        z21' z31' k5 z21' z31' k5 z21' z31' k z21' z31' k5
        k41m z23 z21 k42m z23 z21 k43m z23 z21 k44m z23 z21
298 z23' k41b z31 z23' k42b z31 z23' k43b z31 z23' k44b z31
        z21' z31' k5 z21' z31' k5 z21' z31' k5 z21' z31' k ];

end

```

## Appendix D. Miscellaneous MATLAB Codes

Listing D.1: View Geometry of “params” Structure

```
function geomview(params,type,varargin)
if isempty(varargin)==1
    config=1;
4 elseif isempty(varargin)==0
    config=varargin{1};
end
params=params{config};
if isfield(params,'points')==0
9     mh=params.mesh;
elseif isfield(params,'points')==1
    mh=params.points;
end
[~,n]=size(params.n);
14 ngpx=1;
for i=1:n
    ngpx=ngpx+params.n{i}-1;
end
ngpy=params.m{1};
19 x=mh(:,1); y=mh(:,2);
[ngp,dim]=size(mh);
if dim==2
    z=zeros(ngp,1);
else
24     z=mh(:,3);
end
figure
if strcmp(type,'points')
    plot3(x,y,z,'x'); axis equal; grid
29 else
    x=reshape(x,ngpy,ngpx);
    y=reshape(y,ngpy,ngpx);
    z=reshape(z,ngpy,ngpx);
    if strcmp(type,'mesh')
34         m=mesh(x,y,z); axis equal
            set(m,'facecolor','none')
            set(m,'EdgeColor',[1 0 0])
    elseif strcmp(type,'surf')
        surf(x,y,z); axis equal
39     end
end
end
```

Listing D.2: View Degrees of Freedom for a Geometry

```
function dofview(params,dof,varargin)
if isempty(varargin)==1
    config=1;
elseif isempty(varargin)==0
5     config=varargin{1};
end
params=params{config};
```

```

    if isfield(params,'points')==0
        mesh=params.mesh;
10 elseif isfield(params,'points')==1
        mesh=params.points;
    end
    xcoor=mesh(:,1);
    ycoor=mesh(:,2);
15 n=length(xcoor);
    for i=1:n
        num(i)=dof*(i-1)+1;
    end
    numlbl=num2str(num');
20 fig = figure;
    text(xcoor,ycoor,numlbl)
    axis([min(xcoor) max(xcoor) min(ycoor) max(ycoor)])
    set (fig, 'Units', 'normalized', 'Position', [0,0,1,1]);

25 end

```

Listing D.3: Transfer Aerodynamic Loads from VLM to FEM

```

function [F,Fpx,Fpy,Fpz]=Faero(tor,nelx,nelxp,nely,dof,Dest)
d=tor.drag;
g=tor.geo;
l=tor.lattice;
5 r=tor.ref;
res=tor.results;
s=tor.state;

dofn=dof;dofe=4*dofn;
10 nelp=nely*nelxp;
nel=nelx*nely;
tdof=dof*(nely+1)*(nelx+1);
F=res.F;
xlat=l.XYZ(:,:,1);
15 alpha=s.alpha;
rhoinf=s.rho;
Vinf=s.AS;
S=r.S_ref;
CDo=sum(d.CD0);
20 CDmax=1;

%Split Panel Force into Nodal Forces
Fpx=F(:,2);
Fpy=-F(:,1);
25 Fpz=F(:,3);
Fn=zeros(tdof,1);
edof=zeros(dofe,nel);
for el=1:nel
    n1=floor((el-1)/nely);
30 n2=rem(el-1,nely);
    ni=[n1 n2+1;n1+1 n2+1;n1+1 n2;n1 n2];
    for node=1:4

```

```

n1i=ni(node,1); n2i=ni(node,2);
for dof=1:dofn
35     edof((node-1)*dofn+dof,e1)=...
        dof*n1i*(nely+1)+dof*n2i+dof;
    end
end
switch dof
40     case 2
        Fn(edof(1:2:7,e1),1)=...
            Fn(edof(1:2:7,e1),1)+0.25*Fpx(e1);
        Fn(edof(2:2:8,e1),1)=...
            Fn(edof(2:2:8,e1),1)+0.25*Fpy(e1);
45     case 3
        Fn(edof(1:3:10,e1),1)=...
            Fn(edof(1:3:10,e1),1)+0.25*Fpz(e1);
    case 6
50     Fn(edof(1:6:19,e1),1)=...
            Fn(edof(1:6:19,e1),1)+0.25*Fpx(e1);
        Fn(edof(2:6:20,e1),1)=...
            Fn(edof(2:6:20,e1),1)+0.25*Fpy(e1);
        Fn(edof(3:6:21,e1),1)=...
            Fn(edof(3:6:21,e1),1)+0.25*Fpz(e1);
55     otherwise
        error('Unknown DOF model choice (@Faero: Line 56)')
    end
end
end

60 %Transfer Root Chord Loads to 2nd Chord
Fn(dof*(nely+1)+1:2*dof*(nely+1))=Fn(dof*(nely+1)...
    +1:2*dof*(nely+1))+Fn(1:dof*(nely+1));

if Dest==1
65     %Calculate Skin Friction Drag
    qinf=0.5*rho*Vinf^2;
    Drag=qinf*S*(CDo+(CDmax-CDo)*sin(alpha));
    Dragbody=[cos(alpha) sin(alpha);
              -sin(alpha) cos(alpha)]*[0;Drag];
70     Normal=Dragbody(1); Axial=Dragbody(2);
    chord=zeros(nelx,1); LE=zeros(nelx,1); TE=zeros(nelx,1);
    le=zeros(nelx,1); te=zeros(nelx,1);
    for ch=1:nelx
        LE(ch)=(ch-1)*nely+1;
75     le(ch)=ch*(nely+1)+1;
        TE(ch)=(ch)*nely;
        te(ch)=(ch+1)*(nely+1);
        chord(ch)=xlat(TE(ch),3)-xlat(LE(ch),2);
    end
80     chtot=sum(chord);
    for ch=1:nelx
        Nch=Normal*(chord(ch)/chtot);
        Ach=Axial*(chord(ch)/chtot);
        N=Nch/(nely+1);
    end
end

```

```

85     A=Ach/(nely+1);
      switch dof
        case 2
          Fn(2*le(ch):2:2*te(ch),1)=...
            Fn(2*le(ch):2:2*te(ch),1)-A;
90     Fpy(LE(ch):TE(ch))=Fpy(LE(ch):TE(ch))-A;
        case 3
          Fn(3*le(ch)-2:3:3*te(ch)-2,1)=...
            Fn(3*le(ch)-2:3:3*te(ch)-2,1)+N;
95     Fpz(LE(ch):TE(ch))=Fpz(LE(ch):TE(ch))+N;
        case 6
          Fn(6*le(ch)-4:6:6*te(ch)-4,1)=...
            Fn(6*le(ch)-4:6:6*te(ch)-4,1)-A;
            Fn(6*le(ch)-3:6:6*te(ch)-3,1)=...
            Fn(6*le(ch)-3:6:6*te(ch)-3,1)+N;
100    Fpy(LE(ch):TE(ch))=Fpy(LE(ch):TE(ch))-A;
            Fpz(LE(ch):TE(ch))=Fpz(LE(ch):TE(ch))+N;
        otherwise
          end
      end
105 end

      F=Fn;
      end

```



## Bibliography

1. “Bakshaev LIG-7”, January 2012. URL <http://www.ctrl-c.liu.se/misc/ram/rk.html>.
2. “Eagle Owl flyby”, January 2012. URL <http://kevlewis.deviantart.com/art/Eagle-Owl-Flyby-148980524>.
3. “Eagle Owl perching”, January 2012. URL <http://albertan1956.blogspot.com/2011/08/owl-rescue-stops-traffic-in-germany.html>.
4. “Eagle Owl soaring”, January 2012. URL <http://chrismillerfalconry.weebly.com/>.
5. “MAK-10”, January 2012. URL <http://planeshots.tumblr.com/post/224462915/makhonine-mak-10-101-telescoping-wing-this-is>.
6. Aber, K. “wingGeomGUI.m”, August 2011. Air Vehicles Directorate, Air Force Research Laboratory, WPAFB, Dayton, OH 45433-7531.
7. Aurora, Jasbir S. *Introduction to Optimum Design*. Elsevier Academic Press, 2nd edition, 2004.
8. Bendsøe, M. P. and O. Sigmund. *Topology Optimization: Theory Methods and Applications*. Springer, 2nd edition, 2003-2004.
9. Black, Sara. “The Changing Shape of Future Aircraft”. *High-Performance Composites*. Composites World, 2006.
10. Christensen, Peter W. and Anders Klarbring. *An Introduction to Structural Optimization*, volume 153 of *Solid Mechanics and Its Applications*. Springer, 1st edition, 2010.
11. Cook, Robert D., David S. Malkus, and Michael E. Plesha. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, 1989.
12. Gere, James M. *Mechanics of Materials*. Thomson-Brooks/Cole, 6th edition, 2004.
13. Gosz, Michael R. *Finite Element Method: Applications in Solids, Structures, and Heat Transfer*. Mechanical Engineering: A Series of Textbooks and Reference Books. CRC Press, 1st edition, 2006.
14. Inoyama, Daisaku, Brian P. Sanders, and James J. Joo. “Conceptual Design and Multidisciplinary Optimization of In-plane Morphing Wing Structures”. *Smart Structures and Materials 2006: Modeling, Signal Processing, and Control*, 6166(616601):1–11, 2006.
15. Inoyama, Daisaku, Brian P. Sanders, and James J. Joo. “Topology Synthesis of Distributed Actuation Systems for Morphing Wing Structures”. *Journal of Aircraft*, 44(4):1205–1213, July-August 2007.

16. Inoyama, Daisaku, Brian P. Sanders, and James J. Joo. "Topology Optimization Approach for the Determination of the Multiple-Configuration Morphing Wing Structure". *Journal of Aircraft*, 45(6):1853–1862, November-December 2008.
17. Joo, James J., Gregory W. Reich, and James T. Westfall. "Flexible Skin Design for Morphing Aircraft Applications via Topology Optimization". *Journal of Intelligent Material Systems and Structures*, 20:1969–1985, November 2009.
18. Joo, James J. and Brian Sanders. "Simultaneous Structure and Mechanism Design for an Adaptive Wing Using Topology Optimization". *2005 ASME International Mechanical Engineering Congress and Exposition*. November 2005.
19. Joo, James J., Brian Sanders, Terrence Johnson, and Mary I. Frecker. "Optimal Actuator Location within a Morphing Wing Scissor Mechanism Configuration". *Smart Structures and Materials 2006: Modeling, Signal Processing, and Control*, 6166(616603):1–12, March 2006.
20. Joshi, Shiv P., Zeb Tidwell, William A. Crossley, and Sekaripuram Ramakrishnan. "Comparison of Morphing Wing Strategies Based Upon Aircraft Performance Impacts". *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, April 2004.
21. Kwon, Young W. and Hyochoong Bang. *The Finite Element Method Using MATLAB*. CRC Mechanical Engineering Series. CRC Press, 2nd edition, 2000.
22. Lawlor, Maryann. "The Shape of Wings to Come". Robert K. Ackerman (editor), *SIGNAL*. Armed Forces Communications and Electronics Association, October 2006.
23. Logan, Daryl L. *A First Course in the Finite Element Method*. Thomson, 4th edition, 2007.
24. Lu, Kerr-Jia and Sridhar Kota. "Design of Compliant Mechanisms for Morphing Structural Shapes". *Journal of Intelligent Material Systems and Structures*, 14:379–391, 2003.
25. Lukens, Jennifer M., Gregory W. Reich, and Brian Sanders. "Wing Mechanization Design and Analysis for a Perching Micro Air Vehicle". *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, April 2008.
26. Maute, K. and M. Allen. "Conceptual Design of Aeroelastic Structures by Topology Optimization". *Structural Multidisciplinary Optimization*, 27(1-2):27–42, January 2004.
27. Maute, K. and G.W. Reich. "Integrated Multidisciplinary Topology Optimization Approach to Adaptive Wing Design". *Journal of Aircraft*, 43(1):80309–80429, January-February 2006.

28. Melin, Thomas. “Tornado: A Vortex Lattice Program for Conceptual Aircraft Design”, March 2000. Version 135 released March 20, 2010.
29. Melin, Thomas. *A Vortex Lattice MATLAB Implementation for Linear Aerodynamic Wing Applications*. Master’s thesis, Royal Institute of Technology (KTH), 2000.
30. Murray, Gabriel, Farhan Gandhi, and Charles Bakis. “Flexible Matrix Composite Skins for One-Dimensional Wing Morphing”. *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 13. American Institute of Aeronautics and Astronautics, April 2007.
31. Norton, Robert L. *Machine Design: An Integrated Approach*. Pearson Prentice Hall, 3rd edition, 2006.
32. Olympio, K. and F. Gandhi. “Zero- $\nu$  Cellular Honeycomb Flexible Skins for One-Dimensional Wing Morphing”. *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, April 2007.
33. Olympio, K. Raymond, January 2012. URL <https://sites.google.com/site/raymondolympio/research-1>.
34. Olympio, K. Raymond and Farhan Gandhi. “Modeling and Numerical Analysis of Skin Design Concepts”. *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, May 2009.
35. Olympio, Kingnide Raymond and Farhan Gandhi. “Skin Designs Using Multi-Objective Topology Optimization”. *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 26. American, April 2008.
36. Rao, Singiresu S. *The Finite Element Method in Engineering*. Elsevier Butterworth-Heinemann, 4th edition, 2005.
37. Reich, Gregory W., Franklin E. Eastep, Aaron Altman, and Roberto Albertani. “Transient Poststall Aerodynamic Modeling for Extreme Maneuvers in Micro Air Vehicles”. *Journal of Aircraft*, 48(2):403–411, March 2011.
38. Robertson, Darrel K. “Perching Maneuver Data”, December 2011. Personal Conversation.
39. Robertson, Darrel K., James J. Joo, and Gregory W. Reich. “Vortex Particle Aerodynamic Modeling of Perching Maneuvers with Micro Air Vehicles”. *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, April 2010.
40. Seigler, T.M., J.L. Lubbers, and Gregory W. Reich. “Perch Landing Maneuvers for a Rotating Wing MAV”. *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, April 2010.

41. Song, Arnold, Xiaodong Tian, Emily Israeli, Ricardo Galvao, Kristin Bishop, Sharon Swartz, and Kenneth Breuer. “Aeromechanics of Membrane Wings with Implications for Animal Flight”. *AIAA Journal*, 46(8):2096–2106, August 2008.
42. Svanberg, Krister. “The Method of Moving Asymptotes—A New Method for Structural Optimization”. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.
43. Thomas, B. G. and R. L. Finney. *Calculus and Analytic Geometry*. Addison-Wesley, 1984.
44. Timoshenko, S. *Theory of Plates and Shells*. Engineering Societies Monographs. McGraw-Hill Book Company, 2nd edition, 1959.
45. Wilson, J.R. “Morphing UAVs change the shape of warfare”. *Aerospace America*, 29. American Institute of Aeronautics and Astronautics, February 2004. Program Profile.

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved</i> OMB No. 0704-0188		
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 22-03-2012		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From — To) October 2010—March 2012	
4. TITLE AND SUBTITLE Conceptual Layout of Wing Structure Using Topology Optimization for Morphing Micro Air Vehicles in A Perching Maneuver			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)  James R. Elgersma			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENY) 2950 Hobson Way WPAFB OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER  AFIT/GAE/ENY/12-M11		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)  AFRL MAV Program Dr. Gregory Reich AFRL/RBSA 2210 8TH ST, B20146 R219, WPAFB, OH 45433 Gregory.Reich@WPAFB.AF.MIL (937)-255-8296			10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RBSA		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED					
13. SUPPLEMENTARY NOTES This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT  A topology optimization model for conceptual wing structure layouts of morphing micro air vehicles (MAVs) has been developed and implemented in MATLAB. Specifically, a six degree-of-freedom finite element (FE) model with a general quadrilateral discretization scheme was created by superposition of a known simple linear plane membrane element and a Kirchhoff plate bending element derived herein. The purpose of the six degree-of-freedom model was to accommodate in-plane and out-of-plane aerodynamic loading combinations. The FE model was validated and the MATLAB implementation was verified with classical beam and plate solutions. A compliance minimization optimization objective was then formulated with the Solid Isotropic Material with Penalization (SIMP) method, subject to the equilibrium constraint computed by the FE model, and solved with the Optimality Criteria (OC) method. With the topology optimization model in place, four aerodynamic loading scenarios were extracted from points along a feasible MAV perching flight trajectory and used to determine wing thickness distributions for given planform shapes. The results suggest conceptual structural layouts in morphing MAVs, but equally important, the simple MATLAB implementation of the model can be adapted for a variety of objective statements for MAV morphing wing design.					
15. SUBJECT TERMS morphing, Micro Air Vehicle, perching, topology optimization, finite element, wing structure					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Donald Kunz (AFIT/ENY), Civ, USAF
U	U	U	U	189	19b. TELEPHONE NUMBER (Include Area Code)  (937)-785-3636, ext 4548