

UCLA

UCLA Electronic Theses and Dissertations

Title

Computational Transonic Flutter Solutions for Cranked Wings by the Direct Eulerian-Lagrangian Method

Permalink

<https://escholarship.org/uc/item/9rk54186>

Author

Mellquist, Erik Charles

Publication Date

2014

Supplemental Material

<https://escholarship.org/uc/item/9rk54186#supplemental>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

**Computational Transonic Flutter Solutions for
Cranked Wings by the Direct
Eulerian-Lagrangian Method**

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Aerospace Engineering

by

Erik Charles Mellquist

2014

© Copyright by
Erik Charles Mellquist
2014

ABSTRACT OF THE DISSERTATION

Computational Transonic Flutter Solutions for Cranked Wings by the Direct Eulerian-Lagrangian Method

by

Erik Charles Mellquist

Doctor of Philosophy in Aerospace Engineering

University of California, Los Angeles, 2014

Professor Oddvar O. Bendiksen, Chair

In this dissertation, a three-dimensional computational aeroelastic simulation for cranked, highly-swept wings is developed, and solutions are presented for several wing models. The computational model is a fully nonlinear coupled fluid-structure simulation based on the Direct Eulerian-Lagrangian coupling methodology. The wing is modeled using nonlinear modified von Karman plate finite elements. Large deformation is accounted for through the use of element-attached local coordinate systems referenced to a single global coordinate system. The fluid is modeled using the mixed Eulerian-Lagrangian formulation of the classical Euler equations and is discretized using a Galerkin finite element approach on an unstructured tetrahedral mesh. The fluid and structural models are coupled by the Direct Eulerian-Lagrangian method where the finite-element shape functions and the local element coordinate systems are used to describe the fluid-structure boundary without approximation. Time synchronization and spatial accuracy are main-

tained to ensure accurate exchange of energy between the fluid and the structure.

The computational solutions exhibit multiple types of aeroelastic response including transonic limit cycle flutter at a wide range of dynamic pressures, subsonic and supersonic bending-torsion flutter at higher dynamic pressures and a wide range of Mach numbers, and limit cycle oscillation dependent on both Mach number and angle of attack. Shock motion dependent on wing deformation is shown to play a major role in determining the response of the wings, and, depending on the flow conditions, can either stabilize or destabilize the response. Results from the simulations correlate closely with observed wind tunnel test responses.

The dissertation of Erik Charles Mellquist is approved.

William S. Klug

Ertugrul Taciroglu

Xiaolin Zhong

Oddvar O. Bendiksen, Committee Chair

University of California, Los Angeles

2014

To my Mom, my Dad, Emily, Jack, and Kate.

TABLE OF CONTENTS

List of Figures	x
List of Tables	xxi
Nomenclature	xxii
Symbols	xxiii
Abbreviations	xxviii
1 Introduction	1
1.1 Motivation for Research	1
1.2 Literature Review	3
1.2.1 Transonic Computational Aeroelasticity	3
1.2.2 Transonic Flutter of Cranked Wing Planforms	10
1.2.3 Mesh Generation Techniques	12
1.3 Problem Statement and Research Objectives	16
1.4 Dissertation Outline	17
2 Mixed Eulerian-Lagrangian Formulation of the Fluid Model .	19
2.1 Introduction	19
2.2 Governing Equations	20
2.3 Integration Scheme	23
2.4 Mesh Generation by the Advancing Front Method	24

2.4.1	Mesh Domain	26
2.4.2	Mesh Parameters	26
2.4.3	Background Mesh	29
2.4.4	Boundary Line Discretization	30
2.4.5	2-D Surface Discretization	32
2.4.6	Curved 2-D Surfaces	37
2.4.7	Wing Tip Discretization	39
2.4.8	The 3-D Mesh	40
2.5	Mesh Motion	40
2.6	Boundary Conditions	43
3	Nonlinear Structural Model	45
3.1	Introduction	45
3.2	Large Deformation and the Mixed Eulerian-Lagrangian Formulation	46
3.3	Components of the Structural Model	52
3.4	Finite Element Model of the Wing	56
4	Fluid-Structure Coupling by the Direct Eulerian-Lagrangian Ap- proach	59
4.1	Introduction	59
4.2	Direct Eulerian-Lagrangian Coupling	61
4.2.1	Boundary Location	62
4.2.2	Time Consistency	63

4.2.3	Consistent Load Vector	64
4.2.4	Conservation of Energy	65
4.3	Solution Procedure	67
4.3.1	Static Aeroelastic Solution Procedure	68
4.3.2	Dynamic Aeroelastic Solution Procedure	70
5	Aeroelastic Model Validation	73
5.1	Introduction	73
5.2	Linear Free Vibration Results	74
5.3	Steady CFD Results	77
5.4	Static Aeroelastic Results	83
5.5	Dynamic Aeroelastic Results	87
6	Aeroelastic Solutions for Cranked Wings	96
6.1	Introduction	96
6.2	The NEXST-2 Supersonic Transport Wing	97
6.2.1	NEXST-2 Wing Aeroelastic Model	99
6.2.2	NEXST-2 Wing Aeroelastic Results	100
6.3	Generic Cranked High Speed Wing (HC-Wing)	160
6.3.1	H-Wing Aeroelastic Model	162
6.3.2	H-Wing Aeroelastic Results	164
6.3.3	HC-Wing Aeroelastic Model	169

6.3.4	HC-Wing Aeroelastic Results	170
7	Conclusions and Recommendations for Future Research	189
7.1	Introduction	189
7.2	Conclusions	189
7.3	Recommendations for Future Research	193
A	dfLap User's Guide	194
B	Modifications to the Aeroelastic Code	225
B.1	Introduction	225
B.2	Cranked Wing Geometry	227
B.2.1	<i>G2dex</i> Updates for Cranked Wing Geometry	227
B.2.2	<i>Triang</i> Updates for Cranked Wing Geometry	229
B.2.3	<i>Steady</i> Updates for Cranked Wing Geometry	229
B.2.4	<i>Static</i> Updates for Cranked Wing Geometry	229
B.2.5	<i>Flutter</i> Updates for Cranked Wing Geometry	231
B.3	High Wing Sweep	231
B.4	Wing Tip Meshing	232
B.5	The Intel Fortran Compiler and LAPACK	234
B.6	Thickness and Density Distribution	235
B.7	Flow Properties	236
B.8	Structural Discretization	237

B.9	dfLap	237
B.10	Output Files	238
References	239

LIST OF FIGURES

2.1	Upper half of the mesh domain with the wing on the symmetry plane.	27
2.2	Wing region of the upper half of the mesh domain.	27
2.3	Wing planform parameters.	29
2.4	Background grid on the $Z = 0$ plane.	30
2.5	Background grid on the $Y = 0$ plane.	30
2.6	Background grid on the $X - Z$ plane at the wing tip.	31
2.7	Generation of a new point.	34
2.8	Symmetry plane mesh.	35
2.9	Wing surface mesh.	35
2.10	Wing tip mesh.	36
2.11	Tunnel attach wall mesh.	36
2.12	Tunnel far wall mesh.	36
2.13	Cylindrical far-field mesh.	37
2.14	Cylindrical far-field mesh mapped to two dimensions.	38
2.15	Wing surface mapped to two dimensions.	38
2.16	Wing tip mapped to two dimensions.	39
2.17	2-D mesh.	41
3.1	Aeroelastic wing cutaway courtesy of Reference 116.	46
3.2	Eulerian and Lagrangian coordinate systems.	47

3.3	Local in-plane deformation.	49
3.4	Local out-of-plane deformation.	50
3.5	Classical and mixed Eulerian-Lagrangian kinematics.	51
3.6	Components of the nonlinear structural finite element.	53
3.7	Partitions of an element stiffness matrix.	55
3.8	The (a) thickness distribution, (b) density distribution, and (c) finite element discretization of the NEXST-2 wing structural model.	57
4.1	The Direct Eulerian-Lagrangian approach to fluid-structure coupling.	61
4.2	Aerodynamic node coupled to structural finite element with loca- tion defined by area coordinates.	63
4.3	Two-dimensional representation of fluid-structure boundary defor- mation.	64
4.4	Three-dimensional representation of fluid-structure boundary de- formation.	65
4.5	Consistent load vector calculation.	66
4.6	Gaussian quadrature points and fluid elements.	67
4.7	Conservation of energy during flutter simulation of the NEXST-2 wing.	68
5.1	Structural finite element mesh for the ONERA M6 wing.	74
5.2	Structural thickness profiles.	75

5.3	First four normal modes of the ONERA M6 wing with structural configuration B.	77
5.4	ONERA M6 wing model symmetry plane mesh.	78
5.5	ONERA M6 wing surface mesh.	79
5.6	ONERA M6 wing steady pressure distribution at Mach 0.84, $\alpha = 3.06$ degrees.	81
5.7	ONERA M6 upper wing surface steady Mach distribution.	82
5.8	6x8x2 structural finite element mesh for the ONERA M6 wing.	84
5.9	Lift coefficient vs. span of the ONERA M6 wing.	86
5.10	Local angle of attack vs. span of the ONERA M6 wing.	86
5.11	Static aeroelastic deformation and Mach distribution of the low stiffness ONERA M6 wing.	87
5.12	Tip deflection time histories of the ONERA M6 wing, $h/c_{root} = 0.016$, Mach 0.84, $\alpha = 0$ degrees.	90
5.13	Total energy/aerodynamic work time histories of the ONERA M6 wing, $h/c_{root} = 0.016$, Mach 0.84, $\alpha = 0$ degrees.	90
5.14	Tip deflection time histories of the ONERA M6 wing, $h/c_{root} = 0.018$, Mach 0.84, $\alpha = 3.06$ degrees.	91
5.15	Total energy/aerodynamic work time histories of the ONERA M6 wing, $h/c_{root} = 0.018$, Mach 0.84, $\alpha = 3.06$ degrees.	92
5.16	Tip deflection time histories of the ONERA M6 wing, $h/c_{root} = 0.025$, Mach 0.84, $\alpha = 3.06$ degrees.	93

5.17	Total energy/aerodynamic work time histories of the ONERA M6 wing, $h/c_{root} = 0.025$, Mach 0.84, $\alpha = 3.06$ degrees.	94
6.1	Wind tunnel test model of the NEXST-2 SST wing courtesy of References 72, 73.	98
6.2	Wind tunnel test flutter boundary for the NEXST-2 wing from Reference 72.	98
6.3	Planform of the NEXST-2 wing model.	100
6.4	The (a) thickness distribution, (b) density distribution, and (c) finite element discretization of the NEXST-2 wing structural model.	101
6.5	First four modes of the NEXST-2 wing structural finite element model.	102
6.6	The aerodynamic mesh of the NEXST-2 wing model.	103
6.7	The flutter boundary from nonlinear simulations of the NEXST-2 wing.	104
6.8	Transonic flow over an airfoil.	107
6.9	The structural washout effect for swept wings.	107
6.10	Mach distribution at a range of steady conditions for the NEXST-2 wing.	109
6.11	NEXST-2 wing tip time histories in the transonic dip region at Mach 0.96, $q = 86.4$ kPa.	111
6.12	SRS of the NEXST-2 wing tip trailing edge at Mach 0.96, $q = 86.4$ kPa.	112

6.13	Modal kinetic energy of the NEXST-2 wing at Mach 0.96, $q = 86.4$ kPa.	113
6.14	Aerodynamic work and wing total energy of the NEXST-2 wing at Mach 0.96, $q = 86.4$ kPa.	114
6.15	Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.96, $q = 86.4$ kPa.	116
6.16	Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.96, $q = 86.4$ kPa.	117
6.17	Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.96, $q = 86.4$ kPa.	118
6.18	Wing deformation as viewed from beyond the NEXST-2 wing tip looking toward the root during a period of limit cycle flutter at Mach 0.96, $q = 86.4$ kPa.	119
6.19	NEXST-2 wing tip time histories in the transonic dip region at Mach 1.02, $q = 59.7$ kPa.	121
6.20	SRS of the NEXST-2 wing tip trailing edge at Mach 1.02, $q = 59.7$ kPa.	121
6.21	Modal kinetic energy of the NEXST-2 wing at Mach 1.02, $q = 59.7$ kPa.	122
6.22	Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.02, $q = 59.7$ kPa.	123
6.23	Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.02, $q = 59.7$ kPa.	124

6.24	Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.02, $q = 59.7$ kPa.	125
6.25	NEXST-2 wing deformation as viewed from beyond the wing tip looking toward the root during a period of limit cycle flutter at Mach 1.02, $q = 59.7$ kPa.	126
6.26	NEXST-2 wing tip time histories in the subsonic bending-torsion region at Mach 0.70, $q = 295.3$ kPa.	127
6.27	SRS of the NEXST-2 wing tip trailing edge at Mach 0.70, $q = 295.3$ kPa.	128
6.28	Modal kinetic energy of the NEXST-2 wing at Mach 0.70, $q = 295.3$ kPa.	128
6.29	Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.7, $q = 295.3$ kPa.	130
6.30	Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.7, $q = 295.3$ kPa.	131
6.31	Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.7, $q = 295.3$ kPa.	132
6.32	NEXST-2 wing deformation as viewed from beyond the wing tip looking toward the root during a period of limit cycle flutter at Mach 0.7, $q = 295.3$ kPa.	133
6.33	NEXST-2 wing tip time histories at Mach 0.85, $q = 223.9$ kPa. . .	134
6.34	Waterfall SRS of the NEXST-2 wing tip trailing edge at Mach 0.85, $q = 223.9$ kPa.	135

6.35	Modal kinetic energy of the NEXST-2 wing at Mach 0.85, $q = 223.9$ kPa.	135
6.36	Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.85, $q = 223.9$ kPa.	138
6.37	Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.85, $q = 223.9$ kPa.	139
6.38	Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.85, $q = 223.9$ kPa.	140
6.39	NEXST-2 wing deformation as viewed from beyond the wing tip looking toward the root during a period of limit cycle flutter at Mach 0.85, $q = 223.9$ kPa.	141
6.40	NEXST-2 wing tip time histories in the supersonic bending-torsion region at Mach 1.15, $q = 288.4$ kPa.	142
6.41	Modal kinetic energy of the NEXST-2 wing at Mach 1.15, $q = 288.4$ kPa.	143
6.42	SRS of the NEXST-2 wing tip trailing edge at Mach 1.15, $q = 288.4$ kPa.	143
6.43	Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.15, $q = 288.4$ kPa.	144
6.44	Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.15, $q = 288.4$ kPa.	145
6.45	Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.15, $q = 288.4$ kPa.	146

6.46	NEXST-2 wing deformation as viewed from beyond the wing tip looking toward the root during a period of limit cycle flutter at Mach 1.15, $q = 288.4$ kPa.	147
6.47	NEXST-2 wing tip time histories at Mach 1.04, $q = 136.6$ kPa. . .	148
6.48	Waterfall SRS of the NEXST-2 wing tip trailing edge at Mach 1.04, $q = 136.6$ kPa.	149
6.49	Modal kinetic energy of the NEXST-2 wing at Mach 1.04, $q = 136.6$ kPa.	149
6.50	Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.04, $q = 136.6$ kPa.	151
6.51	Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.04, $q = 136.6$ kPa.	152
6.52	Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.04, $q = 136.6$ kPa.	153
6.53	NEXST-2 wing deformation as viewed from beyond the wing tip looking toward the root during a period of limit cycle flutter at Mach 1.04, $q = 136.6$ kPa.	154
6.54	NEXST-2 wing tip time histories in the high frequency LCO region at Mach 0.75, $q = 129.1$ kPa.	155
6.55	Waterfall SRS of the NEXST-2 wing tip trailing edge at Mach 0.75, $q = 129.1$ kPa.	156
6.56	Modal kinetic energy of the NEXST-2 wing at Mach 0.75, $q = 129.1$ kPa.	156

6.57	NEXST-2 wing tip time histories of a damped response at Mach 0.85, $q = 107.8$ kPa.	158
6.58	Mach distribution on the NEXST-2 wing during the second cycle of the simulation at Mach 0.85, $q = 107.8$ kPa.	159
6.59	NEXST-2 wing tip time histories of a damped response at Mach 1.06, $q = 103.2$ kPa.	160
6.60	Mach distribution on the NEXST-2 wing during the second cycle of the simulation at Mach 1.06, $q = 103.2$ kPa.	161
6.61	Planform and fluid mesh of the H-Wing.	163
6.62	The structural finite element discretization of the H-Wing.	164
6.63	The first four normal modes of the H-Wing.	164
6.64	LCO amplitude of the tip angle of attack vs. Mach number for the H-Wing at $\alpha = 7.5$ degrees.	166
6.65	LCO amplitude of the tip angle of attack vs. root angle of attack for the H-Wing at Mach 0.99.	167
6.66	Wing tip time histories of the thick H-Wing at Mach 0.99, $\alpha = 8.0$ degrees.	167
6.67	Modal kinetic energy of the thick H-Wing at Mach 0.99, $\alpha = 8.0$ degrees.	168
6.68	SRS of the thick H-Wing at Mach 0.99, $\alpha = 8.0$ degrees.	168
6.69	Planform of the HC-Wing model.	169
6.70	Fluid mesh of the HC-Wing model.	169

6.71	Structural model of the HC-Wing with flexible inboard section (a) finite element discretization and (b) thickness distribution.	171
6.72	The first four normal modes of the HC-Wing with flexible inboard section.	171
6.73	LCO amplitude of the tip angle of attack vs. root angle of attack for the HC-Wing at Mach 0.99.	173
6.74	LCO amplitude of the tip angle of attack vs. Mach number for the HC-Wing at $\alpha = 7.5$ degrees.	173
6.75	LCO amplitude of the tip angle of attack vs. Mach number for the thin H-Wing and thin HC-Wing at $\alpha = 7.5$ degrees.	174
6.76	LCO amplitude of the tip angle of attack vs. root angle of attack for the thin H-Wing and thin HC-Wing at Mach 0.99.	174
6.77	LCO amplitude of the tip angle of attack vs. Mach number for the thick H-Wing and thick HC-Wing at $\alpha = 7.5$ degrees.	175
6.78	LCO amplitude of the tip angle of attack vs. root angle of attack for the thick H-Wing and thick HC-Wing at Mach 0.99.	175
6.79	Wing tip time histories of the thick HC-Wing at Mach 1.01, $\alpha = 7.5$ degrees.	176
6.80	SRS of the thick HC-Wing at Mach 1.01, $\alpha = 7.5$ degrees	176
6.81	Modal kinetic energy of the thick HC-Wing at Mach 1.01, $\alpha = 7.5$ degrees.	177
6.82	Mach distribution on the thin H-Wing upper surface at a range of free stream Mach numbers for $\alpha = 7.5$ degrees.	178

6.83	Mach distribution on the thin H-Wing upper surface at a range of root angles of attack for a free stream Mach number of 0.99. . . .	179
6.84	Mach distribution on the thin HC-Wing upper surface at a range of free stream Mach numbers for $\alpha = 7.5$ degrees.	180
6.85	Mach distribution on the thin HC-Wing upper surface at a range of root angles of attack for a free stream Mach number of 0.99. . .	181
6.86	Mach distribution on the thick H-Wing during a period of limit cycle flutter at Mach 0.99, $\alpha = 8.0$ degrees.	184
6.87	Aerodynamic power distribution on the thick H-Wing during a pe- riod of limit cycle flutter at Mach 0.99, $\alpha = 8.0$ degrees.	185
6.88	Mach distribution on the thick H-Wing during a period of damped oscillation at Mach 0.99, $\alpha = 3.0$ degrees.	186
6.89	Mach distribution on the flexible inboard HC-Wing upper surface over a range of free stream Mach numbers for $\alpha = 7.5$ degrees. . .	187
6.90	Mach distribution on the flexible inboard HC-Wing upper surface for a range of root angles of attack for a free stream Mach number of 0.99.	188
B.1	Wing surface with distorted elements.	232
B.2	Wing surface after code update.	233
B.3	Comparison of wing structural frequencies.	236

LIST OF TABLES

2.1	Wing planform parameters.	28
2.2	Aerodynamic mesh surface definitions.	33
5.1	Natural frequencies of the ONERA M6 wing model.	76
5.2	Mesh details for the ONERA M6 wing model.	79
5.3	C_L and C_D for the rigid ONERA M6 wing.	82
5.4	Tip deflection for the flexible ONERA M6 wing.	85
5.5	C_L for the flexible ONERA M6 wing.	85
6.1	Natural frequencies of the NEXST-2 wing structural finite element model.	102
6.2	Mesh details for the NEXST-2 wing model.	103
6.3	Mesh details for the H-Wing model.	163
6.4	Natural frequencies of the H-Wing structural finite element model.	165
6.5	Mesh details for the HC-Wing model.	170
6.6	Natural frequencies of the HC-Wing with flexible inboard section.	172

NOMENCLATURE

Symbols

Ω	=	Fluid control volume
$\partial\Omega$	=	Boundary of a fluid control volume
\mathbf{U}	=	Velocity of the boundary of a fluid control volume with components U_1 , U_2 , and U_3
\mathbf{W}	=	Vector of conserved fluid quantities
ρ	=	Fluid density
\mathbf{u}	=	Fluid velocity with components u_1 , u_2 , and u_3
e	=	Fluid energy per unit mass
σ_{ij}	=	Fluid Cartesian stress tensor
p	=	Fluid pressure
γ	=	Ratio of specific heats of the fluid
\mathbf{W}_j	=	Nodal values of \mathbf{W} in the discretized fluid system
ϕ_j	=	Linear shape functions of the discretized fluid system
\mathbf{m}_{ij}	=	Consistent mass matrix of the discretized fluid system
\mathbf{Q}_i	=	Fluid flux vector
\mathbf{D}_i	=	Fluid dissipative flux vector
α_i	=	Runge-Kutta coefficients
Δt	=	Runge-Kutta integration time step
λ	=	Courant-Friederich-Lewy (CFL) number
Λ_{L1}	=	Wing inboard leading edge sweep angle

Λ_{T1}	=	Wing inboard trailing edge sweep angle
Λ_{K1}	=	Wing kink sweep angle
Y_{L1}	=	Wing span location of kink at leading edge
Λ_{L2}	=	Wing outboard leading edge sweep angle
Λ_{T2}	=	Wing outboard trailing edge sweep angle
Λ_{K2}	=	Wing tip sweep angle
Y_{L2}	=	Wing span location of tip at leading edge
T_{span}	=	Wing thickness along its span relative to root
Y_{wall}	=	Far tunnel wall location
δ	=	Local fluid mesh size
A_s	=	Number of mesh segments on a boundary line
$\delta_c(s)$	=	Function describing the fluid mesh size along a boundary line
l_i	=	Location of grid points along a boundary line
a_i	=	Candidate list of points for the advancing front method
k_m	=	Spring stiffness for a given edge of the aerodynamic mesh
$\delta_{x_i}^{(k)}, \delta_{y_i}^{(k)}, \delta_{z_i}^{(k)}$	=	Displacement of a mesh node at Runge-Kutta stage k
$x_i^{(k)}, y_i^{(k)}, z_i^{(k)}$	=	Location of a mesh node at Runge-Kutta stage k
\mathbf{n}	=	Normal vector of the tunnel wall or wing surface
R	=	Riemann invariant
\mathbf{R}_{ij}	=	Vector from node i to node j of a structural finite element in the global (Eulerian) coordinate system
x, y, z	=	Coordinates of the global (Eulerian) coordinate system
x', y', z'	=	Coordinates of the local (Lagrangian) coordinate system
$[e]$	=	Rotation matrix from global to local coordinate system

\mathbf{q}'_{uv}	=	Local structural in-plane degrees of freedom
\mathbf{q}'_w	=	Local structural out-of-plane degrees of freedom
u', v', w'	=	Local structural displacements in the x', y', z' directions
$\beta_{x'}, \beta_{y'}, \beta_{z'}$	=	Local structural rotations about the x', y', z' directions
U_{total}^e	=	Total strain energy of a structural finite element
U_m	=	Strain energy due to membrane stretching
U_b	=	Strain energy due to bending
U_s	=	Strain energy due to shear
U_{N1}	=	Strain energy due to nonlinear coupling of in-plane and out-of-plane deformation
U_{N2}	=	Strain energy due to nonlinear coupling of slope due to large deflections
$[D_m]$	=	Membrane stress-strain constitutive matrix
$[D_b]$	=	Flexural rigidity matrix
$[D_s]$	=	Shear rigidity matrix
E	=	Young's modulus of elasticity
ν	=	Poisson's ratio
h	=	Plate thickness
k	=	Shear correction factor
G	=	Shear modulus of elasticity
$\{\varepsilon_m\}$	=	Membrane strain vector
$\{\gamma\}$	=	Shear strain vector
$\{\varepsilon_n\}$	=	Nonlinear von Karman strain vector
$\{\kappa\}$	=	Plate element curvature vector

u'_0, v'_0, w'	=	Local structural displacements of the middle surface
$[K_e]$	=	Element stiffness matrix
$[K_t]$	=	Element tangent stiffness matrix
$[K_s]$	=	Element secant stiffness matrix
N_{xe}	=	Number of chordwise finite elements
N_{ye1}	=	Number of inboard spanwise finite elements
N_{ye2}	=	Number of outboard spanwise finite elements
$\{F_{aero}\}$	=	Global aerodynamic load vector
$\{q_t\}$	=	Linear displacement vector
$[M]$	=	Structural system global mass matrix
$[K]$	=	Structural system global linear stiffness matrix
$\{F\}$	=	Global external force vector
$\{F_{elastic}\}$	=	Global elastic force vector
ρ_p	=	Structural plate density
η	=	Wing span location
α	=	Wing root angle of attack
M_∞	=	Mach number
ρ_∞	=	Air density
a_∞	=	Speed of sound
C_l, C_d	=	Sectional lift and drag coefficients
C_L, C_D	=	Total wing lift and drag coefficients
c_{root}	=	Wing root chord
τ	=	Nondimensional time, $\omega_{1T}t/2\pi$
ω_{1B}	=	Wing first bending mode frequency

ω_{1T}	=	Wing first torsion mode frequency
t	=	Time
$\{q\}$	=	System modal response (displacement, velocity, or acceleration)
$[\phi]$	=	System mass orthonormalized modes
$\{x\}$	=	System physical response (displacement, velocity, or acceleration)
KE_i	=	Kinetic energy of mode i

Superscripts and Subscripts

(k)	=	Runge-Kutta stage k
$[n]$	=	Global time step number n
$2D$	=	Flat representation of a mesh surface
$3D$	=	Curved representation of a mesh surface
L	=	Local (Lagrangian) coordinate system
G	=	Global (Eulerian) coordinate system
d	=	Deformed vector or coordinate
t^n	=	Vector at time t^n

Abbreviations

AFM	=	Advancing front method
ALE	=	Arbitrary Lagrangian-Eulerian
CFD	=	Computational fluid dynamics
CFL	=	Courant-Friedrichs-Lewy
CST	=	Constant strain triangle
dfLap	=	Direct Flutter Analysis Program
DST	=	Discrete shear triangle
ESDOF	=	Essentially single degree of freedom
FE	=	Finite element
GUI	=	Graphical user interface
HSCT	=	High Speed Civil Transport
IMSL	=	International Mathematics and Statistics Library
IMKL	=	Intel Math Kernel Library
JAXA	=	Japanese Aerospace Exploration Agency
LAPACK	=	Linear Algebra Package
LCO	=	Limit cycle oscillation
SRS	=	Shock response spectrum
SST	=	Supersonic transport

ACKNOWLEDGMENTS

This work would not have been possible without the contributions of many people to whom I am extremely grateful.

First and foremost, I would like to express my sincere gratitude and appreciation to Professor Bendiksen for all he has done to bring this work to completion. It has been a great privilege to study computational aeroelasticity with one of the leading experts in the field.

I would also like to thank the members of my committee, Professors Zhong, Tacioglu, and Klug, for reading and commenting on my dissertation.

Throughout this process, I have been fortunate to have the support of The Aerospace Corporation. I am thankful to Linda Yocum, Fran Gillis, and Cynthia Dickson for ensuring my tuition was always paid. I am also grateful to my management, Jeff Lollock, Herand Bedrossian, Alvar Kabe, Jeff Emdee, and Chuck Gustafson, who have been beyond flexible, understanding, and supportive.

I would like to thank my colleagues, Earl Parker, Ryan Tuttle, Wing Yeung, Anne Le, Nathan Kinkaid, and Jim Gariffo who have helped me in countless ways.

I am forever indebted to my family. My mom and dad, Darlene and Robert Mellquist, constantly stressed the value of education and hard work, and I cannot thank them enough. My son, Jack, always understood when I had to work and never complained when playtime was cut short. Last, and most important, my wife, Emily, who stood by me through the entire process. I would have given up years ago without her support. This accomplishment is as much hers as it is mine.

VITA

- 2002 B.S. (Aerospace Engineering), The University of Michigan, Ann Arbor, Michigan.
- 2003 M.S. (Aerospace Engineering), The University of Michigan, Ann Arbor, Michigan.
- 2003–present Engineering Specialist, Structural Dynamics Department, The Aerospace Corporation, El Segundo, California.

PUBLICATIONS

Mellquist, E., Waas, A. “Size Effects in the Compressive Crushing of Circular Cell Honeycombs,” *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Denver, Colorado, April 2002.

Mellquist, E., Waas, A. “Size Effects in the Crushing of Honeycomb Structures,” *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Palm Springs, California, April, 2004.

Mellquist, E., Bendiksen, O. “Computational Transonic Flutter Solutions for Cranked Wing Planforms,” *55th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, National Harbor, Maryland, January, 2014.

CHAPTER 1

Introduction

1.1 Motivation for Research

Accurate solutions to transonic aeroelasticity problems are important for the design of modern commercial and military aircraft [1]. These vehicles are required to be lightweight and efficient while operating at transonic Mach numbers. The structures of lighter aircraft are more flexible and therefore susceptible to large, nonlinear deformation. The transonic aerodynamic environment is inherently nonlinear and can impart large, unsteady forces on the aircraft as regions of supersonic flow and shock waves form, move, and disappear on the vehicle surface [2–4]. The flutter boundary at transonic Mach numbers often resides at lower dynamic pressures than it does in the adjacent subsonic and supersonic regions, making its identification critical [5]. Additionally, experimental results have proven difficult with the potential for unexpected model destruction and wind tunnel damage [6]. Due to the nonlinear nature of transonic flutter, obtaining solutions that accurately model and explain the physical phenomena has been difficult, and numerical simulations are often required.

Three-dimensional computational aeroelastic solutions based on the nonlinear Euler [7] and Navier-Stokes [8] equations were first calculated over twenty years

ago. Since that time, most solutions have focused on improving the fluid and/or structural models used in the simulation, and rightfully so, as their accuracy is of the utmost importance. In contrast, the coupling of the two models has often been treated as if its importance was secondary. This is a problem, as it has been shown that the accuracy of the coupling between the fluid and the structure can have a large effect on the solution [9–11]. The Direct Eulerian-Lagrangian coupling method has proven to be accurate at simulating the energy exchange between the fluid and the structure [9], thus ensuring that the conservation laws are satisfied. Up to this point, the Direct Eulerian-Lagrangian method has primarily been applied to simple wing planforms with low to moderate leading edge sweep. Wider application of the method has the potential to improve understanding of transonic aeroelasticity phenomena.

In the present research, a three-dimensional computational aeroelastic simulation for cranked, highly-swept wings is developed, and solutions are presented for several wing models. The computational model is a fully nonlinear coupled fluid-structure simulation based on the Direct Eulerian-Lagrangian coupling methodology. The wing is modeled using nonlinear modified von Karman plate finite elements. Large deformation is accounted for through the use of element-attached local coordinate systems referenced to a single global coordinate system. The fluid is modeled using the mixed Eulerian-Lagrangian formulation of the classical Euler equations and is discretized using a Galerkin finite element approach on an unstructured tetrahedral mesh. The fluid and structural models are coupled by the Direct Eulerian-Lagrangian method where the finite-element shape functions and the local element coordinate systems are used to describe the fluid-structure

boundary without approximation. Time synchronization and spatial accuracy are maintained to ensure accurate exchange of energy between the fluid and the structure.

1.2 Literature Review

1.2.1 Transonic Computational Aeroelasticity

Early developments in transonic computational aeroelasticity were based on CFD research from the 1970's. In 1971, Murman and Cole [12] developed a two-dimensional steady simulation based on transonic small disturbance theory. Their work was extended to three-dimensions by Ballhaus and Bailey [13]. Later the same method would be applied to solve the two-dimensional [14] and three-dimensional [15] unsteady equations.

The first computational transonic flutter analyses applied transonic small disturbance theory to simple two and three-dimensional models. Two-dimensional solutions were computed by Ballhaus and Goorjian [16] and Rizzetta [17] in the late 1970's. Soon after, Borland and Rizzetta extended their previous work [15] to a three-dimensional flutter solution [18] which coupled a transonic small disturbance theory fluid model to a normal modes structural model. The fluid and structural equations were integrated forward in time together. The code, called XTRAN3S, was limited to wings with high aspect ratios, large taper ratios, and small sweep angles because of the coordinate transformation used. Later, Guruswamy and Goorjian would develop an improved coordinate transformation and implement it in a new program called ATRAN3S [19]. The improvement allowed

for aeroelastic analysis of more complicated wing planforms including those of the F5 and B-1 [20]. ATRAN3S and other aeroelastic analysis programs based on transonic small disturbance theory and the full potential equations were the state of the art through the end of the 1980's [7, 21–23]. Despite their popularity, transonic flutter solutions calculated with these programs were of questionable validity. In potential flow theory, entropy production is neglected. This results in a poor prediction of the strength and location of shocks which are important factors in transonic aeroelastic stability. In addition, the small disturbance assumption is likely invalid for aeroelastic analysis where deformation can be large.

As computing speed increased, it became feasible to simulate unsteady transonic flows with the Euler or Navier-Stokes equations. In 1985, Venkatakrishnan and Jameson [24] developed an unsteady Euler solver for the problem of two-dimensional transonic flow over a moving airfoil. A few years later, the first instance of a two-dimensional Euler CFD based aeroelastic analysis was presented by Bendiksen and Kousen [25, 26]. They used a finite volume scheme on a moving, structured mesh, integrating the structural and fluid equations together in time. Aerodynamic forces were applied to a typical section model of an airfoil by enforcement of tangent flow boundary conditions. Around the same time, Wu et al. [27] and Wu [28] presented a method for solving the two-dimensional unsteady compressible Navier-Stokes equations along with flutter solutions for a typical section model. Soon, Guruswamy would use the new ENSAERO code to extend the application of the Euler [29] and Navier-Stokes [8, 30–32] equations to the three-dimensional unsteady aerodynamic and flutter analyses of wings modeled using plate finite elements. Another three-dimensional Euler/Navier-Stokes aeroelastic

code called CFL3D was also in development at the time. This code was applied to many problems including flutter solutions for a two-dimensional airfoil and three-dimensional 45 degree swept wing using a structured, deforming mesh with the Euler [33, 34] and Navier-Stokes [35] equations. The mesh motion algorithm of CFL3D was based on the work of Batina [36], who used a spring analogy to determine the location of the interior points of a two-dimensional unstructured, triangular mesh as the airfoil moved.

Up to this point, all existing aeroelastic simulations were using what is now referred to as the “classical approach” to fluid-structure coupling. In these simulations, the fluid and structure are integrated separately and coupled through the kinematic boundary conditions. Time consistency of the classical methods is difficult to establish, as enforcement of the boundary conditions requires knowledge of the position and velocity of the fluid-structure boundary, but the position and velocity of the boundary are not known until the boundary conditions are used to calculate the surface pressures and apply forces to the structure for the current time step. To manage this problem, approximations are used. Usually the pressure from the previous time step is extrapolated forward in time and applied to the system. This process introduces unknown phase errors. Additionally, the location of the fluid-structure boundary is often estimated with a “virtual” boundary surface using spline or interpolation functions. The estimated boundary may not be accurate, leading to errors in the calculated energy transferred from the fluid to the structure.

In 1991, Bendiksen presented a new computational method for aeroelastic response calculations that solved the problems inherent in the classical approach

[9]. By formulating the governing equations for both the fluid and the structure in integral conservation law form based on a mixed Eulerian-Lagrangian description, the fluid-structure system was treated as a single continuum dynamics problem. In this initial paper and a follow up [4], solutions were calculated for a two-dimensional wing section with camber bending. The fluid was modeled using the Euler equations discretized with a structured quadrilateral finite volume mesh, and the structure was modeled with beam finite elements. Results showed that the new method accurately modeled the transfer of energy between the fluid and the structure. The total energy of the structure (kinetic plus strain) was compared to the work done by the fluid for every time point in the simulation. The difference between the two was constant and equal to the initial energy of the structure. In the absence of structural damping, this indicates that the law of conservation of energy was satisfied. When the same calculations were done on the classical model from Reference 25, the difference between the energy and work changed over time indicating that energy was not being conserved. Flutter response comparisons between the new and classical method showed significant differences, indicating the importance of proper fluid-structure coupling in aeroelastic simulations.

Over the past twenty years, the new method (eventually dubbed the Direct Eulerian-Lagrangian method [37]) has been applied to multiple aeroelastic and aeroservoelastic problems resulting in improved understanding of aeroelastic responses in the transonic region. Some of the applications of the Direct Eulerian-Lagrangian method are listed below.

1. The importance of shock motion in transonic flutter was demonstrated using a two-dimensional wing section with camber bending on a structured,

quadrilateral, finite volume mesh [4].

2. The aeroelastic response of transonic and supersonic flows over two-dimensional panels was calculated using a structured triangular finite element discretization of the Euler equations. Beam finite elements were used to model the panel. Divergence and limit cycle flutter were observed in the solutions [38, 39].
3. Hsiao and Bendiksen [40–42] simulated the transonic aeroelastic stability of cascades by discretizing the Euler equations over a two-dimensional hybrid structured/unstructured triangular finite element mesh.
4. Aeroservoelastic response of the ONERA M6 wing [43] (with 30 degree leading edge sweep) was studied. For the first time, the Direct Eulerian-Lagrangian method was applied in three-dimensions. The simulations used an unstructured tetrahedral mesh to discretize the Euler equations. The wing structure was modeled with Mindlin-Reissner discrete shear triangle plate finite elements. To improve computational performance, the simulations were run on a parallel computer system [37, 44–47].
5. Limit cycle flutter behavior of a three-dimensional wing in transonic flow was studied [48]. Nonlinear shock transitions from Tijdeman Type A (continuous) to Type B (intermittent) [3] were identified as the dominant mechanism for limiting the energy exchange from the fluid to the structure, resulting in limit cycle oscillations (LCO) as opposed to destructive flutter.
6. Bendiksen and Seber studied the effect of nonlinear structural stiffening on the ONERA M6 wing during transonic flutter. The simulation also included

geometric nonlinearity which allowed aerodynamic loads to be modeled as follower forces. [49–51].

7. Bendiksen studied the transonic limit cycle flutter response of a high aspect ratio swept wing. The improved, nonlinear, structural model was included. The new computational results agreed well with existing experimental results [52]. Because of the accurate fluid-structure coupling provided by the Direct Eulerian-Lagrangian method, the energy exchange between the fluid and the structure at different span locations could be studied. Several new observations regarding transonic flutter were made including the effect of shock location on flutter behavior, the stabilizing nature of structural washout, and the “aeroelastic stability balance” based on differences in aerodynamic work on the inboard and outboard portions of the wing. [53–55].
8. A phenomenon, “Mach number freeze,” which had not previously been identified in unsteady transonic flows, was studied. Its existence was used to explain the stabilization of limit cycle flutter near Mach one and the destabilizing effect of wing sweep at transonic Mach numbers [56].

The Direct Eulerian-Lagrangian method has primarily been applied to simple wing planforms with low to moderate sweep, while more complicated wing planforms have only been studied using less rigorous coupling methods.

Baker et al. [57] simulated the response of a cranked high speed transport wing using a test correlated, linear finite element model coupled to an Euler CFD solver. The simulation predicted unstable flutter in a narrow Mach range (0.98–1.0) at a wide range of dynamic pressures. At the same Mach numbers and dynamic

pressures, the test model [6, 58, 59] experienced “increased response.” Another “region of increased response” from the test (Mach 0.8–0.97) was not predicted by the simulation.

Gordnier and Melville [60–62], Attar and Gordnier [63, 64], and Peng and Han [65] developed a simulation of a cropped delta wing undergoing LCO in transonic flow. Multiple coupling methods and structural models were combined with both Euler and Navier-Stokes fluid models. The simulations were able to predict the onset of the test measured flutter boundary [66], but unable to simulate the correct LCO magnitude.

In 2001, Hartwich et al. [67] attempted to simulate the limit cycle flutter of the B-1 wing that had previously been observed in flight [68] and wind tunnel test [69]. The computations used the CFL3D Navier-Stokes fluid solver loosely coupled to a linear, modal finite element model. The simulation was only able to produce limit cycle flutter when a single structural mode was used. The magnitude of response was significantly different from measured test responses.

The costs associated with using classical coupling methods are significant. Not only is the accuracy of the simulations reduced, but the knowledge which can be gained by studying the results is limited. When conservation of energy cannot be guaranteed (as it cannot with classical coupling), the transfer of energy between the fluid and the structure can be incorrect. Because of this, it is difficult to improve the understanding of the physics behind the behavior when the results are studied.

1.2.2 Transonic Flutter of Cranked Wing Planforms

Transonic flutter of cranked wings has been observed in multiple flight and wind tunnel tests. In 1978, transonic limit cycle flutter was observed during flight testing of the variable sweep wing B-1 aircraft [68]. It was initially posited that the oscillations were due to shock motion on the wing. Several years later two dynamically scaled aeroelastic models of the B-1 wing were tested in the transonic wind tunnel at NASA Ames [69]. Limit cycle oscillations similar to those in the flight test were observed for the fully swept (65 and 67.5 degrees) configuration. The LCO was seen at Mach numbers from 0.65 to 1.05 and was highly sensitive to angle of attack, only occurring above 6.9 degrees. The researchers involved discounted the prior assertion that shock motion was responsible for the oscillations. Instead, they argued that a leading edge vortex was transferring energy to the wing in the first bending mode.

Multiple attempts at modeling the phenomenon causing the LCO are recorded in the literature. Guruswamy et al. [20] coupled a transonic small disturbance theory fluid model to a normal modes structural model, but simulations were not able to capture any of the limit cycle behavior that had been observed. Later, Guruswamy claimed to simulate sustained aeroelastic oscillations using a three-dimensional finite element structural model coupled to the ENSAERO Navier-Stokes fluid model, but the results show lightly damped oscillations [8]. More recently, Hartwich et al. used the CFL3D Navier-Stokes fluid solver loosely coupled to a linear, modal finite element model [67]. The simulation was only able to produce limit cycle flutter when a single structural mode was used. The magnitude of response was significantly different from that measured in the wind

tunnel test. In response to Reference 67, Ericsson suggested the causative phenomenon was not a single leading-edge vortex, but rather the interaction between two leading-edge vortices [70, 71].

A second example of a cranked wing experiencing flutter was documented by Silva et al. [6] and Schuster et al. [58, 59]. A series of tests were run for a High Speed Civil Transport (HSCT) wing in NASA Langley’s Transonic Dynamics Tunnel. Both rigid and flexible models were tested. The flexible model was aeroelastically scaled, and built using fiberglass skins with an open-cell foam core. Due to mistakes in the design of the model, it had a flutter dynamic pressure that was beyond the safe limit of the tunnel. To change the structural dynamic properties of the wing, a mass was added to the aft tip section. This brought the flutter dynamic pressure within the range of the tunnel. During testing, a flutter point was initially found, but the response was not repeatable. It was believed that non-elastic deformation of the wing (e.g. loosened rivets) was responsible. During the test, two regions of “increased response in first wing bending mode” were discovered in the transonic regime (Mach 0.8–1.0). Not much detail is given on the nature of the increased response, but it is never identified as unstable flutter. One of the increased response regions, identified as the “chimney,” was only present from Mach 0.98–1.0, but existed at a large range of dynamic pressures. At a high dynamic pressure point in the chimney, unstable flutter was encountered, and the model was destroyed.

Baker et al. [57] attempted to simulate the response of the test wing using a correlated, linear finite element model coupled to an Euler CFD solver. The simulation predicted flutter in the same “chimney” region as the test. There is

not sufficient detail in the literature to determine how accurately the simulation was able to model the full set of wind tunnel observations.

More recently, the NEXST-2 supersonic transport (SST) model, which is similar to NASA's HSCT, was analyzed by the Japan Aerospace Exploration Agency (JAXA) in their Transonic Flutter Wind Tunnel [72, 73]. They insured against the loss of their model by installing a weight on the trailing edge at the wing tip that would detach during large oscillations, changing the structural properties of the model and arresting the flutter response. Because of this, they were able to find a high dynamic pressure flutter region that existed between Mach 0.9 and Mach 1.1 without destroying their model. At a narrower range of Mach numbers (0.96–1.06), but lower dynamic pressures, they discovered a region of stable limit cycle oscillations (LCO). The dynamic responses in the two regions were at unique frequencies, indicating that different modes of the structure were being excited in each case.

1.2.3 Mesh Generation Techniques

The first step in solving any CFD problem is to generate a mesh to discretize the fluid domain. The mesh type and quality greatly influence the efficiency and accuracy of the solution. Meshes are often divided into two categories: structured and unstructured.

The traditional approach has been to use structured meshes to discretize the domain. These grids often employ quadrilaterals in two-dimensions and hexahedra in three-dimensions spaced in regular patterns. Structured meshes have a high degree of regularity and smoothness which lends itself to straightforward

grid generation and a more efficient solution processes. Structured meshes also have the advantage of being well-suited for solving the full Navier-Stokes equations, where, in order to resolve boundary layer characteristics, the element size in the direction perpendicular to the boundary should be small. It is often orders of magnitude smaller than the element size in the flow direction. Structured meshes, however, suffer from the limitation of hexahedral elements to discretize complicated domains [74].

In contrast to structured meshes, unstructured meshes, which often consist of simplicial elements (triangles in two-dimensions and tetrahedra in three-dimensions), excel at discretizing complex geometries. Unstructured meshes are also ideal for the implementation of adaptive meshing techniques where the mesh moves to accommodate a changing boundary, grid points are added to improve accuracy, or grid points are removed to improve efficiency. To their detriment, solving the conservation equations on unstructured meshes is not as computationally efficient as it is on structured grids. In addition, the process of generating an unstructured mesh can be challenging. In response to these challenges, there has been a considerable amount of work in the field of unstructured mesh generation [74–76].

Unstructured mesh generation techniques can be divided into two categories. The first class of methods are those that work by modifying an existing mesh or connecting an existing set of grid points. These methods often use Delaunay triangulation techniques (also identified as Dirichlet or Voronoi tessellation). In the second category, are the advancing front techniques which separate the meshed region from the unmeshed region with a “front.” This front is then advanced into the unmeshed space through the addition of points, edges, and elements [75].

One of the first methods for generating unstructured meshes was developed by Frederick et al. [77] for use in two-dimensional structural finite element stress analysis. In this work, triangular elements were built on a set of points that were manually entered using a magnetic pen on an electromagnetic graph tracing table. The algorithm was dependent on proper numbering of the nodes, and thus somewhat inefficient. Nguyen [78] used a similar algorithm (though without the magnetic pen technology) to generate tetrahedral meshes in three-dimensions with automated mesh refinement.

The first general purpose algorithm for Delaunay triangulation in two-dimensions was developed by Green and Sibson [79] who used Dirichlet tessellations to divide a plane into contiguous regions which were then used to define a triangulation. Bowyer [80] and Watson [81] expanded upon this work by developing algorithms to calculate the Dirichlet tessellation and Delaunay triangulation in Euclidean space of any dimension. The method developed by Watson introduced new points to the triangulation one at a time (though all the points were defined prior to the start of mesh generation) and was the first use of an advancing front technique. The efficiency of Watson's algorithm [81] was improved for two-dimensional discretization by Sloan and Houlsby [82] through the use of improved data storage routines.

Though these methods produced fully meshed regions, setting them up required a large amount of effort. With the goal of reducing the time required to set up a mesh, Lo [83] used an improved algorithm for generating the node locations to produce a two-dimensional grid of triangles that could discretize any arbitrary multi-connected domain. The algorithm allowed for the definition of

sub-domains with independent mesh sizes. This resulted in good mesh resolution in regions with irregular boundaries. Like the work of Watson, Lo used an advancing front technique over a region where grid points (but not the connectivity) were previously defined.

Jameson et al. [84] were the first to use Delaunay triangulation to generate a mesh for the solution of aerodynamic equations. Bowyer's method [80] was applied in three-dimensions to produce a mesh around a complete aircraft for the solution of steady inviscid transonic flow using the finite element method with Euler equations.

The first instance of an advancing front technique which introduced nodes and elements together was developed by Peraire et al. [85]. They used a coarse two-dimensional triangular background grid to define the mesh size over the computational domain. The front was then advanced from the outer edges inward until the domain was filled. In addition, a directional mesh refinement scheme was developed which used error in the solution to identify regions and directions where the mesh should be improved. The initial mesh was then used as a background mesh to generate the refined mesh. The work of Peraire et al. was extended to three-dimensions by Löhner and Parikh [86] and Löhner [87] who derived an efficient algorithm by using a heap list to store the faces (triangles in three-dimensions) and quad/octree structures to store the points. In contrast to previous methods, the advancing front method was more automatic with less user involvement. The method became popular and was implemented by NASA [88] in their GEN2D and GEN3D programs. Later, Peraire et al. [89], added progressive mesh refinement to a three-dimensional implementation of their program.

As CFD mesh sizes increased, more efficient mesh creation programs became necessary. In response, Löhner et al. [90] developed a parallel implementation of the advancing front method in two-dimensions. The domain was divided into sub-domains which were meshed independently and then combined. The algorithm was programmed to mesh the sub-domains in parallel on a computer with multiple processors, and a significant improvement in meshing time was realized.

To this point, unstructured grids were being used to solve the Euler equations, but an unstructured mesh scheme appropriate for the full Navier-Stokes equations had not yet been found. Kallinderis et al. [91] proposed a hybrid prismatic/tetrahedral mesh generation technique which used triangular prisms near the boundary to resolve viscous effects and tetrahedra elsewhere. More recently, NASA has developed their FUN3D (Fully Unstructured Navier-Stokes) code which has the capability to solve the Reynolds averaged Navier-Stokes equations over an unstructured tetrahedral mesh. This code has been applied to many different fluid-structure interaction problems [76, 92].

In recent years, the focus in unstructured grid generation research has been on improving performance [93–95]. This is necessary due to some of the large models currently being used in research and industry. In aeroelastic simulations, the meshes are often much smaller, and such methods are not necessary.

1.3 Problem Statement and Research Objectives

The current state of the art in transonic aeroelastic simulations is to couple the fluid and structure through the Direct Eulerian-Lagrangian method. Up to this

point, the method has primarily been applied to simple wing planforms with low to moderate leading edge sweep. At the same time, some of the most difficult, unsolved transonic flutter problems involve wings with leading and trailing edge kinks and high leading edge sweep. Application of the Direct Eulerian-Lagrangian method to these problems is needed.

The objective of this research is twofold. The first objective is to improve upon the existing computer codes that have been used to implement the Direct Eulerian-Lagrangian method in transonic aeroelastic simulations. The second objective is to apply the updated codes to the solutions of transonic aeroelastic problems which were previously unsolved. The updates to accomplish these tasks are as follows:

1. Update the current three-dimensional unstructured meshing code to generate meshes for highly swept wings with leading and trailing edge kinks and tips that are not parallel to the root.
2. Update the structural dynamic modeling portion of the code to generate nonlinear finite element models for cranked wings that can be coupled to the new fluid mesh.
3. Update the aeroelastic codes to apply the Direct Eulerian-Lagrangian coupling method to the new fluid and structural models.

1.4 Dissertation Outline

Immediately following this introductory chapter, the mixed Eulerian-Lagrangian formulation of the fluid model and the generation of the unstructured tetrahe-

dral mesh is discussed (Chapter 2). Next, Chapter 3 contains a discussion of the nonlinear structural model of the wing. In Chapter 4, coupling of the fluid and structural models by the Direct Eulerian-Lagrangian method is presented. Chapter 5 contains comparisons with previously published computational aeroelastic results that serve to validate the current code. Following this, Chapter 6 contains the results and analysis from transonic aeroelastic simulations of the NEXST-2 supersonic transport wing and a generic high speed cranked wing. Finally, conclusions and recommendations for future research are presented in Chapter 7.

Also included are two appendices. Appendix A contains the User's Guide to a graphical user interface (GUI) front end for the updated aeroelasticity programs. Appendix B summarizes the changes made to the meshing and aeroelasticity programs.

CHAPTER 2

Mixed Eulerian-Lagrangian Formulation of the Fluid Model

2.1 Introduction

In this chapter, the dynamic model of the fluid surrounding the wing is presented. The computational simulation of aeroelastic problems requires that the fluid model accurately calculate the pressure on the moving wing surface. For transonic problems, the location and strength of shocks on the wing surface is also of primary importance. Since the wing surface is moving, the fluid model must also accommodate a moving boundary. In Chapter 4, the fluid model will be coupled with the finite element structural model from Chapter 3. The fluid model presented here is a modified version of the model used in References 10, 11, 37–42, 44–51, 53–56, 96.

The fluid model employs the mixed Eulerian-Lagrangian formulation of the classical Euler equations and is discretized using a Galerkin finite element approach on an unstructured tetrahedral mesh. The mixed Eulerian-Lagrangian formulation (also known as Arbitrary Lagrangian-Eulerian (ALE)) is used to solve the governing equations in the presence of a moving boundary. This formulation was developed by Hirt et al. [97] and has become common in the solution of com-

putational fluid dynamic problems with a moving boundary [4, 9, 25, 26, 98–101]. The Euler equations have proven useful in the analysis of inviscid, transonic flows due to their ability to accurately model the strength and location of shocks while not suffering from the computational cost of full Navier-Stokes solutions [4, 9, 24–26, 29, 33, 34, 57, 61, 62, 64, 65, 102–111]. Discretizing the equations with the Galerkin finite element approach allows for accurate coupling at the fluid-structure boundary via the Direct Eulerian-Lagrangian approach which is explained in Chapter 4. An unstructured, tetrahedral mesh is used because of its ability to discretize the complex geometry of an aircraft wing and deform with the wing as it moves.

2.2 Governing Equations

The fluid model is based on the integral (or weak) form of the conservation laws,

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{W} dV + \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} dS = 0 \quad (2.1a)$$

where Ω is a control volume with boundary $\partial\Omega$ moving with velocity \mathbf{U} , $\mathbf{n} = n_i \hat{\mathbf{e}}_i$ is the outward unit normal of the control volume, $\hat{\mathbf{e}}_i$ are unit vectors, and

$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e \end{bmatrix} \quad (2.1b)$$

$$\mathbf{F} = \mathbf{F}_j \hat{\mathbf{e}}_j, \quad \mathbf{F}_j = \begin{bmatrix} \rho(u_j - U_j) \\ \rho u_1(u_j - U_j) - \sigma_{1j} \\ \rho u_2(u_j - U_j) - \sigma_{2j} \\ \rho u_3(u_j - U_j) - \sigma_{3j} \\ \rho e(u_j - U_j) - \sigma_{ij} u_i \end{bmatrix} \quad (2.1c)$$

Here, ρ is the fluid density, \mathbf{u} is the fluid velocity, e is the total energy per unit mass, σ_{ij} is the Cartesian stress tensor, and u_j and U_j are the Cartesian components of \mathbf{u} and \mathbf{U} . Equation 2.1a represents conservation of mass, conservation of momentum in three directions, and conservation of energy. It should be noted that body forces have been neglected.

In Equation 2.1c, if $U = 0$, the formulation becomes the standard Euler equations. If $U = u$, the equations are Lagrangian and the fluid element boundaries move with the fluid. When $U \neq u$, the description is a mix of Eulerian and Lagrangian, hence “the mixed Eulerian-Lagrangian formulation.” This formulation is particularly useful in flutter simulations where the fluid-structure boundary moves. The fluid element velocity on the wing surface is set to the velocity of the wing. The element velocity at the outer boundary is set to zero. The velocity for elements between the wing and the outer boundary is determined through a mesh motion scheme described in Section 2.5.

The pressure, p , can be eliminated by using the equation of state for a perfect gas.

$$p = \rho(\gamma - 1)\left(e - \frac{1}{2}u_i u_i\right) \quad (2.2)$$

In the Euler fluid model, viscosity is neglected and the stress tensor can be represented by

$$\sigma_{ij} = -p\delta_{ij} \quad (2.3)$$

This simplification yields accurate results except in the cases of shock-induced boundary layer separation or separated flow due to high angle of attack.

The terms in the vector \mathbf{W} can be approximated as a linear combination of

shape functions.

$$\mathbf{W}(x_1, x_2, x_3, t) = \sum_j \mathbf{W}_j \phi_j(x_1, x_2, x_3, t) \quad (2.4)$$

Here, \mathbf{W}_j are the nodal values of \mathbf{W} and ϕ_j are linear shape functions. They are applied over tetrahedral elements. By substituting Equation 2.4 into the Galerkin form of Equation 2.1a and applying the divergence theorem, we get

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{W} \phi_i dV + \int_{\Omega} \phi_i \nabla \cdot \mathbf{F} dV = \frac{\partial}{\partial t} \int_{\Omega} \mathbf{W} \phi_i dV + \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} \phi_i dS - \int_{\partial\Omega} \mathbf{F} \cdot \nabla \phi_i dS = 0 \quad (2.5)$$

The integrals are evaluated numerically using Gaussian integration, and the space-discretized nodal equations for the fluid domain become

$$\frac{d}{dt} \sum_j \mathbf{m}_{ij} \mathbf{W}_j + \mathbf{Q}_i - \mathbf{D}_i = 0 \quad (2.6)$$

where $i = 1, 2, \dots, N$ (all nodes) and the summation on j is over all nodes in the “superelement” or control volume associated with node i (the union of all elements that meet at node i). In Equation 2.6, \mathbf{m}_{ij} is the consistent mass matrix, \mathbf{Q}_i is the flux vector, and \mathbf{D}_i are dissipative fluxes of the Jameson-Mavriplis type which are added to capture shocks and to stabilize the scheme. These were developed in References 106 and 108 and modified in References 96 and 46.

2.3 Integration Scheme

A multi-stage Runge-Kutta scheme is used to integrate Equation 2.6. The scheme is shown below.

$$\begin{aligned}
\mathbf{W}^{(0)} &= \mathbf{W}^{[n]} \\
\mathbf{W}^{(1)} &= [\mathbf{M}]^{-1} \left(\mathbf{M}^{(0)} \mathbf{W}^{(0)} - \alpha_1 \Delta t [\mathbf{Q}(\mathbf{W}^{(0)}) - \mathbf{D}(\mathbf{W}^{(0)})] \right) \\
\mathbf{W}^{(2)} &= [\mathbf{M}]^{-1} \left(\mathbf{M}^{(0)} \mathbf{W}^{(0)} - \alpha_2 \Delta t [\mathbf{Q}(\mathbf{W}^{(1)}) - \mathbf{D}(\mathbf{W}^{(1)})] \right) \\
\mathbf{W}^{(3)} &= [\mathbf{M}]^{-1} \left(\mathbf{M}^{(0)} \mathbf{W}^{(0)} - \alpha_3 \Delta t [\mathbf{Q}(\mathbf{W}^{(2)}) - \mathbf{D}(\mathbf{W}^{(1)})] \right) \\
\mathbf{W}^{(4)} &= [\mathbf{M}]^{-1} \left(\mathbf{M}^{(0)} \mathbf{W}^{(0)} - \alpha_4 \Delta t [\mathbf{Q}(\mathbf{W}^{(3)}) - \mathbf{D}(\mathbf{W}^{(1)})] \right) \\
\mathbf{W}^{(5)} &= [\mathbf{M}]^{-1} \left(\mathbf{M}^{(0)} \mathbf{W}^{(0)} - \alpha_5 \Delta t [\mathbf{Q}(\mathbf{W}^{(4)}) - \mathbf{D}(\mathbf{W}^{(1)})] \right) \\
\mathbf{W}^{[n+1]} &= \mathbf{W}^{(5)}
\end{aligned} \tag{2.7}$$

where α_i for $i = 1, 2, 3, 4, 5$ are $\frac{1}{4}, \frac{1}{6}, \frac{3}{8}, \frac{1}{2}, 1$ respectively, and Δt is the time step. Superscripts in parentheses represent the Runge-Kutta stage number, and superscripts in square brackets represent the global time step number. To save computational time, the dissipative terms are updated only during the first two stages.

The time step, Δt_i^* , is calculated for each node as the minimum time for a wave to propagate from the chosen node, i to its adjacent nodes, $k = 1, 2, \dots, n$, where n is the number of adjacent nodes.

$$\Delta t_i^* = \min(\Delta t_{ik}), \forall k \text{ adjacent to node } i \tag{2.8a}$$

$$\Delta t_{ik} = \frac{\Delta \mathbf{x} \cdot \Delta \mathbf{x}}{\left| \left(\bar{u} - \bar{U} \right) \cdot \Delta \mathbf{x} \right| + \bar{a} |\Delta \mathbf{x}|} \tag{2.8b}$$

$$\Delta \mathbf{x} = (x_k - x_i) \hat{\mathbf{e}}_x + (y_k - y_i) \hat{\mathbf{e}}_y + (z_k - z_i) \hat{\mathbf{e}}_z \tag{2.8c}$$

Here, \bar{u} , \bar{U} , and \bar{a} are the mean flow velocity, mesh velocity, and speed of sound between nodes i and k . The local time step can be used to accelerate the

convergence for steady calculations. In this case, each node proceeds to the final solution at its own speed. This cannot be done for unsteady calculations, where a global time step must be used. The global time step, Δt^* is calculated using Equation 2.9a. The explicit Runge-Kutta method used here must be bounded by the Courant-Friederich-Lewy (CFL) number, λ , in order to obtain numerical stability (Equation 2.9b). The CFL number has a maximum value of $2\sqrt{2}$ for the four-stage scheme [105].

$$\Delta t^* = \min(\Delta t_i^*), \forall \text{ nodes } i \quad (2.9a)$$

$$\Delta t \leq \lambda \Delta t^* \quad (2.9b)$$

2.4 Mesh Generation by the Advancing Front Method

To solve the conservation equations from Section 2.2, they must be discretized over the solution domain with a three-dimensional (3-D) mesh. Meshes can be classified into two types: structured and unstructured. Structured meshes usually employ hexahedral elements and have a high degree of regularity and smoothness. This is beneficial, as it can result in a more efficient solution process. Structured meshes, however, suffer from the limitation of hexahedral elements to discretize complicated domains. Unstructured meshes, in contrast, excel at discretizing complex domains as they often use tetrahedral elements [76, 84, 86, 91, 100, 112, 113]. Computational performance and the generation of flow visualizations is complicated by the irregular arrangement of elements, but the benefits often outweigh these negatives. Unstructured tetrahedral meshes have been used in a large number of CFD and aeroelasticity problems [74, 76, 86, 89, 91, 92, 100, 106, 112–114].

A popular technique for the generation of unstructured tetrahedral meshes is the advancing front method (AFM) [36, 74, 75, 85–90, 93–95, 112]. The availability of a well-developed computer program designed to interface with the aeroelastic solution programs made it a good choice for mesh generation. The AFM presented in this section is a modified version of the process used in References 10, 11, 37, 44–51, 53–56. Using a computer program where the source code is available to be modified has advantages over the use of a packaged mesh generation program. By modifying the source code, the meshing program can be adjusted to work as desired. In addition, for the mesh to be compatible with the aeroelastic programs, each node must be associated to either a specific surface or the interior space of the coupled fluid-structure model. This is especially important on the wing surface and wing tip.

The AFM is carried out in four stages:

1. The fluid domain is defined by a set of boundary lines and discretized using a coarse background mesh which defines the fluid mesh size at different locations in the domain. Note that “background mesh” which only exists to facilitate the AFM is different from the “fluid mesh,” which refers to the 3-D mesh that discretizes the fluid domain in order to solve the conservation equations.
2. Boundary lines are discretized with grid points that will serve as the initial front for discretizing the two-dimensional (2-D) surfaces.
3. Two-dimensional surfaces are defined by groups of boundary lines. The surfaces are discretized with triangles using the 2-D AFM. Surfaces that

are curved (wing surface, wing tip, and cylindrical far-field) are mapped to a flat surface before they are discretized. The curvature is restored after discretization. The triangles on the 2-D surface serve as the initial front for the 3-D AFM.

4. The 2-D surfaces are patched together to form a closed volume. The 3-D AFM is used to discretize the volume with tetrahedral elements.

2.4.1 Mesh Domain

To simplify the mesh creation, only the upper half of the domain is discretized by the AFM. This limits the analysis to symmetric airfoils. The full domain is generated by creating a mirror of the upper domain about the plane at $Z = 0$ and coupling the two halves together. The upper half of the mesh domain for a high speed civil transport (HSCT) wing is shown in Figures 2.1 and 2.2. Boundary line numbers are also shown.

The mesh domain is bounded by six surfaces: the symmetry plane ($Z = 0$), the wing upper surface, the wing tip, the tunnel attach wall ($Y = 0$), the far tunnel wall, and the cylindrical far-field.

2.4.2 Mesh Parameters

Each wing planform requires the creation of a unique fluid mesh. While a few of the mesh parameters are fixed, most of them are user-definable in a series of text-based input files. Planform shape, mesh density, and mesh resolution can be adjusted. Generating the input files can be a tedious, error prone task. To make

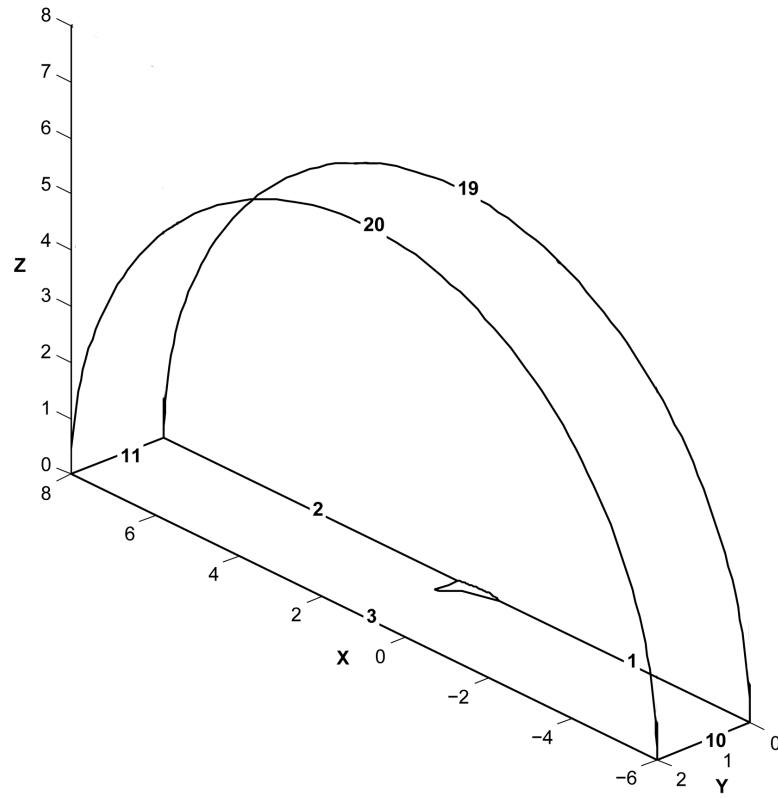


Figure 2.1: Upper half of the mesh domain with the wing on the symmetry plane.

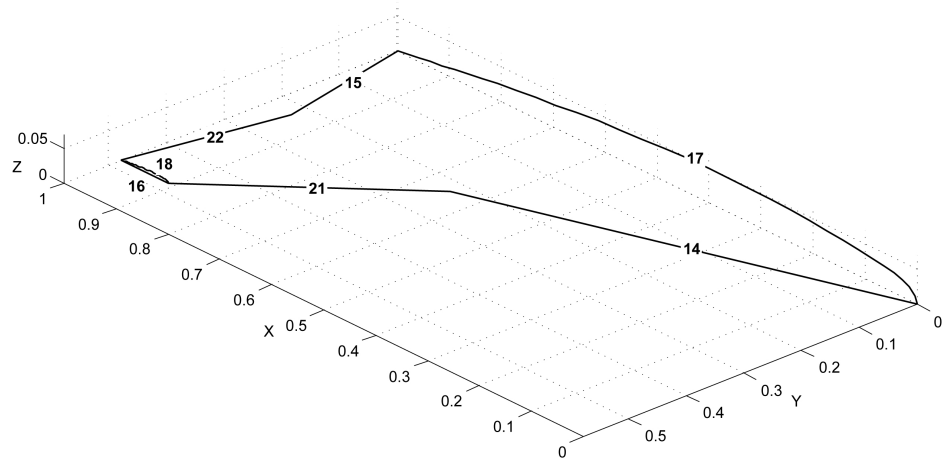


Figure 2.2: Wing region of the upper half of the mesh domain.

Table 2.1: Wing planform parameters.

Parameter	Description
Λ_{L1}	Inboard leading edge sweep angle
Λ_{T1}	Inboard trailing edge sweep angle
Λ_{K1}	Kink sweep angle
Y_{L1}	Span location of kink at leading edge
Λ_{L2}	Outboard leading edge sweep angle
Λ_{T2}	Outboard trailing edge sweep angle
Λ_{K2}	Tip sweep angle
Y_{L2}	Span location of tip at leading edge
T_{span}	Thickness at user defined span locations relative to root
Y_{wall}	Far tunnel wall location

the process of generating a mesh as simple as possible, a series of Python [115] scripts with a graphical user interface (GUI) were created to generate all the required input files. The collection of these scripts is a program called dfLap (Direct Flutter Analysis Program). Along with quicker, more reliable mesh creation, dfLap also interfaces with the steady CFD, static aeroelastic, and flutter programs. It can also be used to generate visualizations of meshes and aeroelastic results. The dfLap User's Guide can be found in Appendix A.

The parameters used to define the wing planform are shown in Table 2.1 and Figure 2.3. Note that Λ_{T1} is negative while all other angles are shown positive.

The wing root chord is always normalized to a length of 1.0, and the cylindrical far-field always has a radius of 7 with a center at $x = 1$. The location of the far

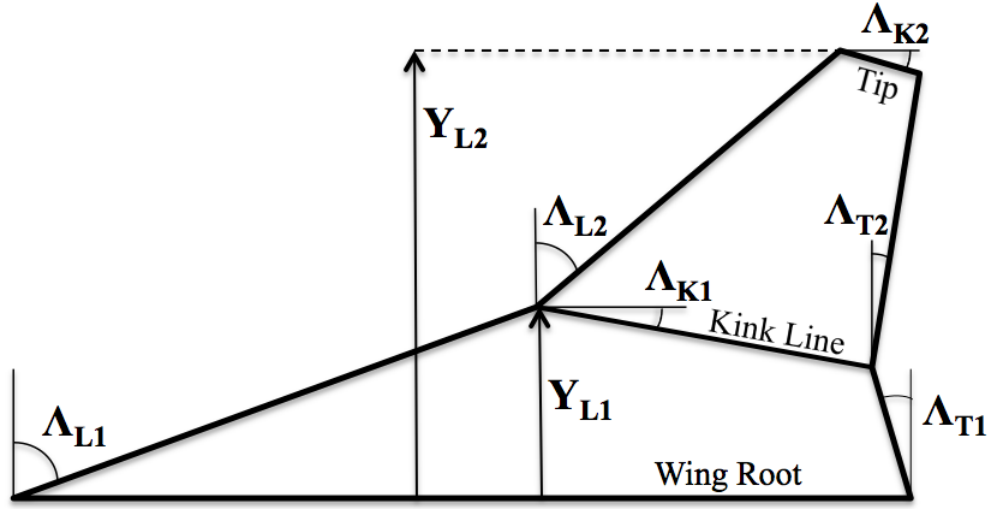


Figure 2.3: Wing planform parameters.

tunnel wall (Y_{wall}) can be changed as desired to accommodate wings of different span.

2.4.3 Background Mesh

The mesh domain is initially discretized by a coarse, tetrahedral background mesh. Each vertex in the background mesh has a local fluid mesh size, δ , associated with it. By linearly interpolating between the vertices, the fluid mesh size at any point in the domain can be calculated. The background mesh for a HSCT wing is shown in Figures 2.4, 2.5, and 2.6. In the figures, the wing is shown as a bold red line. The background mesh must be adjusted to match each different wing planform. This can be a tedious process, but is done automatically based on the wing planform shape by dfLap. Figures 2.5 and 2.6 show that the background grid extends outside of the cylindrical far-field boundary of the computational domain. The background mesh is designed this way for simplicity. No fluid mesh

nodes are created in the space outside of the far-field boundary.

2.4.4 Boundary Line Discretization

Boundary lines are discretized by the creation of fluid mesh grid points. The location of the grid points can be calculated based on the δ values on the line or the locations can be defined explicitly. For most boundary lines, defining the points explicitly is inefficient and provides no benefit. In this case, δ values are

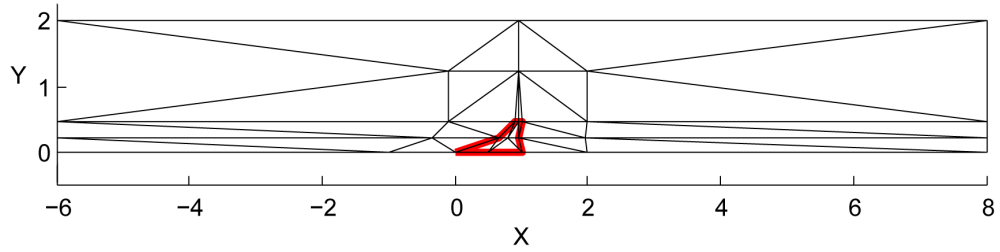


Figure 2.4: Background grid on the $Z = 0$ plane.

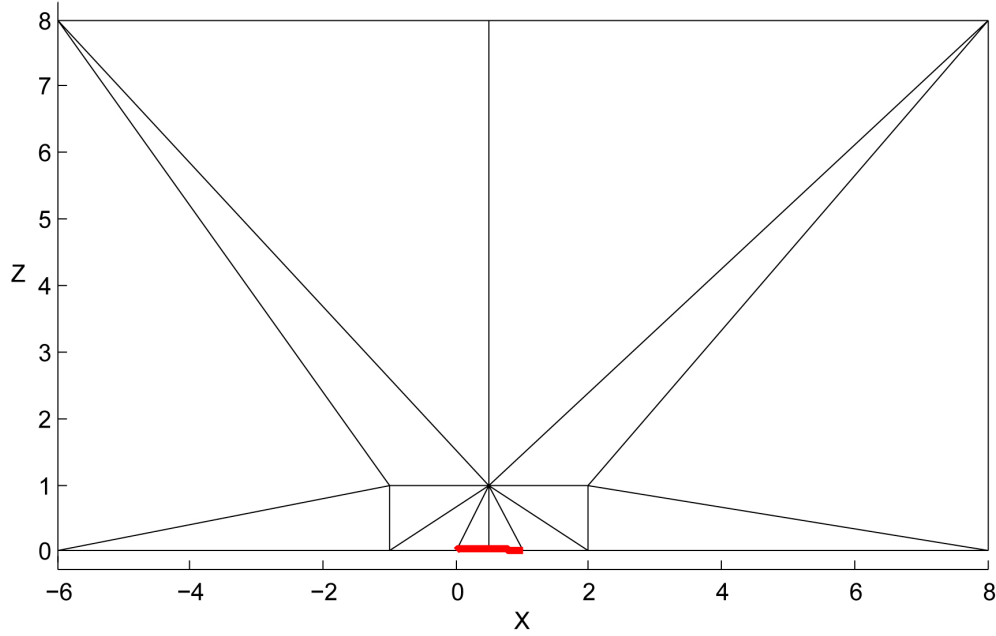


Figure 2.5: Background grid on the $Y = 0$ plane.

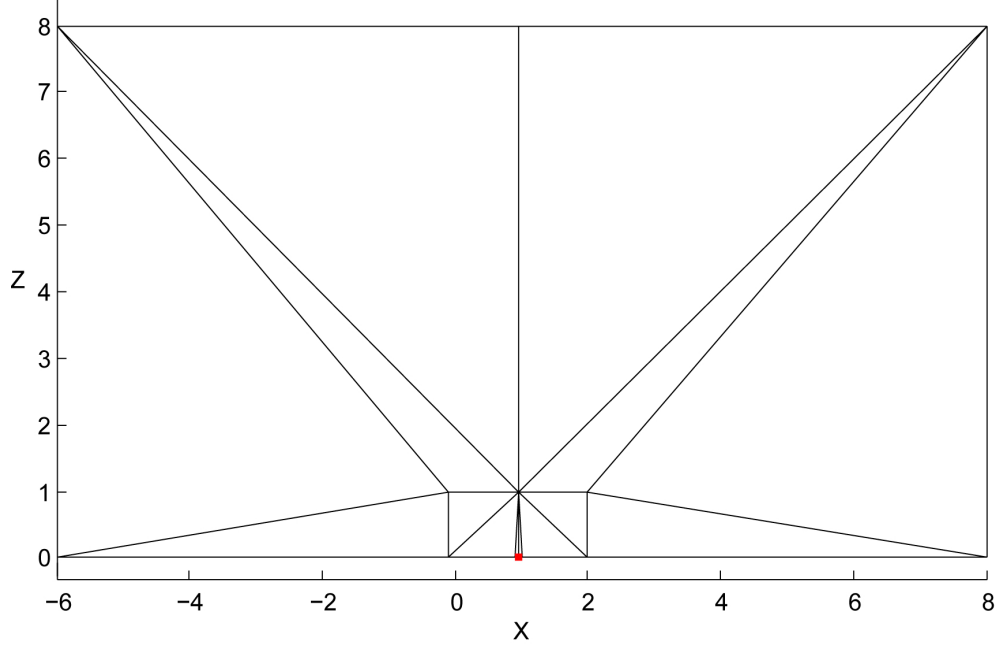


Figure 2.6: Background grid on the $X - Z$ plane at the wing tip.

used. This method is used on all boundary lines other than 16, 17, and 18. It is the author's experience that for areas where boundary lines intersect at a small angle (wing tip trailing edge), a better mesh can be created by explicitly defining the grid points. This method is used on boundary lines 16, 17, and 18. When grid points are explicitly defined, it is necessary that a δ value for each grid point also be defined. The δ values must be matched to the local mesh size (the distance between grid points). If this is not done, a poorly formed mesh will result.

To calculate grid points on a boundary line using δ values, the first step is to use Equation 2.10. Here, A_s is an integer that represents the number of segments the boundary line will be divided into, and $\delta_c(s)$ is a function that defines the δ values along a boundary line with length l .

$$A_s = \int_0^l \frac{1}{\delta_c(s)} ds \quad (2.10)$$

After A_s is calculated, the location of grid points along the boundary line is determined by solving for l_i in Equation 2.11.

$$i = \int_0^{l_i} \frac{1}{\delta_c(s)} ds, \quad i = 0, \dots, A_s \quad (2.11)$$

In Equations 2.10 and 2.11, δ values are interpolated from the background mesh for boundary lines 1–3, 10, 11, 19, and 20. For lines on the wing leading and trailing edge (14, 15, 21, and 22), δ values are calculated based on the defined δ values at the leading and trailing edges of lines 17 and 18. The δ values along lines 14, 15, 21, and 22 are scaled based on the local wing chord length. This process allows tighter control over the mesh on the wing surface and has proven very reliable [10, 11, 48–51, 53–56].

2.4.5 2-D Surface Discretization

Before a 2-D surface can be discretized by the AFM, the edges of the surface must be defined using the previously discretized boundary lines. Six 2-D surfaces define the outer boundary of the mesh domain: the symmetry plane, the wing upper surface, the wing tip, the tunnel attach wall, the far tunnel wall, and the cylindrical far-field. The boundary lines that make up each surface are listed in Table 2.2. When the boundary lines for each surface are linked together, they form a closed surface which serves as the initial front for the 2-D discretization. The nodes in the initial front are numbered in the counterclockwise direction, thus as one moves along the front from smaller to larger numbered nodes, the domain to be discretized is always on the left-hand side.

Once the initial front has been assembled, the 2-D AFM is carried out as

follows:

1. Select the shortest line in the front. Call this line \overline{AB} . Finding this line can be a time-consuming process. To speed up the search, a heap list based on a binary tree data structure as presented by Löhner is used [87].
2. Calculate δ at the midpoint of \overline{AB} . Call this point M . For the symmetry plane, the tunnel attach wall, the tunnel far wall, and the cylindrical far-field, δ is interpolated from the background mesh. For the wing surface and wing tip, δ is interpolated from the explicitly defined values associated with the boundary lines at the wing root (line 17). In these cases, the delta values are scaled by the local chord length.
3. Generate a new point inside the domain as shown in Figure 2.7, where C is

Table 2.2: Aerodynamic mesh surface definitions.

ID	Surface	Boundary Lines
1	Symmetry plane	1, 14, 21, 16, 22, 15, 2, 11, 3, 10
2	Wing surface	14, 21, 18, 22, 15, 17
3	Wing tip	16, 18
4	Tunnel attach wall	1, 17, 2, 19
5	Tunnel far wall	3, 20
6	Cylindrical far-field	10, 19, 11, 20

the new grid point. The value of δ_1 is determined according to

$$\delta_1 = \begin{cases} \delta & \text{if } 0.55 |\overline{AB}| \leq \delta \leq 2 |\overline{AB}| \\ 0.55 |\overline{AB}| & \text{if } 0.55 |\overline{AB}| > \delta \\ 2 |\overline{AB}| & \text{if } \delta > 2 |\overline{AB}| \end{cases} \quad (2.12)$$

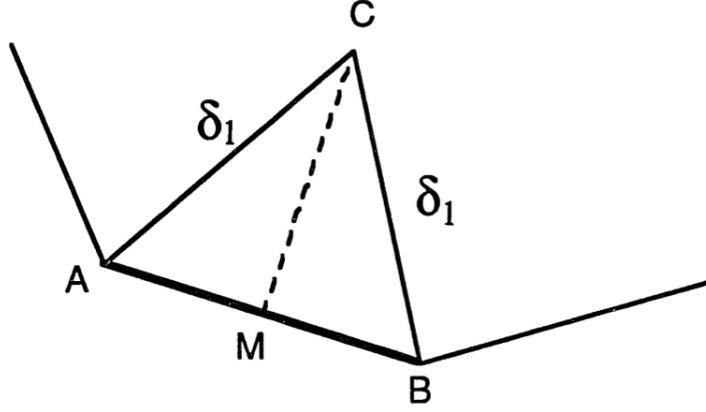


Figure 2.7: Generation of a new point.

4. Create a list of candidate points to form a triangle with \overline{AB} . Call this list a_i , $i = 1, \dots, n$. Any nodes in the front that lie inside a circle with radius 2.5δ with center at point C are included in a_i . The list is sorted with points closest to C first and points furthest from C last. Point C is only included in the list if no point in a_i is within a distance of $1.5\delta_1$ from both points A and B . If no points meet this criteria, C is placed at the top of the list of candidate points.
5. Select the best candidate to form the new triangle. Start at the top of the list from step 4 and check for intersection between $\overline{Aa_i}$ and $\overline{Ba_i}$ with any other line in front. If there is no intersection, $\triangle ABa_i$ is the new triangle. If

there is an intersection with an existing line, try the next point in the list,
 a_i .

6. As new nodes and lines are added to the mesh, the front advances, and lines that were previously in the front are left behind. This process is repeated until no faces remain in the front.

Two dimensional meshes for all six boundary surfaces are shown in Figures 2.8–2.13.

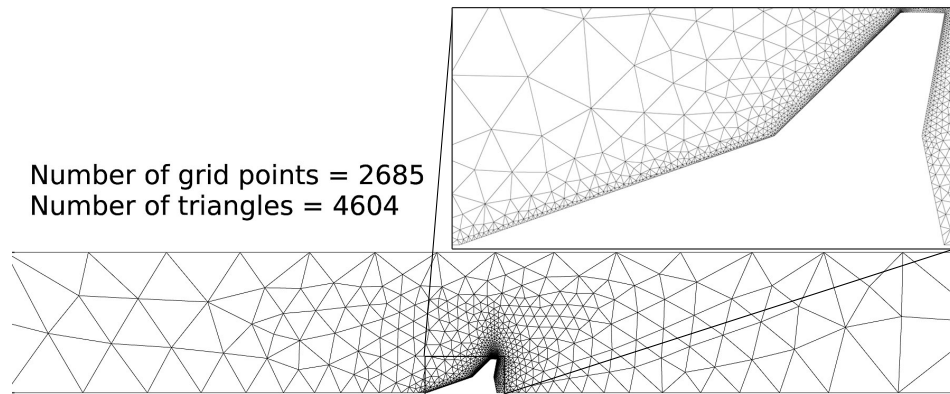


Figure 2.8: Symmetry plane mesh.

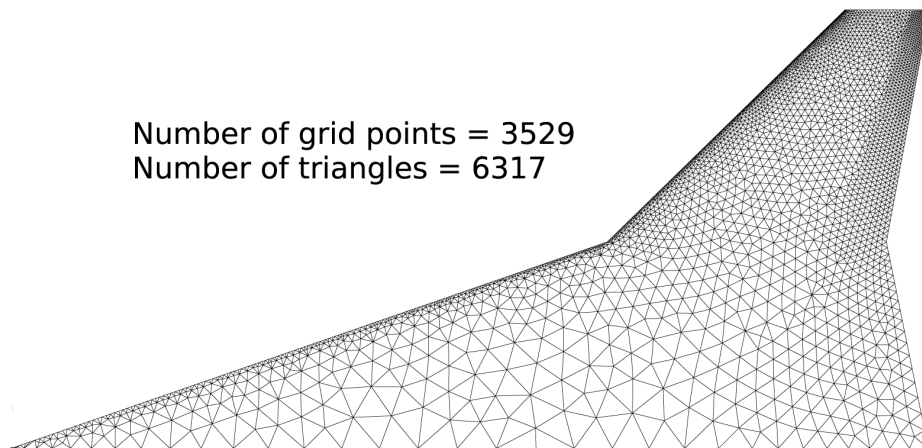


Figure 2.9: Wing surface mesh.

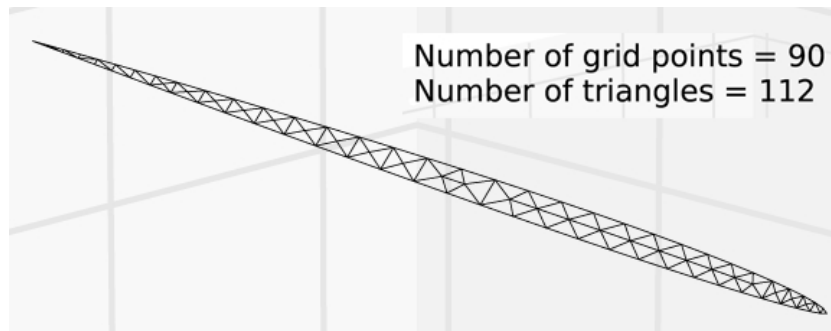


Figure 2.10: Wing tip mesh.

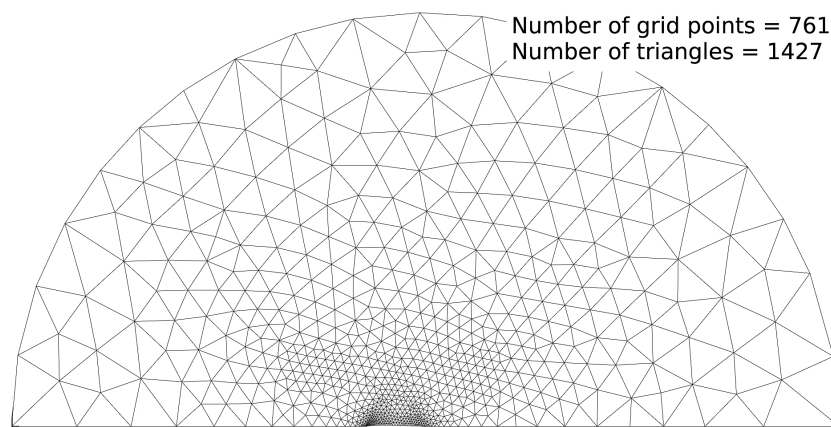


Figure 2.11: Tunnel attach wall mesh.

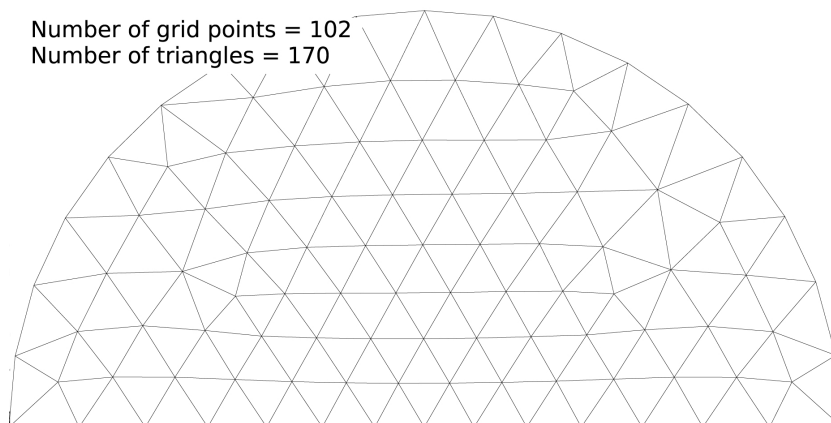


Figure 2.12: Tunnel far wall mesh.

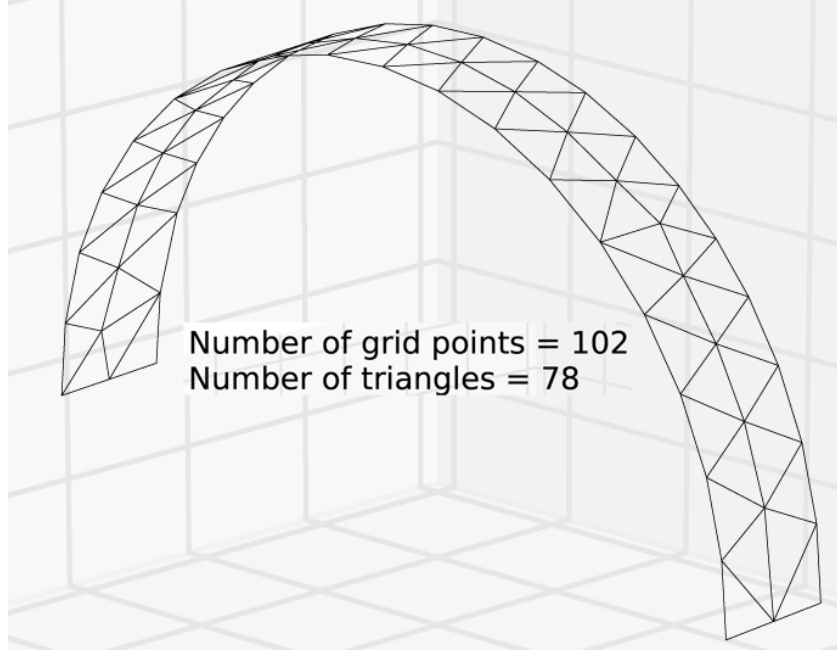


Figure 2.13: Cylindrical far-field mesh.

2.4.6 Curved 2-D Surfaces

Three of the surfaces in the mesh domain are not flat (wing surface, wing tip, and cylindrical far-field). These surfaces must be mapped to a flat surface before they are discretized. Once discretization of these surfaces is complete, the 2-D mesh is mapped back to the original, curved shape.

The cylindrical far-field is the simplest surface to map. The surface is a cylinder with radius 7. When mapped to 2-D, it becomes a rectangle with long edge length 7π . Equations 2.13a–b are used to map points between the 2-D and 3-D representations. The flat representation of the cylindrical far-field is shown in Figure 2.14. For the curved representation, see Figure 2.13.

$$x_{2D} = 7 \arccos\left(\frac{1 - x_{3D}}{7}\right) \quad (2.13a)$$

$$y_{2D} = y_{3D} \quad (2.13b)$$

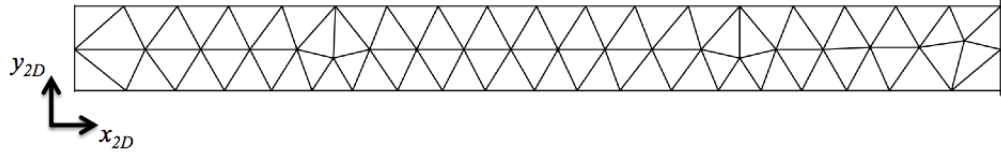


Figure 2.14: Cylindrical far-field mesh mapped to two dimensions.

To map the wing surface, the leading edge coordinates do not change between the two representations. The wing is mapped by moving the trailing edge root, trailing edge kink, and trailing edge tip along the root line, kink line, and tip line respectively (see Figure 2.3). The curved and flat wing surface are shown in

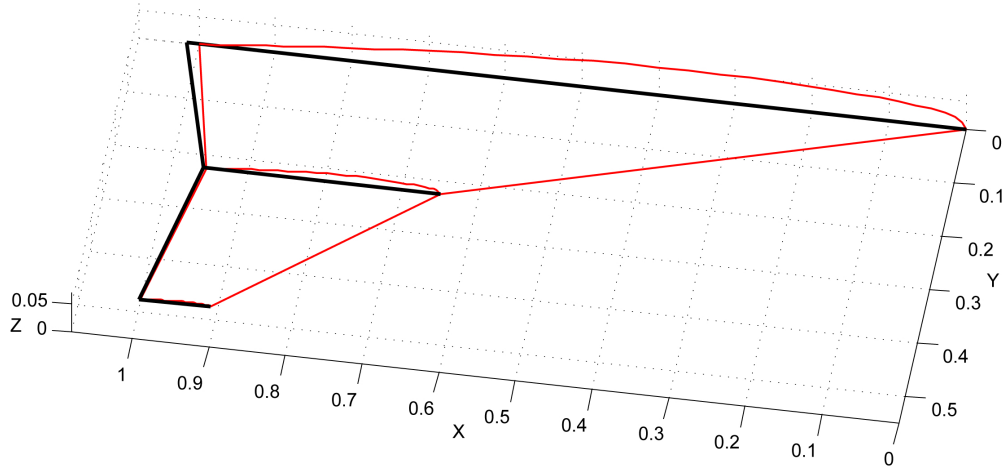


Figure 2.15: Wing surface mapped to two dimensions.

Figure 2.15. The curved wing is drawn in red, the flat wing is drawn in black. Once the flat representation of the wing is discretized, it is mapped back to the curved surface by calculating the percent chord and percent span of each point on the flat surface and mapping it back to the same percent chord and span on the curved surface. The out-of-plane coordinates that give the wing its thickness are

also calculated using percent chord and percent span. At each percent span, the wing thickness is defined by the T_{span} parameter from Table 2.1. Wing thickness at each percent chord is defined by the root chord boundary line (i.e. the airfoil shape). Using these parameters together, the thickness of the wing is calculated at each point.

The curved wing tip boundary is formed by two airfoil-shaped curves on perpendicular planes. Line 16 lies on the $X - Y$ plane, while line 18 is in a perpendicular plane. The tip surface between these lines is similar to a quarter cylinder, but it has a varying radius. Mapping to the flat surface results in the shape shown in Figure 2.16. Points on the two surfaces are mapped according to the relation

$$x_{3D} = x_{2D} \quad (2.14a)$$

$$y_{3D} = T(x_{2D}) \sin \left(\frac{\pi}{4} \left(1 - \frac{y_{2D}}{T(x_{2D})} \right) \right) \quad (2.14b)$$

$$z_{3D} = T(x_{2D}) \cos \left(\frac{\pi}{4} \left(1 - \frac{y_{2D}}{T(x_{2D})} \right) \right) \quad (2.14c)$$

where $T(x_{2D})$ is the distance between the boundary lines and the midline at x_{2D} .

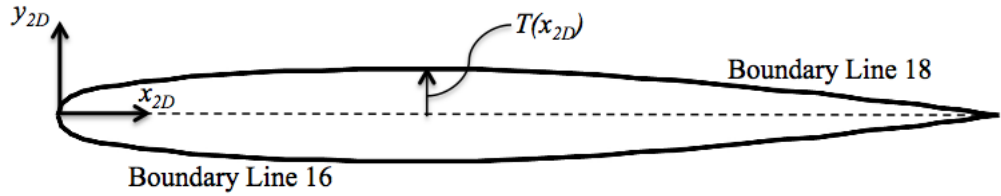


Figure 2.16: Wing tip mapped to two dimensions.

2.4.7 Wing Tip Discretization

The 2-D discretization of the wing tip has been a difficult problem throughout the life of the meshing code. Historically, it has been a common occurrence for the

code to fail while attempting to create a mesh on the wing tip. Different solutions have been attempted, but often, the user was forced to go into the source code and alter the logic until a usable mesh was generated. To avoid issues like this, the code has been updated to read in a user provided tip mesh which can be mapped to any airfoil shape and thickness. This tip mesh can be generated automatically in dfLap for a range of mesh densities. The dfLap method generates a mesh using a repeating pattern and has proven very reliable. Using the AFM to discretize the wing tip is still an option, but it is not recommended. Details of the tip meshing process can be found in the dfLap User’s Guide in Appendix A.

2.4.8 The 3-D Mesh

Once the 2-D surfaces have been discretized, they are assembled as shown in Figure 2.17. Note that the tunnel far wall has been removed to allow for easier viewing. At this point, the 3-D AFM is used to discretize the volume. The discretization procedure employed in this work is the same as that presented by Hwang [46]. The process is similar to that of the 2-D AFM, but is more complicated due to the additional dimension being considered. Once the half-domain has been discretized, a mirror of it is created and attached to form the lower half of the domain.

2.5 Mesh Motion

During the aeroelastic simulations, the wing deforms, and the aerodynamic mesh must move to accommodate the changes in geometry. The mesh is fixed to the

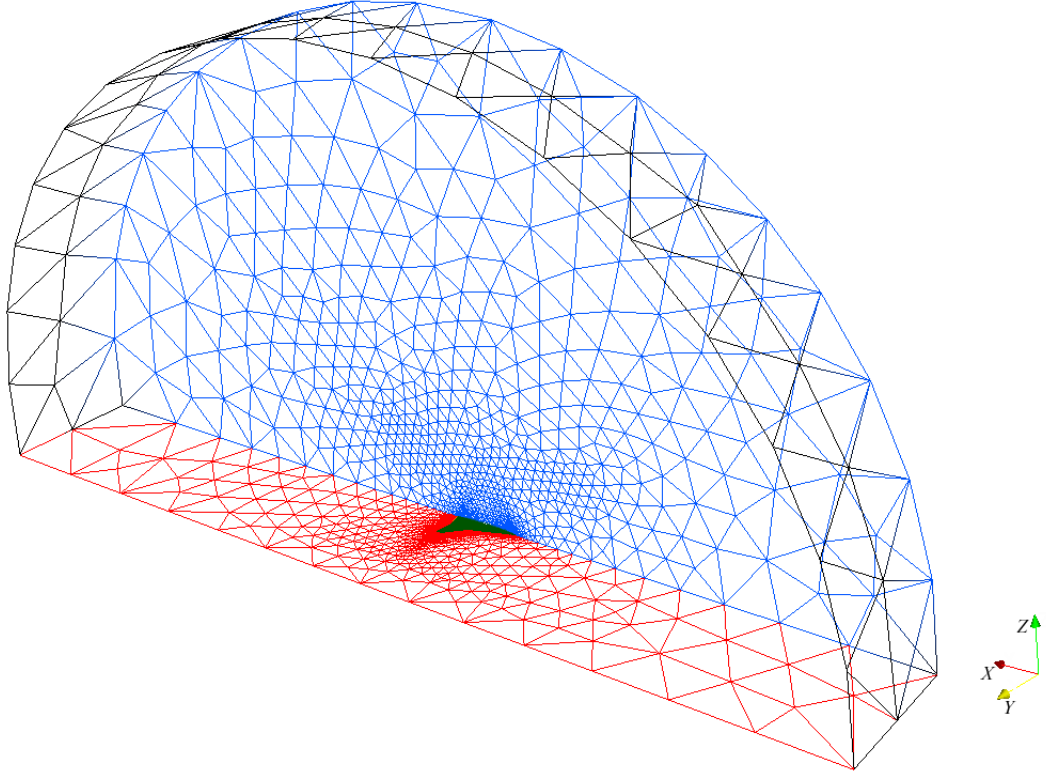


Figure 2.17: 2-D mesh.

wing, tunnel wall, and far-field boundaries, and a dynamic mesh motion algorithm is used to calculate the position of the interior nodes. This mesh motion algorithm treats each edge in the mesh as a spring. The stiffness of the spring is inversely proportional to the square of the length of the edge. Jacobi iterations are used to calculate the equilibrium of the system at each Runge-Kutta stage. Usually three iterations are sufficient for convergence. The algorithm was originally proposed by Batina [36, 113] and was implemented in the meshing code by Hwang [46]. It has since been updated to the nonlinear spring analogy [49] that was designed to handle larger wing deformation.

The mesh motion algorithm works with the Runge-Kutta integration scheme from Section 2.3. For each Runge-Kutta stage, Jacobi iterations are used to

calculate the equilibrium state of the spring system. Steps one and two below are completed at only the first Runge-Kutta stage of each global time step, while steps three through seven are completed at every stage of every time step.

1. Calculate the spring stiffness for a given edge, \overline{ij} by

$$k_m = \frac{1}{\left((x_j^{[n]} - x_i^{[n]})^2 + (y_j^{[n]} - y_i^{[n]})^2 + (z_j^{[n]} - z_i^{[n]})^2\right)} \quad (2.15)$$

where superscripts in square brackets represent the global time step number.

2. Predict the displacements of each interior node, i , by extrapolating the displacements from the previous two iterations as shown in Equation 2.16.

$$\forall i \text{ interior: } \delta_{x_i}^P = 2\delta_{x_i}^{[n]} - \delta_{x_i}^{[n-1]}; \quad \delta_{y_i}^P = 2\delta_{y_i}^{[n]} - \delta_{y_i}^{[n-1]}; \quad \delta_{z_i}^P = 2\delta_{z_i}^{[n]} - \delta_{z_i}^{[n-1]} \quad (2.16)$$

3. Multiply the predicted displacements of interior nodes from step 2 by the Runge-Kutta coefficients from Equation 2.7.

$$\forall i \text{ interior: } \delta_{x_i}^{(k)} = \alpha_k \delta_{x_i}^P; \quad \delta_{y_i}^{(k)} = \alpha_k \delta_{y_i}^P; \quad \delta_{z_i}^{(k)} = \alpha_k \delta_{z_i}^P \quad (2.17)$$

Here, superscripts in parentheses, represent the Runge-Kutta stage number.

4. Get the location of the nodes on the wing surface, $x_i^{(k)}$, $y_i^{(k)}$, and $z_i^{(k)}$, for stage (k) from the aeroelastic code. Use these to calculate the displacements for each node on the wing.

$$\forall i \text{ on wing surface: } \delta_{x_i}^{(k)} = x_i^{(k)} - x_i^{[n]}; \quad \delta_{y_i}^{(k)} = y_i^{(k)} - y_i^{[n]}; \quad \delta_{z_i}^{(k)} = z_i^{(k)} - z_i^{[n]} \quad (2.18)$$

5. Set the displacements for nodes on the tunnel walls and cylindrical far-field to 0.

$$\forall i \text{ on tunnel wall: } \delta_{x_i}^{(k)} = 0; \quad \delta_{y_i}^{(k)} = 0; \quad \delta_{z_i}^{(k)} = 0 \quad (2.19)$$

6. Use Jacobi iterations to solve the static equilibrium of the spring system.

Usually two or three iterations are required.

$$\forall i \text{ interior: } \delta_{x_i}^{(k)} = \frac{\sum k_m \delta_{x_i}^{(k)}}{\sum k_m}; \quad \delta_{y_i}^{(k)} = \frac{\sum k_m \delta_{y_i}^{(k)}}{\sum k_m}; \quad \delta_{z_i}^{(k)} = \frac{\sum k_m \delta_{z_i}^{(k)}}{\sum k_m} \quad (2.20)$$

7. Update the locations of the nodes.

$$\forall i \text{ interior: } x_i^{(k)} = x_i^{[n]} + \delta_{x_i}^{(k)}; \quad y_i^{(k)} = y_i^{[n]} + \delta_{y_i}^{(k)}; \quad z_i^{(k)} = z_i^{[n]} + \delta_{z_i}^{(k)} \quad (2.21)$$

After the calculations are made for the fourth Runge-Kutta stage, the nodal locations for the current time step are updated.

$$\forall i \text{ interior: } x_i^{[n+1]} = x_i^{(4)}; \quad y_i^{[n+1]} = y_i^{(4)}; \quad z_i^{[n+1]} = z_i^{(4)} \quad (2.22)$$

2.6 Boundary Conditions

The fluid model described in the previous sections interacts with the fixed tunnel walls and the flexible wing structure at the fluid-structure boundary. Thus, an appropriate description of the boundary conditions is necessary.

On the tunnel walls, the mesh velocity is zero. The boundary condition for the walls is

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad (2.23)$$

where \mathbf{u} is the fluid velocity and \mathbf{n} is the normal vector of the wall surface.

On the wing surface, the mesh is moving with the wing, thus the boundary condition for the wing is

$$(\mathbf{u} - \mathbf{U}) \cdot \mathbf{n} = 0 \quad (2.24)$$

where \mathbf{U} is the mesh velocity.

On the far-field boundary, non-reflecting boundary conditions are used to minimize the reflection of outgoing disturbances [106]. By assuming that the flow is locally one-dimensional, the incoming Riemann invariant, R_∞ , can be calculated from the freestream flow (Equation 2.25a), and the outgoing invariant, R_i , can be determined by extrapolation from the interior domain (Equation 2.25b).

$$R_\infty = \mathbf{u}_\infty \cdot \mathbf{n} - \frac{2a_\infty}{\gamma - 1} \quad (2.25a)$$

$$R_i = \mathbf{u}_i \cdot \mathbf{n} - \frac{2a_i}{\gamma - 1} \quad (2.25b)$$

Because the mesh velocity at the far-field boundary is zero, the normal velocity of the flow and the speed of sound can be calculated as

$$\mathbf{u} \cdot \mathbf{n} = \frac{1}{2} (R_i + R_\infty) \quad (2.26a)$$

$$a = \frac{\gamma - 1}{4} (R_i + R_\infty) \quad (2.26b)$$

The tangential velocity and the entropy are also necessary to provide a complete definition of the flow at the far field boundary. For an inflow boundary, the tangential velocity and the entropy are set equal to their free stream values. For an outflow boundary, they are extrapolated from the interior.

CHAPTER 3

Nonlinear Structural Model

3.1 Introduction

In this chapter, the nonlinear structural model of the wing is presented. To accurately model the response of a real wing, a geometrically nonlinear three-dimensional model is created that accounts for large displacements and changes in stiffness due to deformation. In Chapter 4, a finite element (FE) representation of the wing structural model is coupled to the fluid model from Chapter 2. The wing model presented here is a modified version of the model used in References 11, 49–51, 53–56.

The model is based on a type of aeroelastic wing that is often used in wind tunnel tests [72, 73, 116]. The structural part of the wing consists of a thin aluminum alloy plate whose planform matches that of the wing. The plate is covered by balsa wood or flexible plastic foam to create the desired airfoil shape without adding mass or stiffness. Figure 3.1 shows a cutaway view of an example aeroelastic wing.

The wing is modeled using triangular finite elements with five degrees-of-freedom at each node (three translation and two out-of-plane rotation). Only the in-plane rotation (drilling) degree of freedom is excluded. Large displacements are

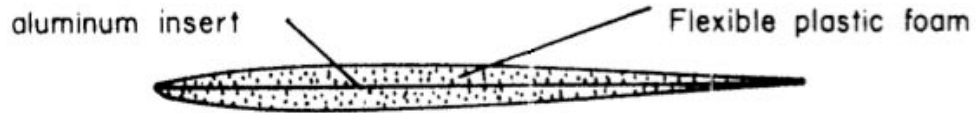


Figure 3.1: Aeroelastic wing cutaway courtesy of Reference 116.

accounted for by using element attached coordinate systems via a mixed Eulerian-Lagrangian formulation. Details can be found in Section 3.2. Changes in plate stiffness due to deformation are included by way of a stress-stiffening matrix based on von Karman plate theory. Details can be found in Section 3.3.

3.2 Large Deformation and the Mixed Eulerian-Lagrangian Formulation

In flutter problems, the deformation of the flexible wing structure is often large enough to affect the direction of both aerodynamic and internal elastic restoring forces. The deformation does not necessarily result in large strains, but instead, is mainly composed of rigid body motions. These large deformations can be represented in simulations as geometric nonlinearity, and have been modeled accurately in previous studies using an updated element coordinate system approach.

The updated coordinate system approach used in this work is identified as the mixed Eulerian-Lagrangian formulation since it uses a single global, fixed coordinate system which is Eulerian along with multiple element-attached, local, moving coordinate systems which are Lagrangian. The deformation of each element in its local coordinate system is Eulerian. In this sense, the Eulerian and Lagrangian descriptions of the deformation are truly “mixed.” Though sometimes identified

by different names, similar formulations have been used by others working in structural mechanics, structural dynamics, and aeroelasticity [64, 65, 117–121].

A triangular element is shown in Figure 3.2 along with the global and local coordinate systems. Each structural finite element has its own local coordinate system that moves with it throughout the simulation. The global coordinate system is defined by the unprimed axes x , y , and z while the local coordinate systems are defined by the primed axes x' , y' , and z' .

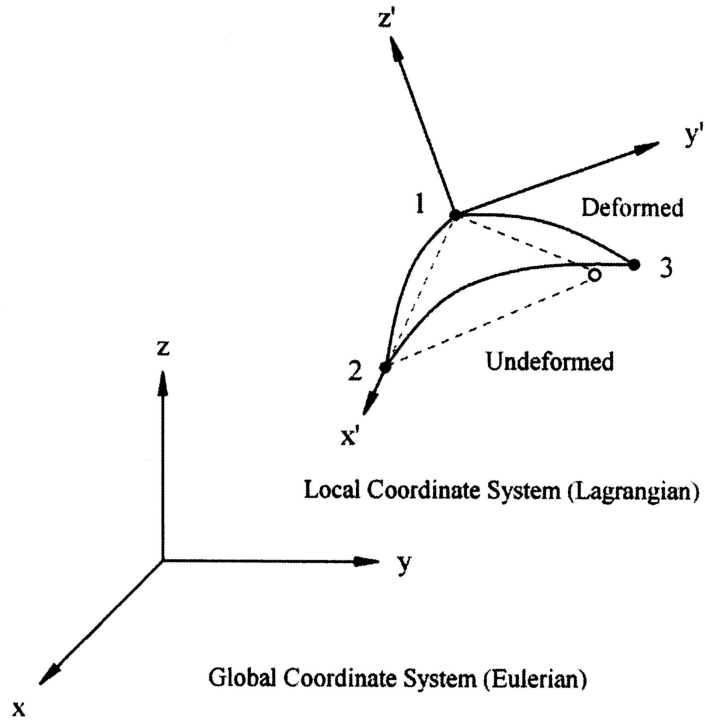


Figure 3.2: Eulerian and Lagrangian coordinate systems.

By carefully selecting the orientation of the local coordinate system relative to its finite element, local displacement computations can be simplified. As shown in Figure 3.2, the nodes of the element lie in the $x' - y'$ plane. A unit vector in the z' direction, expressed in the global coordinate system, can be calculated using

Equation 3.1.

$$\hat{\mathbf{e}}_{z'} = \frac{\mathbf{R}_{21} \times \mathbf{R}_{31}}{|\mathbf{R}_{21} \times \mathbf{R}_{31}|} \quad (3.1)$$

Here, \mathbf{R}_{21} and \mathbf{R}_{31} are vectors in the $x' - y'$ plane expressed in the global coordinate system.

$$\mathbf{R}_{21} = (x_2 - x_1)\hat{\mathbf{e}}_x + (y_2 - y_1)\hat{\mathbf{e}}_y + (z_2 - z_1)\hat{\mathbf{e}}_z \quad (3.2a)$$

$$\mathbf{R}_{31} = (x_3 - x_1)\hat{\mathbf{e}}_x + (y_3 - y_1)\hat{\mathbf{e}}_y + (z_3 - z_1)\hat{\mathbf{e}}_z \quad (3.2b)$$

The x' axis is chosen to be aligned on the finite element edge defined by nodes 1 and 2. The unit vector in the x' direction can be calculated using Equation 3.3.

$$\hat{\mathbf{e}}_{x'} = \frac{\mathbf{R}_{21}}{|\mathbf{R}_{21}|} \quad (3.3)$$

Once the x' and z' axes have been defined, the y' axis can be calculated as the cross product of the other two unit vectors.

$$\hat{\mathbf{e}}_{y'} = \hat{\mathbf{e}}_{z'} \times \hat{\mathbf{e}}_{x'} \quad (3.4)$$

Any vector can be expressed in either the local or the global coordinate system. The relationship between the coordinate systems is defined by the orthogonal transformation defined in Equations 3.5a and 3.5b,

$$\mathbf{q}^L = [e] \mathbf{q}^G \quad (3.5a)$$

$$[e] = \begin{bmatrix} (e_{x'})_1 & (e_{x'})_2 & (e_{x'})_3 \\ (e_{y'})_1 & (e_{y'})_2 & (e_{y'})_3 \\ (e_{z'})_1 & (e_{z'})_2 & (e_{z'})_3 \end{bmatrix} \quad (3.5b)$$

where \mathbf{q}^L and \mathbf{q}^G are vectors expressed in local and global coordinates, and $[e]$ is an orthogonal direction cosine matrix that is constructed from the components of the local unit vectors $\hat{\mathbf{e}}_{x'}$, $\hat{\mathbf{e}}_{y'}$, and $\hat{\mathbf{e}}_{z'}$ expressed in the global coordinate system.

Because of the careful selection of the local coordinate system orientation, several local degrees of freedom are removed from the finite element formulation. The full set of local in-plane degrees of freedom is represented by

$$\mathbf{q}'_{uv}{}^T = \left\{ u'_1 \quad v'_1 \quad u'_2 \quad v'_2 \quad u'_3 \quad v'_3 \right\} \quad (3.6a)$$

As shown in Figure 3.3, \mathbf{q}'_{uv} reduces to

$$\mathbf{q}'_{uv}{}^T = \left\{ 0 \quad 0 \quad u'_2 \quad 0 \quad u'_3 \quad v'_3 \right\} \quad (3.6b)$$

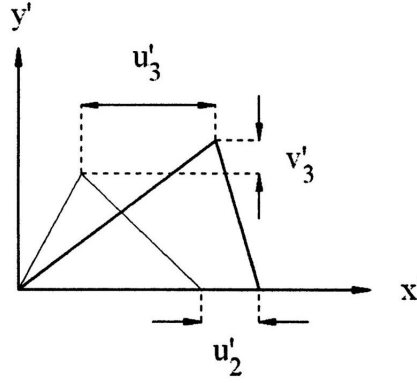


Figure 3.3: Local in-plane deformation.

Similarly, the local out-of-plane degrees of freedom shown in Figure 3.4 are reduced from

$$\mathbf{q}'_w{}^T = \left\{ w'_1 \quad \beta_{x'_1} \quad \beta_{y'_1} \quad w'_2 \quad \beta_{x'_2} \quad \beta_{y'_2} \quad w'_3 \quad \beta_{x'_3} \quad \beta_{y'_3} \right\} \quad (3.7a)$$

to

$$\mathbf{q}'_w{}^T = \left\{ 0 \quad \beta_{x'_1} \quad \beta_{y'_1} \quad 0 \quad \beta_{x'_2} \quad \beta_{y'_2} \quad 0 \quad \beta_{x'_3} \quad \beta_{y'_3} \right\} \quad (3.7b)$$

It is important to note that because of the mixed Eulerian-Lagrangian formulation, local deformations are not affected by element rigid body motion. The out-of-plane displacement degrees of freedom, w'_1 , w'_2 , and w'_3 , are completely

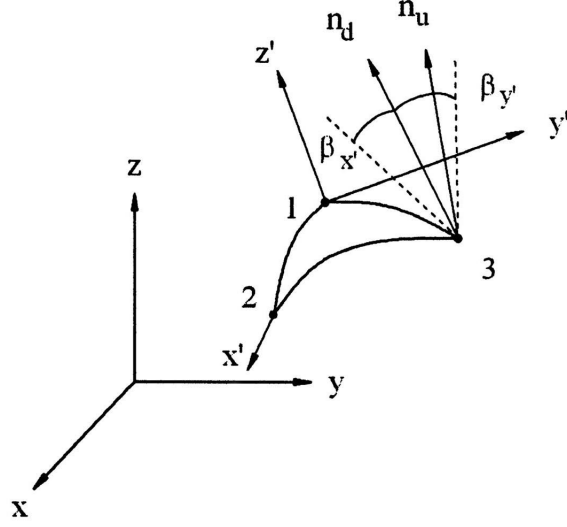


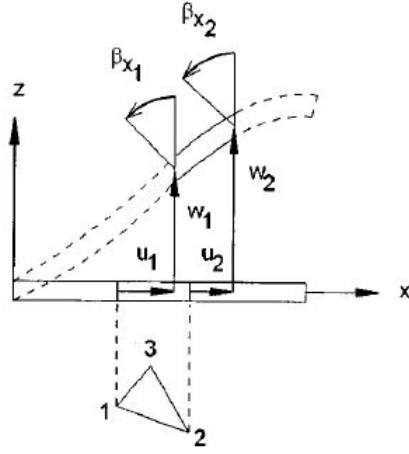
Figure 3.4: Local out-of-plane deformation.

eliminated. This allows for correct calculation of the local element elastic forces including stress-stiffening. Without the removal of the rigid body motion, stiffening due to coupling between the in-plane and out-of-plane deformation would be overestimated, leading to an overly stiff structural model [49]. Figure 3.5 shows the deformation kinematics for the classical formulation, where rigid body motion contributes to elastic deformation, and the mixed Eulerian-Lagrangian formulation, where rigid body motion is not included in the elastic deformation. It should also be noted that the deformations shown in Figures 3.2, 3.3, and 3.4 are exaggerated. The nonlinearity in the structural model is limited to elastic material behavior.

Calculation of the in-plane displacements is straightforward and given by Equations 3.8a-c where coordinates with a d subscript represent the deformed nodal locations.

$$u'_2 = (x'_2)_d - x'_2 \quad u'_3 = (x'_3)_d - x'_3 \quad v'_3 = (y'_3)_d - y'_3 \quad (3.8a-c)$$

Classical Displacement Kinematics



Mixed Eulerian-Lagrangian Kinematics

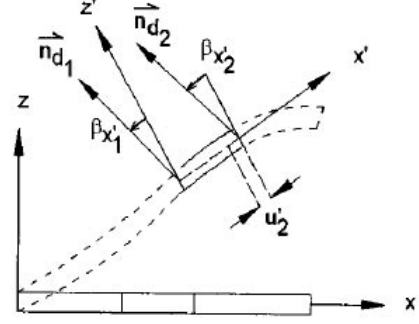


Figure 3.5: Classical and mixed Eulerian-Lagrangian kinematics.

The calculation of the local out-of-plane rotational deformations in \mathbf{q}'_w can be determined using the finite rotation relation

$$\mathbf{n}_d^{t^{n+1}} = \cos(\Delta\phi) \mathbf{n}_d^{t^n} + (1 - \cos(\Delta\phi)) (\mathbf{n} \cdot \mathbf{n}_d^{t^n}) \mathbf{n} + \sin(\Delta\phi) (\mathbf{n} \times \mathbf{n}_d^{t^n}) \quad (3.9a)$$

$$\Delta\phi = \sqrt{(\Delta\beta_x)^2 + (\Delta\beta_y)^2 + (\Delta\beta_z)^2} \quad (3.9b)$$

$$\mathbf{n} = (-\Delta\beta_y \mathbf{i} + \Delta\beta_x \mathbf{j} + \Delta\beta_z \mathbf{k}) \quad (3.9c)$$

where $\Delta\phi$ is the magnitude of the rotation, \mathbf{n} is the normalized rotation vector, and $\mathbf{n}_d^{t^n}$ and $\mathbf{n}_d^{t^{n+1}}$ are the deformed unit vectors at times t^n and t^{n+1} expressed in the global coordinate system. Local rotational deformations can be calculated in terms of the deformed unit vector $(\mathbf{n}_d^{t^n})^L$ expressed in local coordinates using the relationships in Equations 3.10a and 3.10b.

$$\beta_{x'} = (\mathbf{n}_d^{t^n})^L_x \frac{\cos^{-1} \left((\mathbf{n}_d^{t^n})^L_z \right)}{\sqrt{\left((\mathbf{n}_d^{t^n})^L_x \right)^2 + \left((\mathbf{n}_d^{t^n})^L_y \right)^2}} \quad (3.10a)$$

$$\beta_{y'} = (\mathbf{n}_d^{t^n})_y \frac{\cos^{-1} \left((\mathbf{n}_d^{t^n})_z \right)}{\sqrt{\left((\mathbf{n}_d^{t^n})_x \right)^2 + \left((\mathbf{n}_d^{t^n})_y \right)^2}} \quad (3.10b)$$

3.3 Components of the Structural Model

The deformation of a wing due to flutter is often large enough to alter the stiffness of the structure. When the in-plane stresses become large enough, the out-of-plane bending stiffness increases. This nonlinear stress-stiffening effect has been modeled accurately in many problems of structural mechanics and dynamics using variations of the von Karman plate theory, which accounts for changes in stiffness due to in-plane forces generated by out-of-plane deflection. As mentioned in Section 3.2, the classical von Karman formulation tends to overestimate the stiffening when large deformations occur. This problem is removed when deformation is calculated in an element attached, Lagrangian coordinate system as is done in the present work.

The nonlinear structural model uses three finite elements to model in-plane, out-of-plane, and nonlinear stress-stiffening behavior as shown in Figure 3.6. This finite element formulation has been used in numerous computational flutter simulations and has shown good correlation with experimental results [11, 49–51, 53–56]. The three finite elements used in the formulation are

1. A discrete shear triangle (DST) originally developed in Reference 122 and extended to dynamics problems in Reference 123 is used to model the out-of-plane bending and transverse shear behavior. This is a three node, nine degree of freedom triangular plate bending element with proper rank that is

valid for thin to thick plates. The formulation, based on Mindlin-Reissner plate theory, is free of shear locking and coincides with the discrete Kirchhoff triangular element if transverse shear deformation becomes negligible.

2. A constant strain triangle (CST) [124] is used to model the in-plane behavior.
3. A stress-stiffening matrix based on von Karman plate theory is used to model the in-plane and out-of-plane coupling effects. It assumes that nonlinear strains due to in-plane displacement can be neglected because they are small relative to nonlinear strains due to transverse displacement gradients. These assumptions limit the use of the element to small deformations. This is not a problem when the mixed Eulerian-Lagrangian formulation is employed because deformations in the local, element fixed coordinate system do not include rigid body motion, as detailed in Section 3.2 .

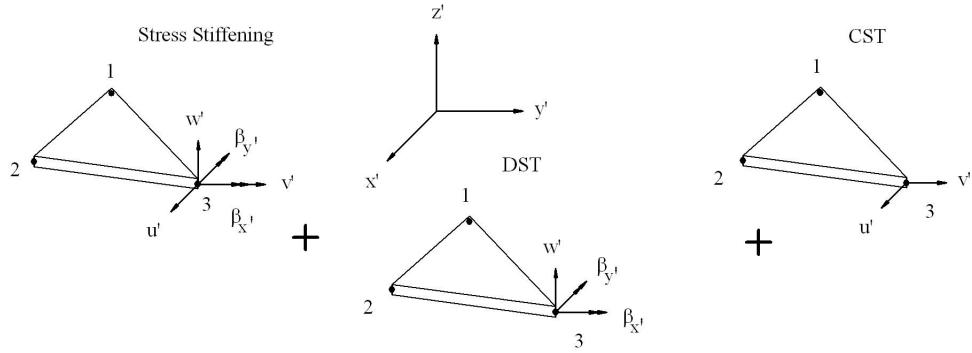


Figure 3.6: Components of the nonlinear structural finite element.

The total strain energy of an element, U_{total}^e , can be calculated as the sum of the strain energies due to membrane stretching of the CST, U_m , out-of-plane bending, U_b , and shear, U_s , from the DST, nonlinear coupling of in-plane and out-of-plane motions, U_{N1} , and the nonlinear coupling effect of slope due to large

deflections, U_{N2} .

$$U_{total}^e = U_m + U_b + U_s + U_{N1} + U_{N2} \quad (3.11)$$

The individual strain energy contributions from Equation 3.11 are given in Equations 3.12a–3.12e,

$$U_m = \frac{1}{2} \int_A \{\varepsilon_m\}^T [D_m] \{\varepsilon_m\} dA \quad (3.12a)$$

$$U_b = \frac{1}{2} \int_A \{\kappa\}^T [D_b] \{\kappa\} dA \quad (3.12b)$$

$$U_s = \frac{1}{2} \int_A \{\gamma\}^T [D_s] \{\gamma\} dA \quad (3.12c)$$

$$U_{N1} = \int_A \{\varepsilon_m\}^T [D_m] \{\varepsilon_n\} dA \quad (3.12d)$$

$$U_{N2} = \int_A \{\varepsilon_n\}^T [D_m] \{\varepsilon_n\} dA \quad (3.12e)$$

where U_m , U_b , and U_s are quadratic-order, U_{N1} is cubic-order, and U_{N2} is quartic-order. The constitutive relations are given by Equations 3.13a–d. In Equation 3.13d, k is the shear correction factor, which is $\frac{5}{6}$ for isotropic plates.

$$[D] = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (3.13a)$$

$$[D_m] = h[D] \quad (3.13b)$$

$$[D_b] = \frac{h^3}{12}[D] \quad (3.13c)$$

$$[D_s] = khG \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.13d)$$

The strain displacement relationships are defined in Equations 3.14a–d. Note the nonlinear bending terms in 3.14c which contribute to the cubic-order and quartic-order strain energy expressions in Equations 3.12d and 3.12e respectively.

$$\{\varepsilon_m\}^T = \left\{ \frac{\partial u'}{\partial x'} \quad \frac{\partial v'}{\partial y'} \quad \frac{\partial v'}{\partial x'} + \frac{\partial u'}{\partial y'} \right\} \quad (3.14a)$$

$$\{\gamma\}^T = \left\{ \frac{\partial w'}{\partial x'} + \beta_{x'} \quad \frac{\partial w'}{\partial y'} + \beta_{y'} \right\} \quad (3.14b)$$

$$\{\varepsilon_n\}^T = \left\{ \frac{1}{2}\beta_{x'}^2 \quad \frac{1}{2}\beta_{y'}^2 \quad \beta_{x'}\beta_{y'} \right\} \quad (3.14c)$$

$$\{\kappa\}^T = \left\{ \frac{\partial \beta_{x'}}{\partial x'} \quad \frac{\partial \beta_{y'}}{\partial y'} \quad \frac{\partial \beta_{x'}}{\partial y'} + \frac{\partial \beta_{y'}}{\partial x'} \right\} \quad (3.14d)$$

The displacements are defined as follows:

$$u' = u'_0(x', y', t) + z'\beta_{x'}(x', y', t) \quad (3.15a)$$

$$v' = v'_0(x', y', t) + z'\beta_{y'}(x', y', t) \quad (3.15b)$$

$$w' = w'(x', y', t) \quad (3.15c)$$

The principle of stationary potential energy can be used with the total strain energy from Equation 3.11 to calculate the complete element stiffness matrix. The stiffness matrix is organized as shown in Figure 3.7.

$$[\mathbf{K}_e]_{15 \times 15} = \begin{bmatrix} \begin{bmatrix} \text{out-of-plane} \\ \text{partition} \\ \text{(DST)} \end{bmatrix}_{9 \times 9} & \begin{bmatrix} \text{coupling} \\ \text{partition} \\ \text{(nonlinear)} \end{bmatrix}_{9 \times 6} \\ \begin{bmatrix} \text{coupling} \\ \text{partition} \\ \text{(nonlinear)} \end{bmatrix}_{6 \times 9} & \begin{bmatrix} \text{in-plane} \\ \text{partition} \\ \text{(CST)} \end{bmatrix}_{6 \times 6} \end{bmatrix}$$

Figure 3.7: Partitions of an element stiffness matrix.

Each structural finite element has two element stiffness matrices associated with it; these are the tangent stiffness matrix, $[K_t]$, and the secant stiffness matrix, $[K_s]$. The tangent stiffness matrix is used in the calculation of the incremental deformation in the global system. The secant stiffness matrix is used to calculate element elastic forces in the local coordinate system. These forces are then

transformed to the global coordinate system to form the global elastic force vector. Details of the use of the tangent and secant stiffness matrices are included in Chapter 4. For details on the derivation of these matrices, see Seber [51] and Rajasekaran and Murray [125].

3.4 Finite Element Model of the Wing

The nonlinear triangular plate finite elements described in Section 3.3 are assembled to form the full structural wing model as shown in Figure 3.8c. The model is fully clamped along the edge closest to the wing root, with displacements and rotations set to zero. The structural model root does not necessarily correspond to the aerodynamic wing root. By moving the structural root outboard, the portion of the wing inboard of the structural root will not deform during simulations, making the structure rigid in this area. This is an important improvement over previous versions of the code, as the inboard sections of some cranked wings are much stiffer than the outboard portions. Assigning a high stiffness to these areas can produce very high natural frequencies which can slow down the simulation or generate high frequency oscillations that can corrupt the solution.

The discretization of the wing is controlled by two vectors: spanwise element edge locations and chordwise element edge locations. Previous versions of the code split the wing into evenly spaced regions to create the finite element model. By using the vectors, the wing can be discretized as desired with more elements concentrated in some areas than others. From the spanwise and chordwise vectors, the following parameters, which describe the discretization, are defined:

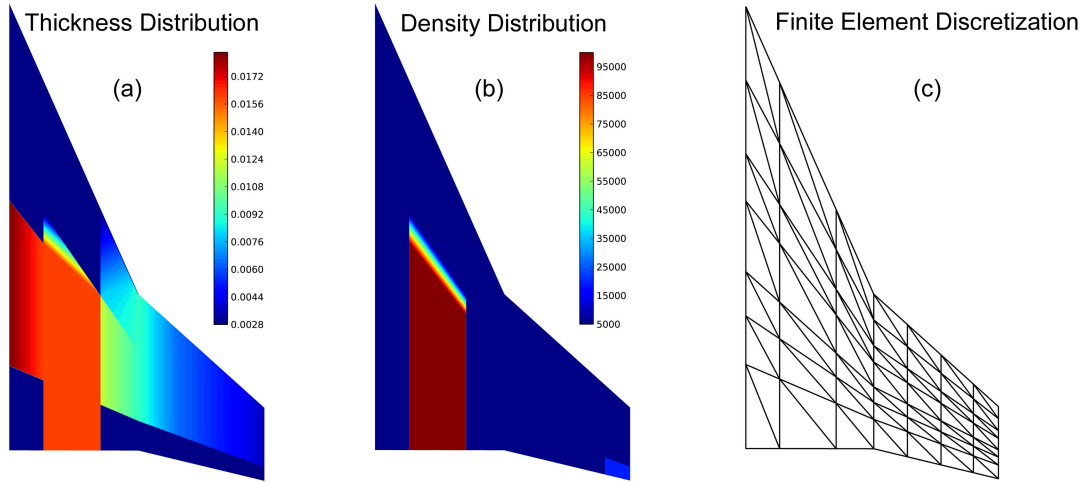


Figure 3.8: The (a) thickness distribution, (b) density distribution, and (c) finite element discretization of the NEXST-2 wing structural model.

1. N_{xe} is the number of chordwise elements. This number is the same for both the inboard and outboard sections of the wing.
2. N_{ye1} is the number of spanwise elements on the inboard section of the wing.
3. N_{ye2} is the number of spanwise elements on the outboard section of the wing.

The total number of finite elements in the wing model is $2 \times N_{xe} \times (N_{ye1} + N_{ye2})$. For the wing shown in Figure 3.8c, $N_{xe} = 7$, $N_{ye1} = 3$, and $N_{ye2} = 4$. The total number of elements is 98. The planform of the structural wing model is defined by the fluid mesh parameters discussed in Chapter 2. This ensures that the structural and fluid models match each other and that no gaps exist between the two. The wing discretization parameters are set in the aeroelastic program input files and can be changed as needed by the analyst. All the parameters can easily be set using dfLap (Appendix A).

The discretization occurs along lines of constant percent chord and constant

percent span. This allows an even distribution of elements in the presence of kink and/or tip sweep (when the kink/tip line(s) are not parallel to the root). The percent chord/span calculations on the inboard and outboard sections of the wing are kept independent of each other to simplify the formulation.

The mass and stiffness distribution over the wing can be controlled by adjusting the plate thickness and plate density distribution. Both thickness and density can be defined at any point on the wing. The properties must be defined at the leading and trailing edge root, kink, and tip at a minimum. Bilinear interpolation is used to determine the values between the defined locations. An example of the thickness and mass distribution for the NEXST-2 SST wing model is shown in Figure 3.8a-b. For details on how these parameters are input to the code, see the dfLap User's Guide (Appendix A).

CHAPTER 4

Fluid-Structure Coupling by the Direct Eulerian-Lagrangian Approach

4.1 Introduction

In this chapter, the method used to couple the computational fluid dynamic model from Chapter 2 to the structural dynamic model from Chapter 3 is presented. In computational aeroelasticity, a great deal of attention must be paid to the structural dynamic and fluid dynamic models, as their accuracy is of the utmost importance. The coupling of the two models is often treated as if its importance was secondary. This is problematic, as it has been shown that the accuracy of the coupling between the fluid and the structure can have a large effect on the solution [9–11]. It is at the fluid-structure boundary where energy is extracted from the flow and imparted to the wing, and, at the most basic level, energy transfer between the fluid and the structure is the process that determines the response of the system. Accurate modeling of the energy transfer is dependent on the coupling of the fluid and structural systems, and may be the most important numerical consideration in any aeroelastic simulation. The coupling approach presented here is a modified version of that used in References 4, 9–11, 37–42, 44–51, 53–56, 126.

Many aeroelastic codes employ classical methods that involve using separate codes to model the fluid and the structure. The two independent solutions are then coupled through the kinematic boundary conditions at the fluid-structure boundary. There are two problems with this approach: time consistency and boundary location accuracy.

To accurately calculate the energy exchange between the fluid and the structure, solutions of the two systems must move forward together in time. Time consistency of the classical methods is complicated by the fact that enforcement of the boundary conditions requires knowledge of the position and velocity of the fluid-structure boundary, but the position and velocity of the boundary are not known until the boundary conditions are used to calculate the surface pressures and apply forces to the structure for the current time step. To manage the problem, approximations are used. Usually the pressure from the previous time step is extrapolated forward in time and applied to the system. In 1991, Bendiksen showed the errors that approximations like this can cause [9]. Despite this, classical methods remain popular.

In addition to time consistency, the location of the moving fluid-structure boundary must be modeled accurately if the energy exchange between the fluid and the structure is to be calculated correctly. In classical aeroelastic simulations, the location of the boundary is often estimated using interpolation functions. The danger here is that the estimated boundary may not be close enough to the true boundary to calculate accurate pressures, forces (both in magnitude and direction), or shock locations. Errors like this can lead to spurious energy generation at the fluid-structure boundary and a violation of the conservation laws.

4.2 Direct Eulerian-Lagrangian Coupling

The Direct Eulerian-Lagrangian method of coupling the fluid and structural models has proven to be an accurate way of modeling the energy exchange across the fluid-structure boundary [4,9–11,37–42,44–51,53–56,126]. To avoid errors involved with the location of the boundary, the structural finite element shape functions, which are used to calculate the deformation of the wing structure, are also used to define the location of the boundary. This results in the fluid-structure boundary being consistent with the finite element modeling assumptions. The time consistency problem is solved by modeling the fluid-structure system as a single continuum dynamics problem at the element level. This is accomplished by formulating the equations of motion in mixed Eulerian-Lagrangian coordinates and switching from an Eulerian to a Lagrangian description at the fluid-structure boundary. When this is done, the boundary conditions are satisfied as the conservation laws are integrated, and no time-lag is introduced to the system. The Direct Eulerian-Lagrangian method is depicted in Figure 4.1.

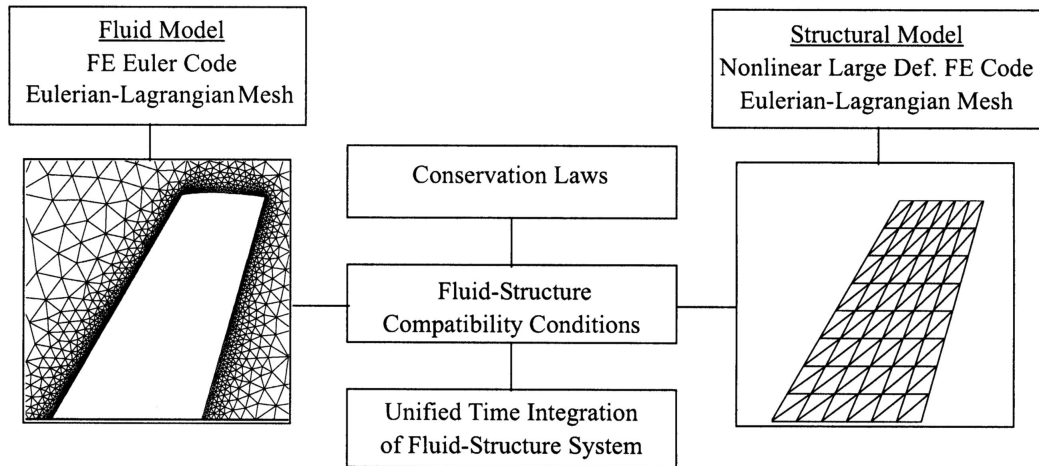


Figure 4.1: The Direct Eulerian-Lagrangian approach to fluid-structure coupling.

4.2.1 Boundary Location

Representing the location of the fluid-structure boundary accurately should be a primary goal of any coupling method. In the Direct Eulerian-Lagrangian method, the finite element shape functions are used to define the boundary. When this is done, consistency with the finite element modeling assumptions is maintained, and the conservation laws are not violated. When spline or interpolation functions are used to represent the boundary, consistency with the finite element modeling assumptions can be broken which can lead to a violation of the conservation laws.

Before the simulation is started, each fluid mesh node on the wing surface is pinned to a structural element with a fixed offset, Δz , with respect to the wing mid-surface. The offset is defined by the thickness of the wing. The location of the fluid node inside the structural finite element is defined in area coordinates as depicted in Figure 4.2. The area coordinates for each node do not change during the simulation, even when the element deforms under loading.

When the structural finite elements deform, the finite element shape functions are used with the area coordinates to calculate the exact deformation at each aerodynamic node. The new location of the aerodynamic nodes are then calculated as illustrated in Figures 4.3 and 4.4. The new location of the nodes is calculated in the structural element's local coordinate system. The coordinate transformation given in Equations 3.5a and 3.5b are used to calculate the location in the global coordinate system.

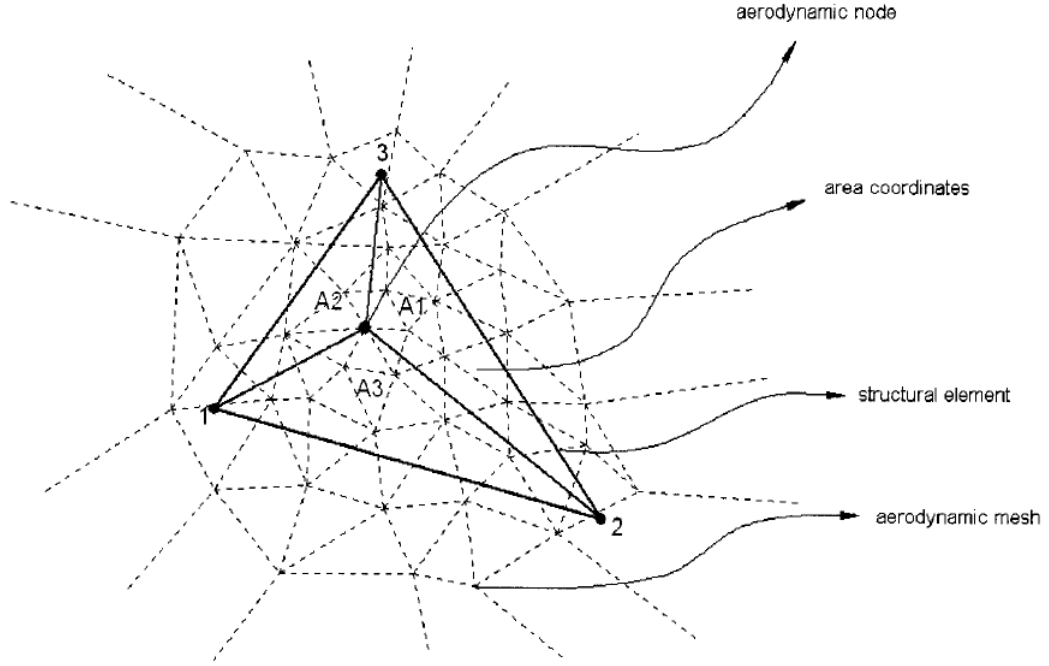


Figure 4.2: Aerodynamic node coupled to structural finite element with location defined by area coordinates.

4.2.2 Time Consistency

It is important in fluid-structure interaction simulations that the fluid and structural models be integrated in time consistently. Time lag between the two systems can result in convergence to an incorrect solution [10, 11].

In Equation 2.24 from Chapter 2, the tangent flow boundary condition for the fluid-structure boundary was established. By applying this condition directly to the conservation equations for the fluid, and by calculating the mesh velocity directly from the finite element shape functions, time lag is removed from the solution process. In classical coupling schemes, the boundary location must be estimated at the current time step based on motion from previous steps. In the Direct Eulerian-Lagrangian method, this estimation is not necessary, as the

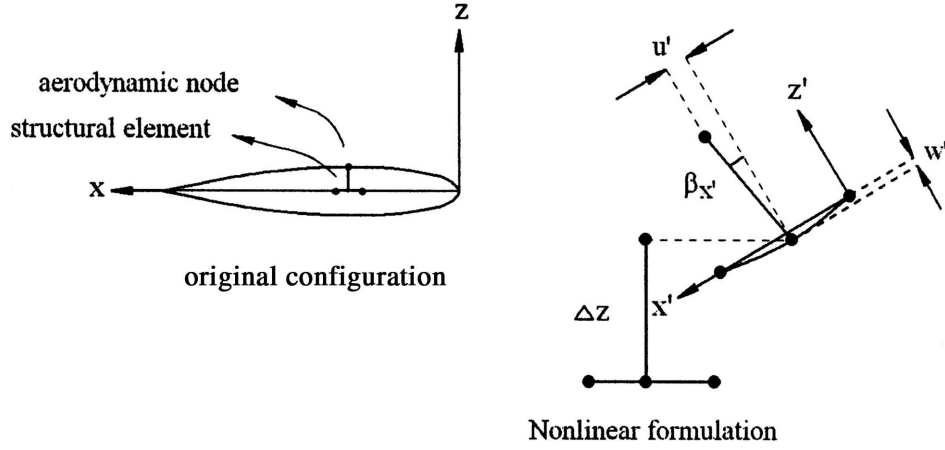


Figure 4.3: Two-dimensional representation of fluid-structure boundary deformation.

location of the boundary is solved for exactly at the current time step as part of the conservation equations.

4.2.3 Consistent Load Vector

For aeroelastic problems where large deformation is present, the direction of forces acting on a structure can be altered by the deformation. These kinds of forces are often referred to as follower forces. In the Direct Eulerian-Lagrangian method, follower forces are modeled through a consistent load vector approach, as shown in Figure 4.5.

To apply the aerodynamic loads as follower forces, pressures are calculated on the deformed fluid-structure boundary in the local structural finite element coordinate system and then transformed to forces in the global system. Before the simulation is started, Gaussian quadrature points on each structural element are assigned to a fluid element on the boundary. The location of the quadrature

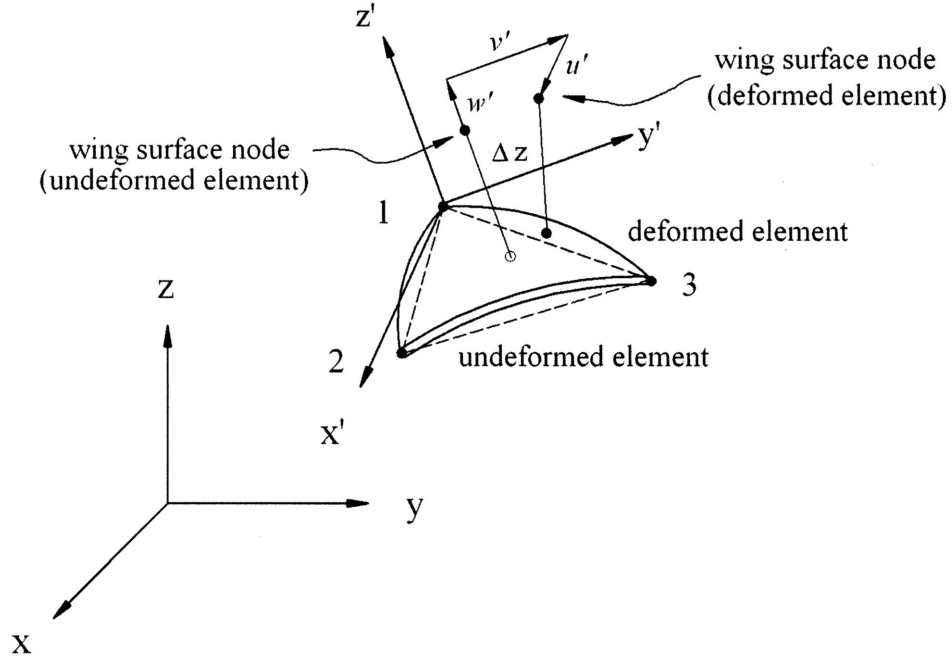


Figure 4.4: Three-dimensional representation of fluid-structure boundary deformation.

point in the fluid element is represented in area coordinates (Figure 4.6). These coordinates do not change during the analysis. At each step in the analysis, the pressure is calculated in the local coordinate system at the quadrature points by linear interpolation of the nodal pressures from the appropriate fluid elements. The pressure is transformed to global coordinates using Equations 3.5a and 3.5b.

4.2.4 Conservation of Energy

A discussion of aeroelastic fluid-structure coupling would not be complete without mentioning conservation of energy across the fluid-structure boundary. Both the fluid and structural mathematical models are based on the principle that en-

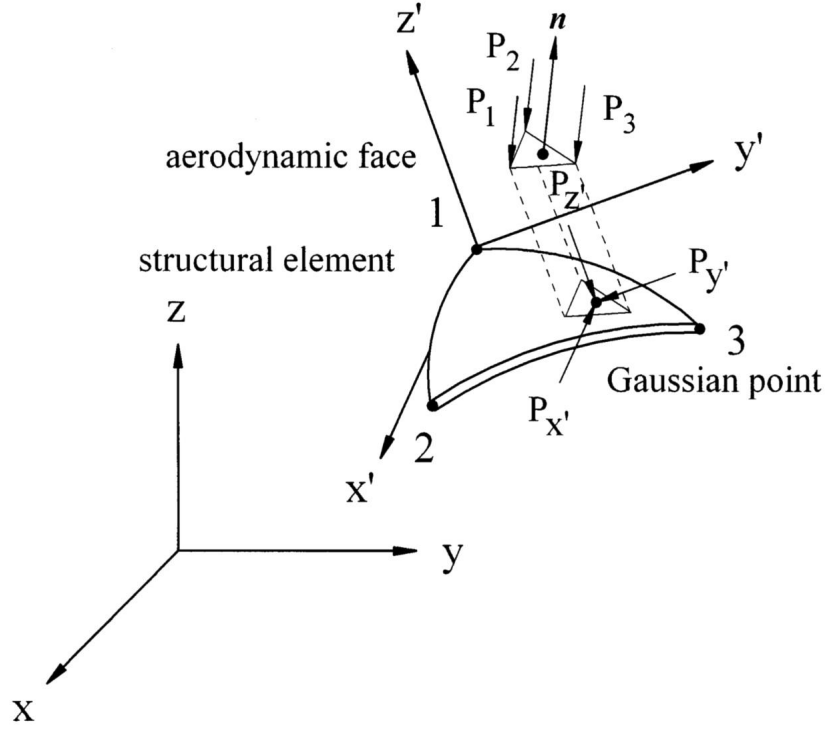


Figure 4.5: Consistent load vector calculation.

ergy is conserved. However, when these models are coupled together incorrectly, the conservation laws can be violated. Bendiksen et al. have shown that the Direct Eulerian-Lagrangian coupling method accurately satisfies the principle of conservation of energy by comparing the kinetic and strain energy in the structure to the work done by the fluid on the structure throughout the simulation time [4, 9–11, 37, 49, 50]. If energy is conserved (in the absence of structural damping), the difference between the energy and the work will be constant and equal to the initial energy in the structure. If an aeroelastic code cannot show that the kinetic and strain energy in the wing are only generated by the work that the fluid does on the structure, the validity of the code should be questioned. Figure 4.7 shows the conservation of energy during limit cycle flutter of the NEXST-2 wing. Additional details on this wing and the response are included in Chapter 6.

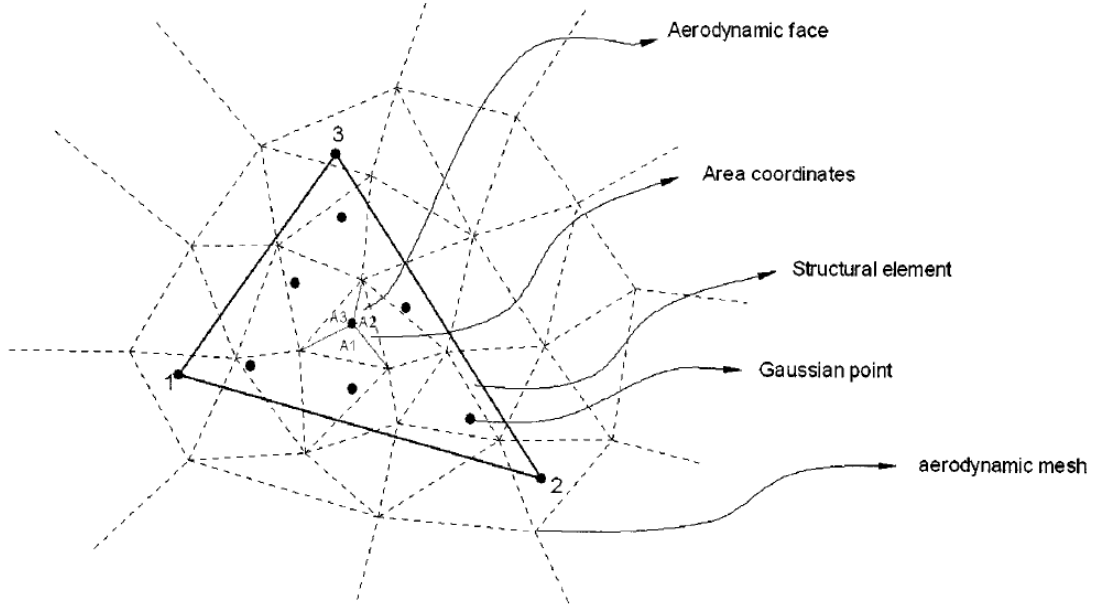


Figure 4.6: Gaussian quadrature points and fluid elements.

4.3 Solution Procedure

The solution procedure for the static and flutter aeroelastic solutions are discussed in this section. The static solution uses the flow conditions from a steady aerodynamic solution as initial conditions. The flutter solution uses the flow conditions, wing deformation, and structural state from the static solution as initial conditions. Before the time integration of either solution is started, the structural model is built, and the assignment of fluid nodes to structural elements (Figure 4.2) and structural Gaussian force application points to fluid elements (Figure 4.5) is completed.

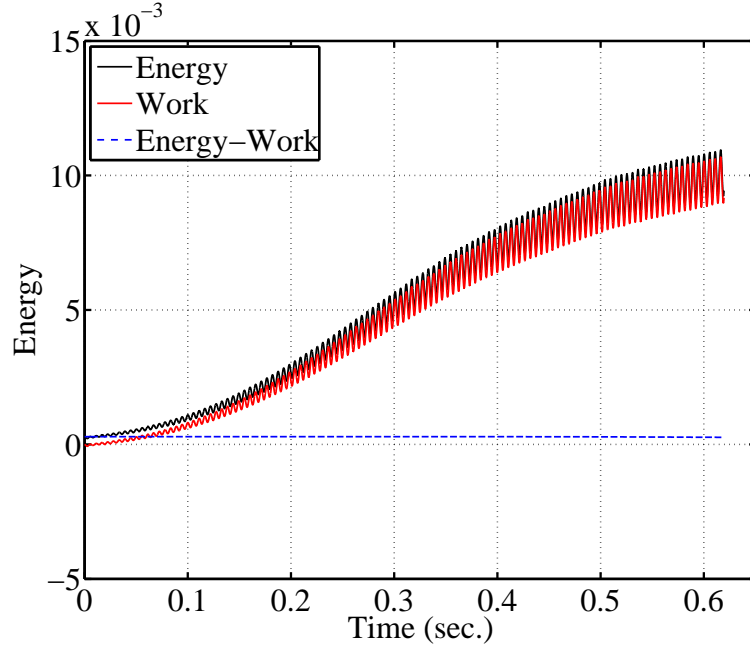


Figure 4.7: Conservation of energy during flutter simulation of the NEXST-2 wing.

4.3.1 Static Aeroelastic Solution Procedure

The static aeroelastic solution uses a four-stage explicit Runge-Kutta scheme for the fluid solution and Newton-Raphson iterations for the structure as follows:

1. Calculate the global tangent stiffness matrix, $[K_t]$, for the structure.
2. Calculate the global aerodynamic load vector, $\{F_{aero}\}$.
3. Calculate the linear displacements of the structure, $\{q_t\}$ using the tangent stiffness matrix and aerodynamic load vector.

$$[K_t] \{q_t\} = \{F_{aero}\} \quad (4.1)$$

4. Calculate the updated system geometry based on the linear displacements.

5. Calculate the local element coordinate systems and deformation.
6. Calculate the local element elastic force vectors by multiplying the element secant stiffness matrices, $[K_s]$, by the local element deformation vectors from Equations 3.6b and 3.7b.
7. Transform the local element elastic force vectors into the global elastic force vector using the transformation from Equations 3.5a and 3.5b.
8. Calculate the force unbalance that will drive the structure to equilibrium by taking the difference between the global aerodynamic load vector and the global elastic force vector.
9. Calculate the incremental displacements using the global tangent stiffness matrix and the force unbalance.
10. Add the incremental displacements to the current shape of the structure to get the new coordinate of the structural nodes.
11. Update the deformed nodal vectors with the incremental rotations.
12. Calculate the updated local structural coordinates systems.
13. Calculate the new location of the fluid nodes on the wing using the local structural coordinate systems and the incremental rotations.
14. Calculate the position of the remainder of the fluid nodes using the mesh motion algorithm.
15. If this is the first or second stage of the Runge-Kutta scheme, calculate the adaptive dissipation vector.

16. Solve the right hand side of the Runge-Kutta equation (2.7) for the appropriate stage.
17. Update the conserved quantities of the flow (Equation 2.1b).

Each of these steps is repeated for each stage of the Runge-Kutta scheme. All four stages of the Runge-Kutta scheme are run for each time step.

4.3.2 Dynamic Aeroelastic Solution Procedure

The dynamic aeroelastic solution uses a five-stage explicit Runge-Kutta scheme to solve the second order differential equation,

$$[M]\{\ddot{q}_t\} + [K]\{q_t\} = \{F\} \quad (4.2)$$

which describes the motion of linear structures in the absence of damping. Here, $[M]$ and $[K]$ are the structural system global mass and stiffness matrices, $\{q_t\}$ is the global displacement vector, $\{F\}$ is the global external force vector, and overdots represent derivatives with respect to time. All the terms vary through time with the exception of the mass matrix $[M]$. Bathe et al. showed that this matrix does not change as a function of time, so it only needs to be calculated once at the onset of the simulation [117]. The external forces on the system are a combination of aerodynamic forces and forces due to damping which can be viscous or hysteretic.

The second term in Equation 4.2 represents the global elastic force vector, $\{F_{elastic}\}$ mentioned in step 7 from Section 4.3.1. By moving this term to the right hand side, the equation becomes

$$[M]\{\ddot{q}_t\} = \{F\} - \{F_{elastic}\} \quad (4.3)$$

The solution of Equation 4.3 proceeds as follows:

1. The current state of the structure (acceleration, velocity, displacement) either comes from the previous time step/Runge-Kutta stage or is defined as an initial condition.
2. Calculate the global aerodynamic load vector, $\{F_{aero}\}$.
3. Calculate the global element elastic force vectors as was done in steps 6 and 7 for the static solution.
4. Calculate any damping based on the current velocity and displacement of the system. Include it as a component of the external applied forces, $\{F\}$.
5. Solve Equation 4.3 for the accelerations by multiplying the inverted mass matrix by the terms on the right hand side of the equation.
6. Calculate the appropriate velocities and displacements by integrating the accelerations.
7. Update the deformed nodal vectors with the incremental rotations.
8. Calculate the updated local structural coordinates systems.
9. Calculate the new location of the fluid nodes on the wing using the local structural coordinate systems and the incremental rotations.
10. Calculate the position of the remainder of the fluid nodes using the mesh motion algorithm.
11. If this is the first or second stage of the Runge-Kutta scheme, calculate the adaptive dissipation vector.

12. Solve the right hand side of the Runge-Kutta equation (2.7) for the appropriate stage.
13. Update the conserved quantities of the flow (Equation 2.1b).

Each of these steps is repeated for each stage of the Runge-Kutta scheme. All five stages of the Runge-Kutta scheme are run for each time step.

CHAPTER 5

Aeroelastic Model Validation

5.1 Introduction

The present research is a continuation of the work of Hwang [37, 44–46], Seber [51], and Bendiksen [10, 11, 47–50, 53–56, 126], and the computer codes used here are modified versions of their codes. To establish confidence in the present work, comparisons are made with published results. Structural dynamic free vibration, steady CFD, static aeroelastic, and transonic flutter results are presented for the ONERA M6 wing. This wing is often used in the validation of CFD and aeroelastic codes due to a well documented database of wind tunnel test results [43].

Free vibration results are discussed in Section 5.2 and consist of normal modes and fundamental frequencies of the linear structural model. Section 5.3 contains comparisons of steady CFD pressure and Mach distributions. Static aeroelastic results are compared in Section 5.4. Dynamic aeroelastic results can be found in Section 5.5. The current set of results correlate well with both experimental and computational published studies, thus validating the computer codes used in the present work.

5.2 Linear Free Vibration Results

The structural model of the wing consists of a thin aluminum alloy plate whose planform matches that of the wing. The plate is covered by balsa wood or flexible plastic foam to create the desired airfoil shape without adding mass or stiffness. The plate is modeled using triangular finite elements. Details on the finite element formulation are given in Chapter 3.

The structural model is a modified version of that used by Bendiksen, Hwang, and Seber [10, 11, 37, 44–51, 53–56, 126] and should be able to reproduce results from their research. In this study, the linear free vibration properties (natural frequencies and normal modes) of multiple models of the ONERA M6 wing are compared to published results. The wing is modeled as was done in Reference 47 with a 4x4x2 structural mesh (Figure 5.1) and the following material properties:

$$\begin{aligned}\text{Density, } \rho_p &= 2700 \text{ kg/m}^3 \\ \text{Young's Modulus, } E &= 7.102 \times 10^{10} \text{ Pa} \\ \text{Poisson's Ratio, } \nu &= 0.32\end{aligned}$$

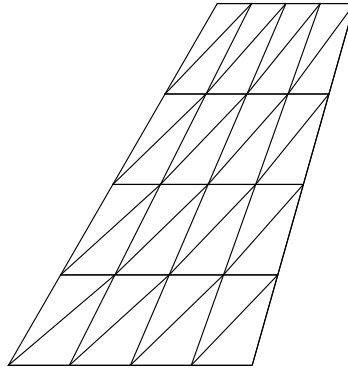


Figure 5.1: Structural finite element mesh for the ONERA M6 wing.

The thickness profiles of the models are shown in Figure 5.2. Wing A has a uniform plate thickness of 0.025 m. Wing B also has a uniform thickness, but it is reduced to 0.018 m. The thickness of wing S is 0.018 m on the inboard 50% span and 0.01575 m on the outboard 50% span. Wing T has a tapered thickness, such that the plate thickness to chord ratio (h/c) is always 0.018. The root chord of the wing is 1.0 m.

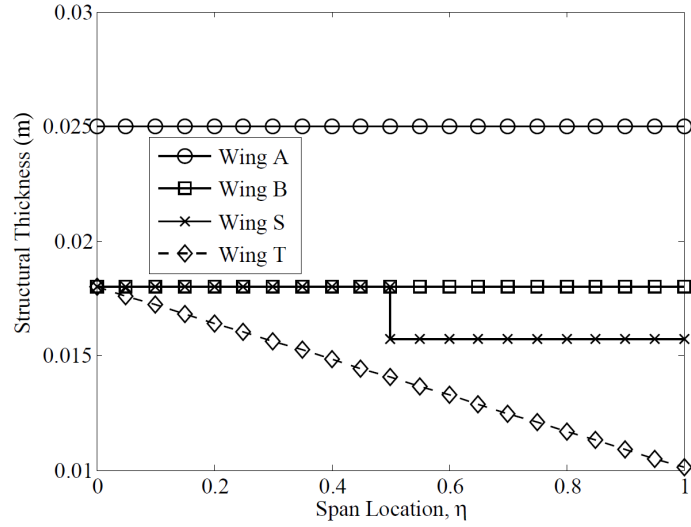


Figure 5.2: Structural thickness profiles.

Table 5.1 contains comparisons of the first four natural frequencies of each model between the present work and Reference 47. For wing A and wing B, the results are nearly identical. These two wings have constant structural thickness along their span. For the wings with varying thickness, wing S and wing T, there are differences in the results. The differences are due to a programming error that was present in the code used in Reference 47. The error was discovered and corrected for the present set of results. The programming error is discussed in Section B.6 of Appendix B.

Table 5.1: Natural frequencies of the ONERA M6 wing model.

Wing A				Wing B		
	Present	Ref. 47	Difference	Present	Ref. 47	Difference
ω_1	64.76	64.78	0.0%	46.64	46.64	0.0%
ω_2	294.72	294.9	0.0%	212.16	212.4	-0.1%
ω_3	388.97	389.2	0.0%	280.20	280.2	0.0%
ω_4	796.24	796.5	0.0%	573.09	573.5	-0.1%
Wing S				Wing T		
ω_1	48.68	52.70	-7.6%	48.98	50.03	-2.1%
ω_2	209.25	192.9	8.5%	190.49	196.5	-3.1%
ω_3	267.37	235.6	13.5%	242.56	246.9	-1.8%
ω_4	531.78	442.9	20.1%	453.34	467.9	-3.1%

Normal mode shapes from wing B are shown in Figure 5.3. Upon visual inspection, the modes match those provided in References 46 and 47.

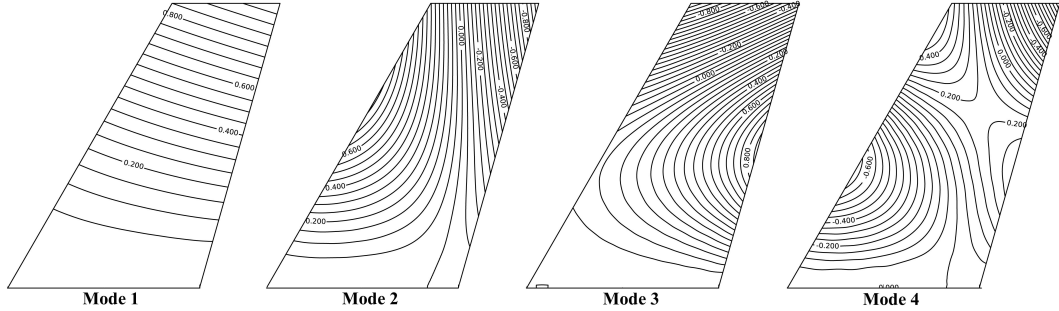


Figure 5.3: First four normal modes of the ONERA M6 wing with structural configuration B.

In general, the structural model comparisons presented here correlate well with published results. Where the correlation is not as good, the differences are well understood and justifiable. The computer code used to generate the structural finite element model can be considered validated and ready for use in aeroelastic studies.

5.3 Steady CFD Results

For the steady CFD analysis, flow around a rigid wing is simulated by time marching the fluid model presented in Chapter 2 using a four-stage Runge-Kutta integrator until the solution converges to steady state. Using this method, steady aerodynamic results were computed for a rigid ONERA M6 wing for the following flow conditions:

$$\text{Angle of Attack, } \alpha = 3.06 \text{ degrees}$$

$$\begin{aligned}
\text{Mach Number, } M_\infty &= 0.84 \\
\text{Air Density, } \rho_\infty &= 1.223 \text{ kg/m}^3 \\
\text{Speed of Sound, } a_\infty &= 338.82 \text{ m/s}
\end{aligned}$$

The ONERA M6 wing has been studied at these conditions both in the wind tunnel [43] and computationally [46, 51, 127] making for a good test case. To be considered valid, the CFD code used in this work must be able to reproduce results from these studies.

A new aerodynamic mesh was generated with the updated meshing program. Portions of the mesh are shown in Figures 5.4 and 5.5 and described in Table 5.2. The present mesh is denser than the one used by Hwang and Seber in References 46 and 51.

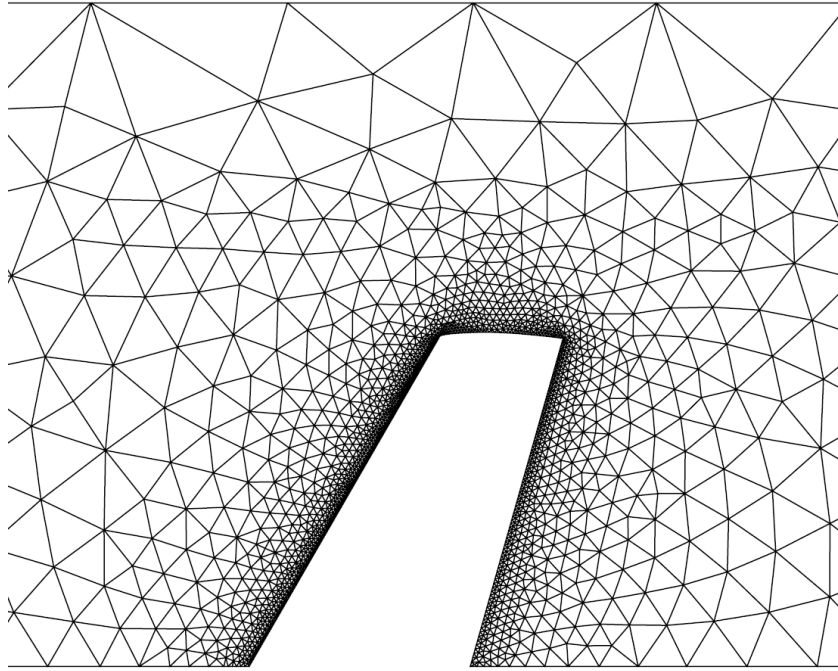


Figure 5.4: ONERA M6 wing model symmetry plane mesh.

Number of grid points = 4730
Number of triangles = 9081

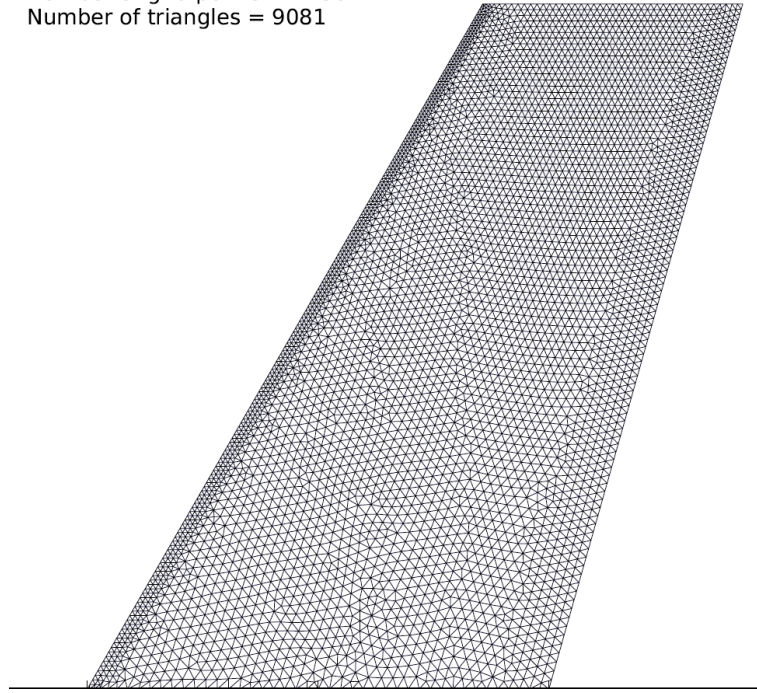


Figure 5.5: ONERA M6 wing surface mesh.

Table 5.2: Mesh details for the ONERA M6 wing model.

	Present	Refs. 46, 51
Nodes	47,777	29,600
Faces	482,536	294,041
Cells	234,944	142,594
Nodes on Wing	10,583	7,468
Faces on Wing	20,792	14,670

Comparisons between results from the steady CFD simulation and a wind tunnel test [43] are presented in Figure 5.6. The steady pressure coefficients show a supersonic region on the forward part of the wing from root to tip. On the inboard portion of the wing, there are two shocks, one at approximately 10% of chord and the second just aft of mid-chord. On the outboard portion of the wing, the shock is stronger and closer to the leading edge.

The lower surface pressure coefficients correlate well with the test data. The upper surface results accurately predict the minimum pressure at the wing leading edge. On the inboard portion of the wing the pressure in the supersonic region closer to the shock is slightly over-predicted. The predicted shock location tends to be fairly accurate, but at some spans is approximately 5% of chord aft of the experimental data. At the trailing edge tip, the upper surface pressure is under-predicted. This could be due to differences in the shape of the wing tip between the wind tunnel and simulation models. The results are similar to those of Hwang [46].

The Mach distribution on the upper wing surface is shown in Figure 5.7. The region of supersonic flow on the wing can easily be identified as any region colored yellow, orange, or red. There is a strong shock on the outboard forward portion of the wing. On the inboard section, there are two weaker shocks, one near the leading edge, and one at about 65% chord. Upon visual inspection, the distribution agrees with Figure 27 of Reference 46.

Total wing lift and drag coefficient comparisons are shown in Table 5.3. The present results are in-line with other published results. Relative to Reference 46, the present simulation gives a C_L that is about 4% smaller and a C_D that

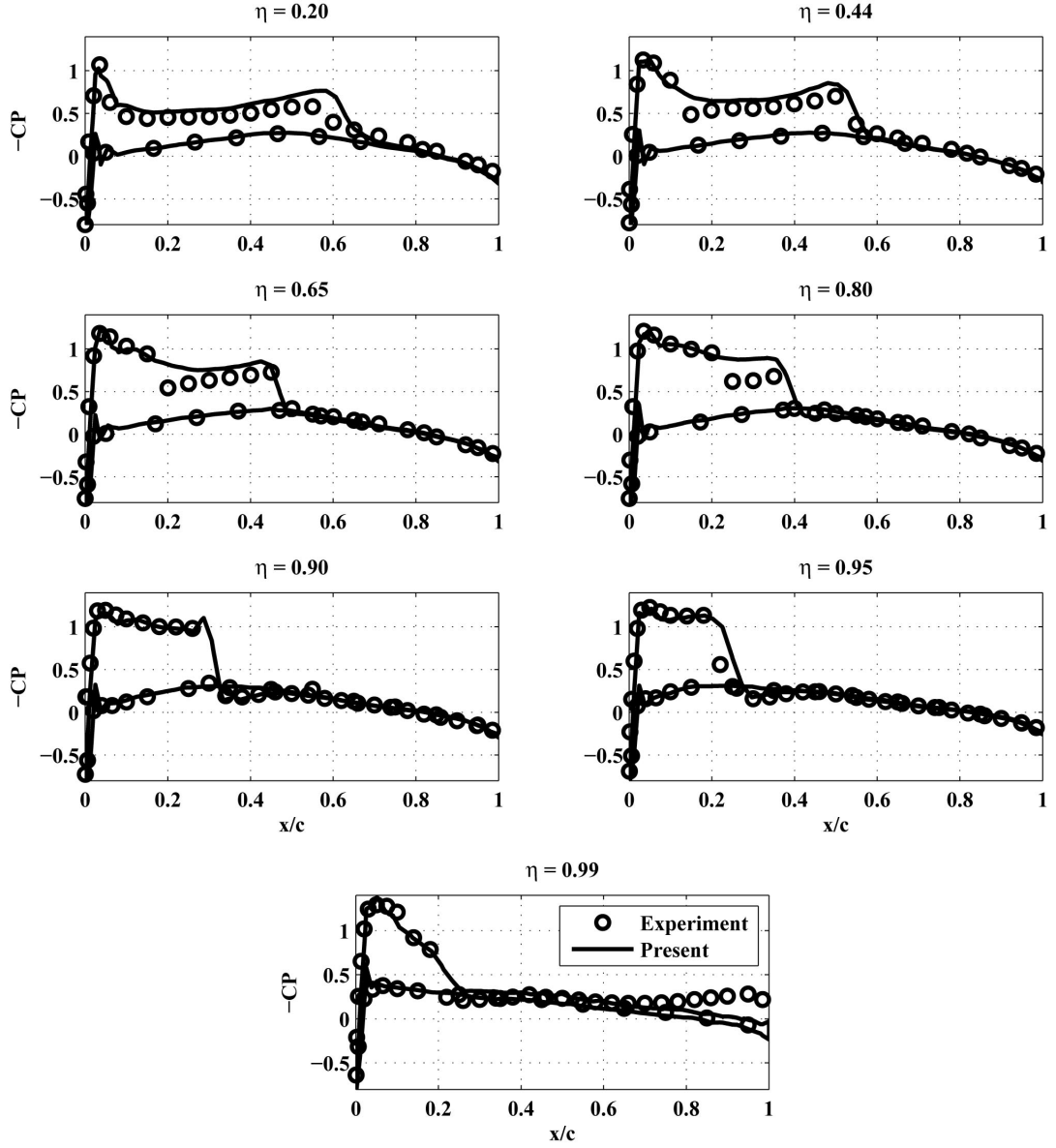


Figure 5.6: ONERA M6 wing steady pressure distribution at Mach 0.84, $\alpha = 3.06$ degrees.

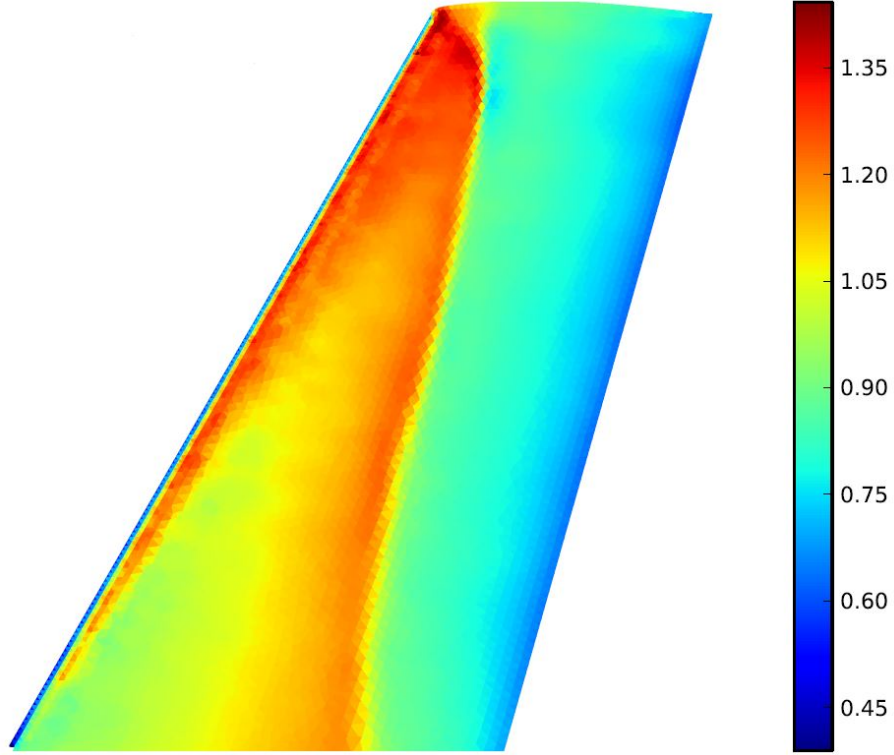


Figure 5.7: ONERA M6 upper wing surface steady Mach distribution.

Table 5.3: C_L and C_D for the rigid ONERA M6 wing.

	C_L	C_D
Present	0.294	0.0216
Hwang [46], Mesh 1	0.303	0.0201
Hwang [46], Mesh 2	0.307	0.0190
Mavriplis [127]	0.292	0.0131

is about 7% larger. The differences are due to the updated aerodynamic mesh and the associated dissipation parameters used in the analysis. Also included are results from Reference 127, where the mesh contained 357,900 nodes and over 2 million tetrahedra.

The steady CFD comparisons presented here correlate with published results. The computer code used to generate the steady fluid response can be considered validated and ready for use in aeroelastic studies.

5.4 Static Aeroelastic Results

In the static aeroelastic analysis, steady air flow around a flexible wing is simulated by coupling the finite element Euler fluid model (Chapter 2) to the nonlinear finite element structural model (Chapter 3) by the Direct Eulerian-Lagrangian method (Chapter 4). The static analysis uses the flow conditions from a converged steady CFD solution as the initial conditions. The simulation is run until the solution converges. A four-stage explicit Runge-Kutta scheme is used for the fluid, and Newton-Raphson iterations are used for the structure. The solution process is described in Section 4.3.1.

Static aeroelastic results for the ONERA M6 wing were published by Seber [51] using a nonlinear structural finite element formulation that is similar to the one used in the present calculations (see Chapter 3). Comparisons are made for three wings modeled with a 6x8x2 structural mesh. This mesh was chosen because it matches the mesh used in Reference 51. The structural model of the wing is shown in Figure 5.8. The wings each have a different structural thickness to root

chord ratio, $h/c_{root} = 0.014, 0.018, 0.025$, representing low, medium, and high stiffness respectively. The structural thickness is constant over the entire wing. The material properties, flow properties, and fluid mesh are the same as those used in Section 5.3.

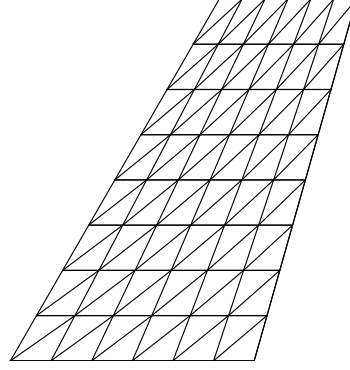


Figure 5.8: 6x8x2 structural finite element mesh for the ONERA M6 wing.

Tip displacement comparisons at the leading and trailing edge for the three structural models are shown in Table 5.4. The high flexibility wing (0.14 structural thickness) converged to nearly the same deflection reported in the previous work, while the two thicker wings deflected slightly less. The results are all less than 6% different, and the differences in deflection are all less than 40% of the wing thickness.

The smaller deflections in the present simulations are due to the smaller lifting force produced with the updated aerodynamic mesh. The aerodynamic mesh for the static analysis is the same mesh used in the steady analysis presented in Section 5.3. The mesh used by Seber is the same as the mesh used by Hwang [46]. Table 5.5 shows that the updated aerodynamic mesh used in the present work produces smaller lift coefficients for both the rigid and flexible wings. This results

Table 5.4: Tip deflection for the flexible ONERA M6 wing.

	Present	Seber [51]	Difference
0.025 Leading Edge	0.0662	0.0692	-4.3%
0.025 Trailing Edge	0.0680	0.0720	-5.6%
0.018 Leading Edge	0.159	0.165	-3.6%
0.018 Trailing Edge	0.165	0.172	-4.1%
0.014 Leading Edge	0.261	0.260	0.4%
0.014 Trailing Edge	0.276	0.275	0.4%

Table 5.5: C_L for the flexible ONERA M6 wing.

h/c_{root}	Present C_L	Seber [51] C_L	Difference
Rigid	0.294	0.304	-3.3%
0.025	0.288	0.297	-3.0%
0.018	0.266	0.273	-2.6%
0.014	0.219	0.222	-1.4%

in smaller deflections for the updated simulation.

Sectional lift coefficient and angle of attack as a function of wing span are shown in Figures 5.9 and 5.10. Both sectional lift and angle of attack are smaller for the flexible wings relative to the rigid wing. This is due to the structural washout effect which reduces the local angle of attack in the streamwise direction for swept wings as they deflect. Upon visual inspection, the results agree with Seber [51] after accounting for the lower lift generated in the present results.

Static aeroelastic deformations of the ONERA M6 wing are shown in Figure

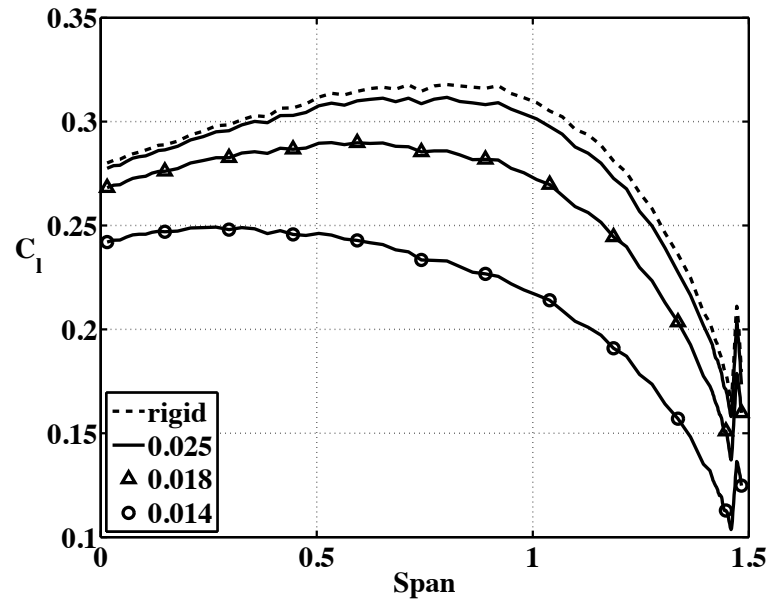


Figure 5.9: Lift coefficient vs. span of the ONERA M6 wing.

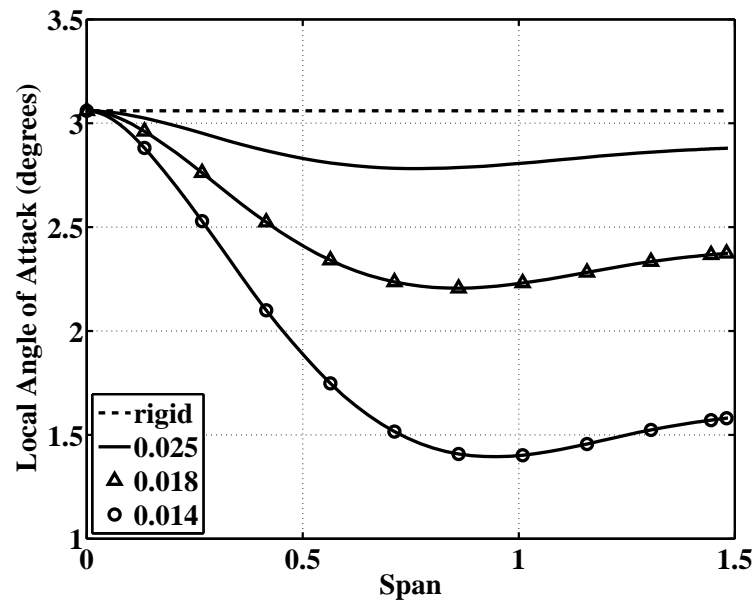


Figure 5.10: Local angle of attack vs. span of the ONERA M6 wing.

5.11. The wing is shown from three orthogonal viewpoints. Both the tip deflection and structural washout effect can be seen in the figure. Mach distribution is plotted on the wing surface and on the tunnel attach wall. The two regions of supersonic flow on the inboard portion of the wing are visible, as is the shape of the supersonic region above the wing (at least at the root). The shock at the aft end of the supersonic region is normal to the wing surface.

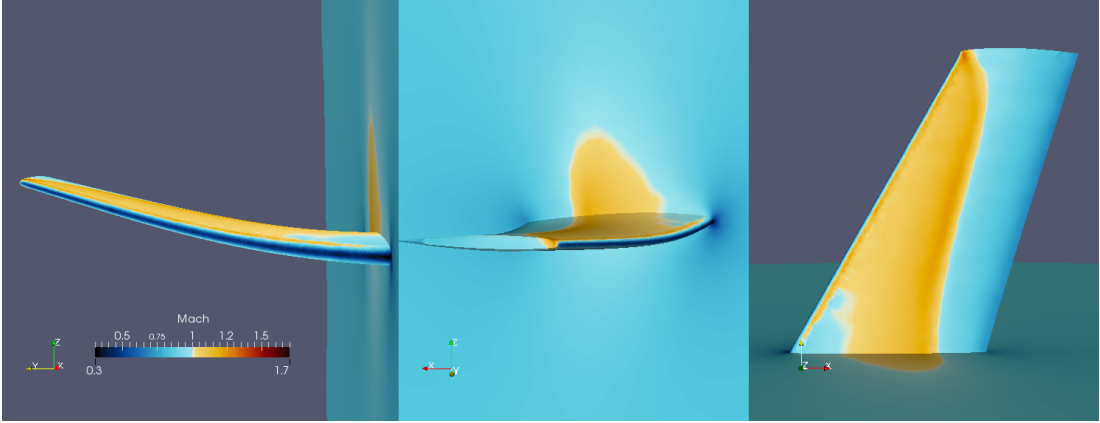


Figure 5.11: Static aeroelastic deformation and Mach distribution of the low stiffness ONERA M6 wing.

The static aeroelastic results presented here correlate with published results. The computer code used to generate the static aeroelastic response can be considered validated and ready for use in dynamic aeroelastic studies.

5.5 Dynamic Aeroelastic Results

In the dynamic aeroelastic (i.e. flutter) analysis, unsteady air flow around a flexible wing is simulated by coupling the finite element Euler fluid model (Chapter 2) to the the nonlinear finite element structural model (Chapter 3) by the Direct

Eulerian-Lagrangian method (Chapter 4). The flutter analysis uses the flow conditions from a converged static aeroelastic simulation as the initial conditions for the flow and the wing deformation. An initial velocity can be imparted to the wing in the shape of any of the normal modes. A five-stage explicit Runge-Kutta scheme is used to solve the coupled fluid-structural equations of motion. The solution process is described in Section 4.3.2.

Flutter simulations were made for three different structural models of the ONERA M6 wing. The models are similar to the ones used in Section 5.4 with constant structural thickness to root chord ratios of $h/c_{root} = 0.016, 0.018, 0.025$ for the three models. The material properties, flow properties, and fluid mesh are unchanged from Sections 5.3 and 5.4. In all three cases, the wing was given a small initial velocity in the shape of the first mode.

Results for the three flutter simulations are shown in Figures 5.12–5.17. For each case, there are two figures: wing tip deflection time histories and wing total energy/aerodynamic work time histories. In all the cases, results are plotted against nondimensional time, which is normalized by the first torsion mode of the wing (the second overall mode) as shown in Equation 5.1.

$$\tau = t \frac{\omega_{1T}}{2\pi} \quad (5.1)$$

The wing tip deflection time histories provide a good sense of the dynamic stability of the fluid-structure system. If the response grows in time, the system is unstable, i.e. the wing is fluttering. If the response decays, the system is stable and the wing is not fluttering. Wing tip time histories are shown in Figures 5.12, 5.14, and 5.16.

The wing total energy and aerodynamic work time histories are useful for verifying the accuracy of the fluid-structure coupling in the simulation. If energy is conserved (in the absence of structural damping), the difference between the energy and the work will be constant and equal to the initial energy in the structure. If an aeroelastic code cannot show that the kinetic and strain energy in the wing is only generated by the work that the fluid does on the structure, the validity of the code should be questioned. The Direct Eulerian-Lagrangian coupling method used in the present work provides a very accurate transfer of energy between the fluid and the structure. This statement is supported by Figures 5.13, 5.15, and 5.17.

Similar flutter results for the ONERA M6 wing were published by Seber [51]. Comparisons are made between these and the present results.

The flutter response of the softest wing ($h/c_{root} = 0.016$) is shown in Figures 5.12–5.13. This simulation was run at Mach 0.84 and an angle of attack of 0 degrees. The growing tip displacement in Figure 5.12 indicates unstable dynamic flutter. The response is predominately bending with some wing twist contributed by the structural washout effect. The trailing edge oscillation magnitude is larger than the leading edge oscillation magnitude causing the streamwise angle of attack to decrease when the wing deflects up and increase when the wing deflects down. The response growth rate for this case is the fastest of the three simulations. Comparisons of the aerodynamic work and wing total energy in Figure 5.13 show that energy is conserved in the simulation, thus we can be confident that Direct Eulerian-Lagrangian coupling has been implemented correctly. There is no direct comparison to the results from Reference 51, as the thinnest wing in those results

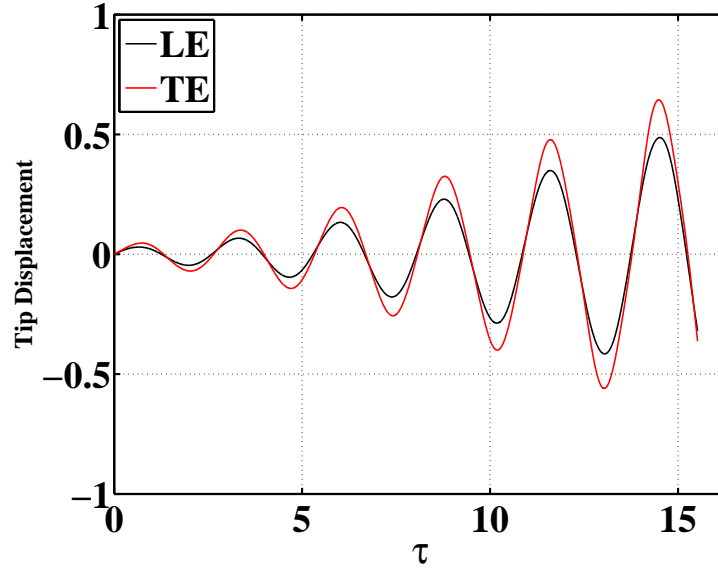


Figure 5.12: Tip deflection time histories of the ONERA M6 wing, $h/c_{root} = 0.016$, Mach 0.84, $\alpha = 0$ degrees.

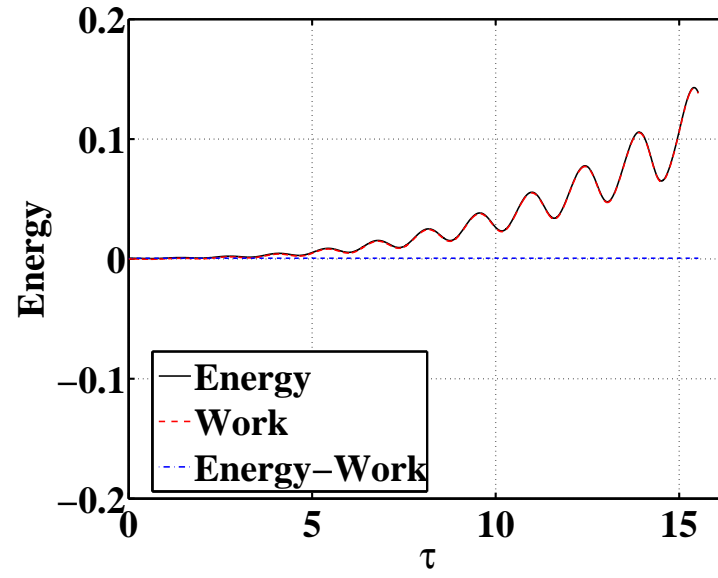


Figure 5.13: Total energy/aerodynamic work time histories of the ONERA M6 wing, $h/c_{root} = 0.016$, Mach 0.84, $\alpha = 0$ degrees.

had a thickness of $h/c_{root} = 0.018$. This wing does follow the trend observed by Seber of thinner wings exhibiting a response rate that grows more quickly [51].

The flutter response of the $h/c_{root} = 0.018$ wing is shown in Figures 5.14–5.15. This simulation was run at Mach 0.84 and an angle of attack of 3.06 degrees. Similar to the $h/c_{root} = 0.016$ wing, this wing also exhibits unstable bending-torsion flutter, though the response growth rate is slower than the thinner wing. Comparisons of the aerodynamic work and wing total energy in Figure 5.15 show

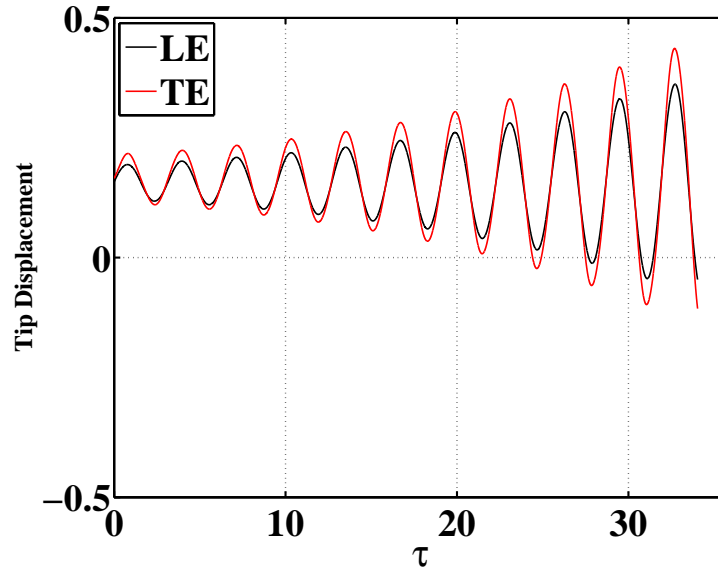


Figure 5.14: Tip deflection time histories of the ONERA M6 wing, $h/c_{root} = 0.018$, Mach 0.84, $\alpha = 3.06$ degrees.

that energy is conserved in the simulation. The difference between the wing total energy and the aerodynamic work is equal to the initial energy in the wing due to static strain energy and kinetic energy from the initial velocity. A similar wing was run at the same flow conditions in Reference 51. The results in Reference 51 exhibited a faster growth rate than the current simulation, but oscillated at the

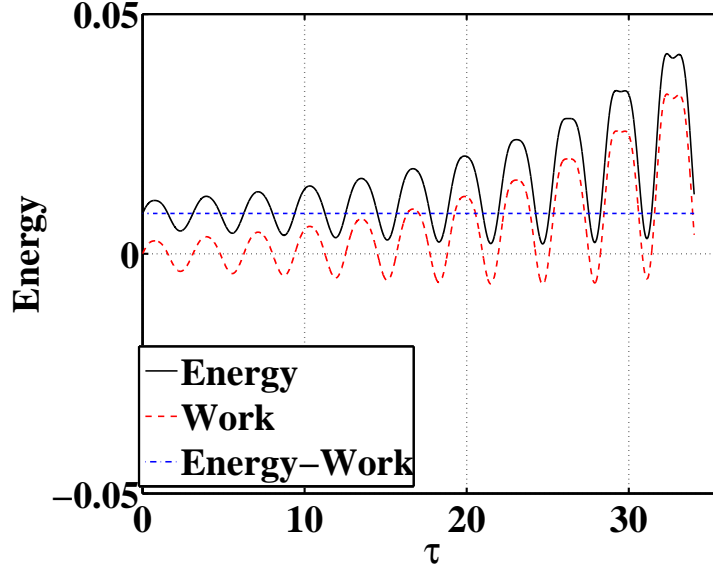


Figure 5.15: Total energy/aerodynamic work time histories of the ONERA M6 wing, $h/c_{root} = 0.018$, Mach 0.84, $\alpha = 3.06$ degrees.

same frequency. The different growth rates are due to differences in aerodynamic forces between the two simulations which were noted for the static aeroelastic results as well (Tables 5.4 and 5.5). A brief study showed that aerodynamic forces and flutter growth rates were dependent on the artificial dissipation parameters which themselves are sensitive to the aerodynamic mesh. By adjusting these parameters, it is likely that the results could be made to match. Despite the differences, qualitatively, the results are similar, and there is no reason to believe that the present results are invalid.

The flutter response of the $h/c_{root} = 0.025$ wing is shown in Figures 5.16 and 5.17. This simulation was run at Mach 0.84 and an angle of attack of 3.06 degrees. Unlike the other simulations, the response of this wing is stable. The response is still bending-torsion, but the frequency is closer to the bending mode

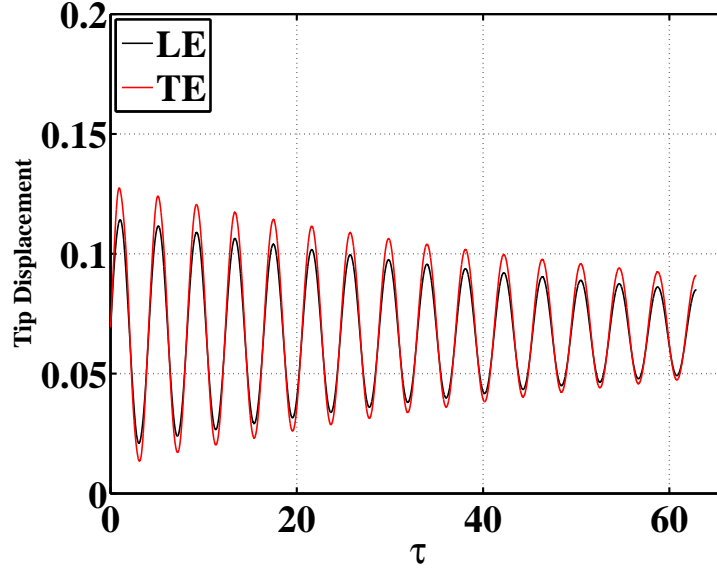


Figure 5.16: Tip deflection time histories of the ONERA M6 wing, $h/c_{root} = 0.025$, Mach 0.84, $\alpha = 3.06$ degrees.

(first structural mode) than it is for the other two simulations. Comparisons of the aerodynamic work and wing total energy in Figure 5.17 show that energy is conserved in the simulation and that the flow is doing net negative work on the wing during each oscillation cycle. A similar wing was run at the same flow conditions in Reference 51. The reference wing had a small growth rate, as opposed to the current simulation where the response is lightly damped. The wings both oscillate at the same frequency. As with the $h/c_{root} = 0.018$ wing, the difference in response is due to differences in aerodynamic forces as a result of the artificial dissipation parameters and the aerodynamic mesh.

In Reference 51, results were provided for an additional wing with $h/c_{root} = 0.027$ at Mach 0.84 and an angle of attack of 3.06 degrees. The response of this wing was neutrally stable. It was observed that as the wing thickness increases,

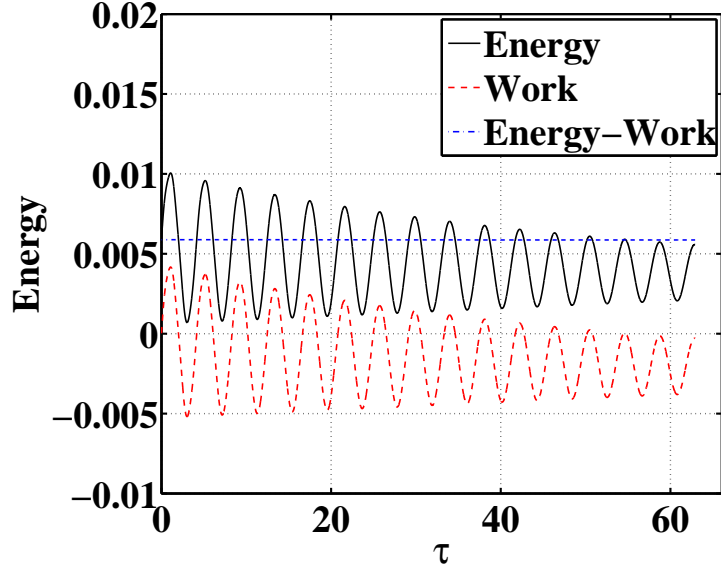


Figure 5.17: Total energy/aerodynamic work time histories of the ONERA M6 wing, $h/c_{root} = 0.025$, Mach 0.84, $\alpha = 3.06$ degrees.

the growth rate of the response decreases, eventually transitioning from growth to decay at some thickness associated with neutral stability. In Reference 51, the thickness for neutral stability is approximately $h/c_{root} = 0.027$. In the present work, the thickness is somewhere between $h/c_{root} = 0.018$ and 0.025, and, based on the growth/decay of the simulated responses, likely closer to 0.025. Differences in the stability behavior are likely due to the previously mentioned differences in aerodynamic forces as a result of artificial dissipation parameter values and the aerodynamic mesh. From these comparisons it is apparent how small variations in the structural and aerodynamic system can result in larger variation in the flutter behavior. This serves as a reminder of the delicate balance governing the dynamic aeroelastic response of wings in transonic flow.

The dynamic aeroelastic results presented here differ slightly from published

results; the reasons for the variation are understood. The computer code used to generate these responses can be considered validated and ready for use in dynamic aeroelastic studies.

CHAPTER 6

Aeroelastic Solutions for Cranked Wings

6.1 Introduction

In this chapter, results from transonic computational aeroelastic simulations of two cranked wings are presented and analyzed. In all simulations, the wing is modeled using nonlinear plate finite elements. Large deformation is accounted for through the use of element-attached local coordinate systems referenced to a single global coordinate system. The formulation of the structural model is discussed in Chapter 3. The fluid is modeled using the mixed Eulerian-Lagrangian formulation of the classical Euler equations and is discretized using a Galerkin finite element approach on an unstructured tetrahedral mesh. Details of the fluid model are included in Chapter 2. The fluid and structural models are coupled by the Direct Eulerian-Lagrangian method where the finite-element shape functions and the local element coordinate systems are used to describe the fluid-structure boundary without approximation. Time synchronization and spatial accuracy are maintained to ensure accurate exchange of energy between the fluid and the structure. For details of the Direct Eulerian-Lagrangian coupling method, see Chapter 4.

In Section 6.2, the response of the NEXST-2 supersonic transport wing is

presented. The computational solutions exhibit multiple types of aeroelastic response including transonic limit cycle flutter at a wide range of dynamic pressures in addition to subsonic and supersonic bending-torsion flutter at higher dynamic pressures and a wide range of Mach numbers. At dynamic pressures at and above the bending-torsion flutter boundary, nonlinear limit cycle flutter was observed. Shock motion dependent on wing deformation was found to play a major role in the response of the wing, and, depending on the flow conditions, was shown to either stabilize or destabilize the response. The computational results correlate closely with observed wind tunnel test responses in Reference 72.

In Section 6.3, the results from a generic high-speed cranked wing are presented. The wing is studied both with and without the inboard sections. Responses show angle of attack and Mach number dependent transonic limit cycle flutter in the second bending mode of the wing. Additionally, and somewhat counterintuitively, the LCO amplitude increased as the wing was made stiffer. The inboard section (when rigid) was observed to delay the onset of LCO to higher angles of attack and Mach numbers due to aerodynamic interaction with the outboard section. When compliance was added to the inboard section, no instability was found. The softening of just a small portion of the inboard section stabilized the response at all conditions tested.

6.2 The NEXST-2 Supersonic Transport Wing

The NEXST-2 supersonic transport wing was analyzed by the Japan Aerospace Exploration Agency (JAXA) in their Transonic Flutter Wind Tunnel to determine

its transonic flutter characteristics [72, 73] (Figure 6.1). Testing revealed a high dynamic pressure flutter region between Mach 0.9 and Mach 1.1. At a narrower range of Mach numbers (0.96–1.06), but lower dynamic pressures, a region of stable limit cycle oscillations (LCO) was found (Figure 6.2). The dynamic responses in the two regions were at unique frequencies, indicating that different modes of the structure were being excited in each case.

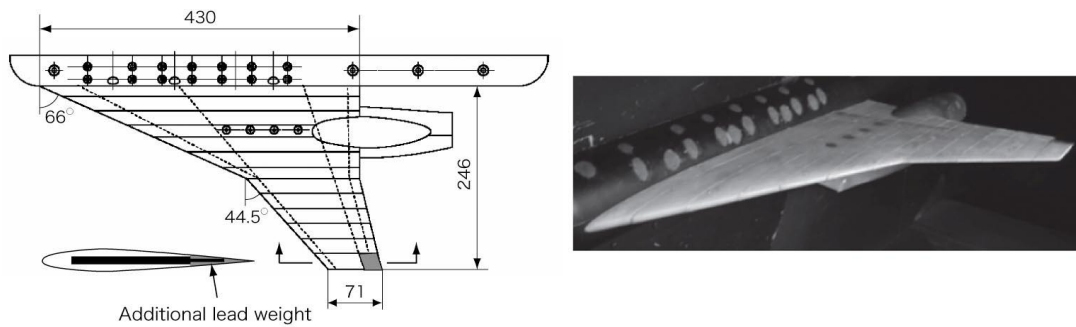


Figure 6.1: Wind tunnel test model of the NEXST-2 SST wing courtesy of References 72, 73.

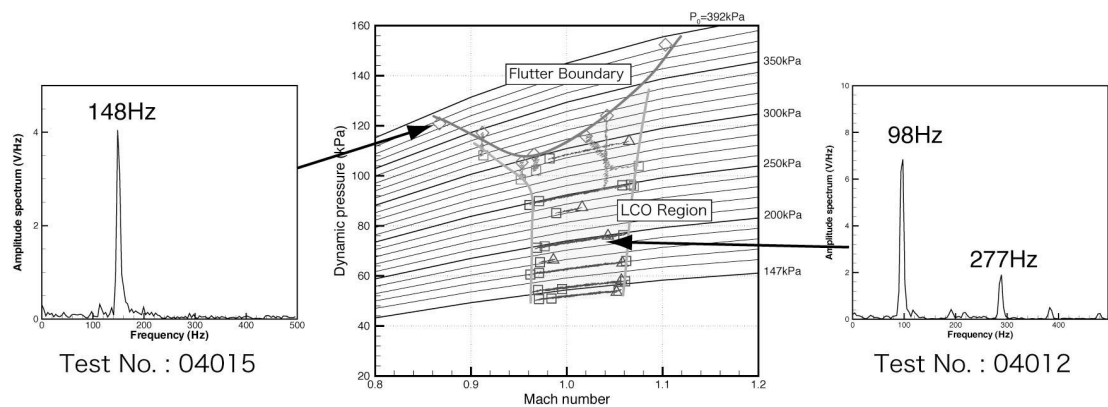


Figure 6.2: Wind tunnel test flutter boundary for the NEXST-2 wing from Reference 72.

6.2.1 NEXST-2 Wing Aeroelastic Model

The structural model used in the simulations is similar to the physical hardware from the wind tunnel tests [72]. The wing stiffness is provided by a thin plate with varying thickness whose planform matches that of the wing. The planform is shown in Figure 6.3. The plate is covered by flexible plastic foam to create the desired airfoil shape. The structural plate thickness and density along with the finite element discretization of the NEXST-2 finite element model are shown in Figure 6.4. The high-density, high-thickness section on the inboard, aft portion of the wing represents the mass and added stiffness provided by the engine which was mounted under and through the wing in the wind tunnel test (Figure 6.1). There is a second high density region at the trailing edge tip that represents a lead weight added at the same location in the tunnel model. The plate thickness values are based on information in Reference 73. The full specifications of the wind tunnel model (stiffness, density, mass of engine, etc.) are not available, so the model is tuned to match the modal frequencies and descriptions in Reference 72.

The first four linear mode shapes of the NEXST-2 structural model are shown in Figure 6.5. A comparison of these modes to published values from Reference 72 is shown in Table 6.1. Excellent agreement between the present model and the wind tunnel hardware has been obtained. Both the frequencies and the mode shapes match the published values. Additionally, it was reported in Reference 72 that the second and third modes featured significant pitching of the attached engine. In the mode shape plots in Figure 6.5, the second and third modes of the present model include large gains along the inboard trailing edge, indicating

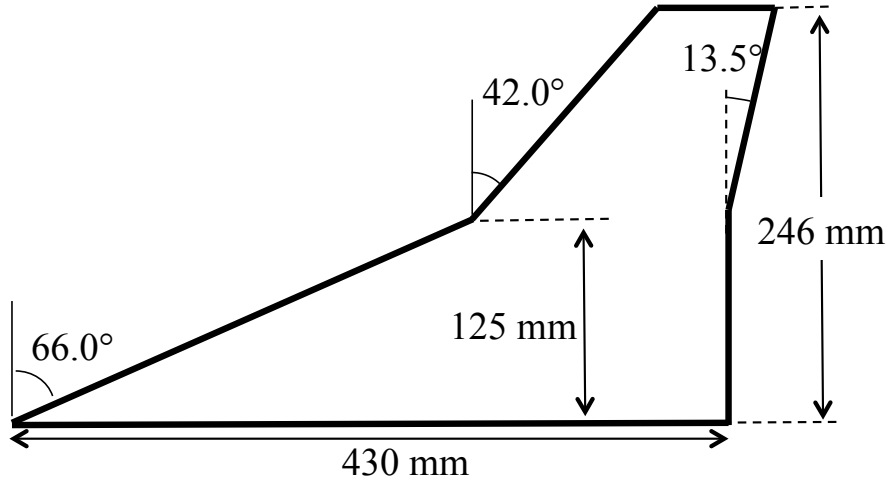


Figure 6.3: Planform of the NEXST-2 wing model.

similar deformation.

For the aerodynamic model of the NEXST-2 flutter simulation, an unstructured, tetrahedral mesh was created with the updated meshing program discussed in Chapter 2. The mesh is pictured from four different viewpoints in Figure 6.6. The model was created to match the published shape from Reference 72 which is based on a symmetric NACA006 airfoil. Details of the mesh are provided in Table 6.2.

6.2.2 NEXST-2 Wing Aeroelastic Results

Flutter simulations for the NEXST-2 wing were carried out at over 350 flow conditions. The Mach number and dynamic pressure were varied to span a wide range of conditions, including those used in the wind tunnel test. All simulations were run at 0 degrees angle of attack, and started from a converged static solution with a small initial velocity in the first mode of the wing.

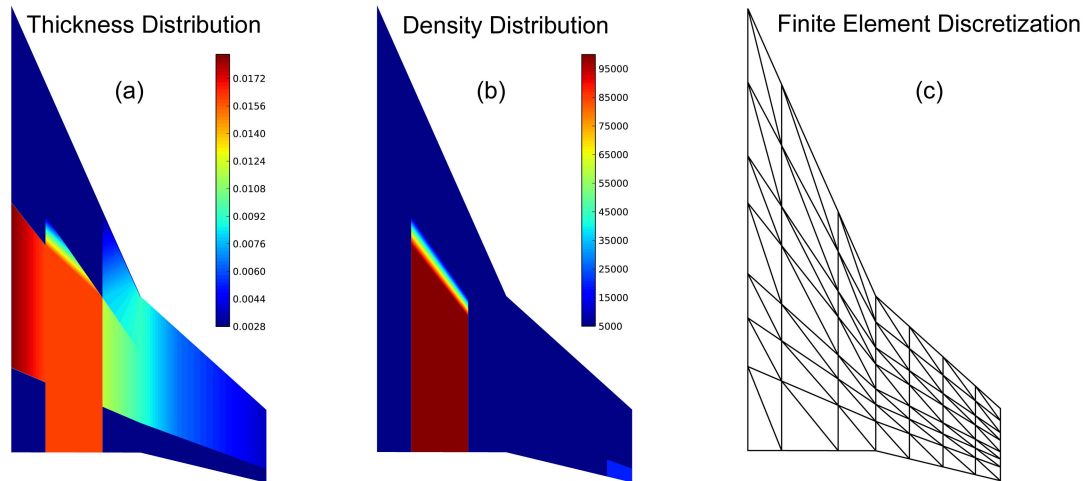


Figure 6.4: The (a) thickness distribution, (b) density distribution, and (c) finite element discretization of the NEXST-2 wing structural model.

Much like the wind tunnel test, the computational solutions revealed multiple types of aeroelastic response at different regions in the Mach versus dynamic pressure plane. Response types are shown in Figure 6.7. A region of stable LCO with the primary response in the first structural mode was found at a narrow range of Mach numbers (0.94–1.04) and a wide range of dynamic pressures. The region is shown in blue and labeled as the *Transonic Dip Region* in Figure 6.7. Responses in this region are very similar to that described in Reference 72 and shown in Figure 6.2. Additionally, subsonic and supersonic bending-torsion flutter regions were found at higher dynamic pressures and a wide range of Mach numbers. Like the wind tunnel test, responses in these regions were at a higher frequency than the transonic dip responses. Shock motion dependent on wing deformation was found to play a major role in defining the region boundaries. Depending on the flow conditions, it was found that the position of the shock could either stabilize or destabilize the response. An additional region labeled the *High Frequency*

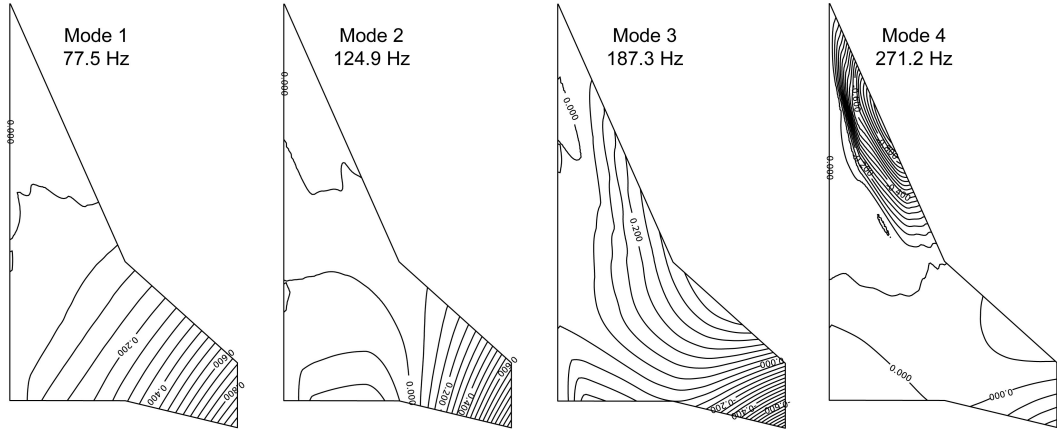


Figure 6.5: First four modes of the NEXST-2 wing structural finite element model.

Table 6.1: Natural frequencies of the NEXST-2 wing structural finite element model.

Description				
Mode Number	from Ref. 72	Ref. 72	FE Model	% Difference
1	1st bending	77 Hz	77.5 Hz	0.6%
2	2nd bending	117 Hz	124.9 Hz	6.8%
3	Outboard torsion	189 Hz	187.3 Hz	-0.9%
4	Outboard bending	270 Hz	271.2 Hz	0.4%

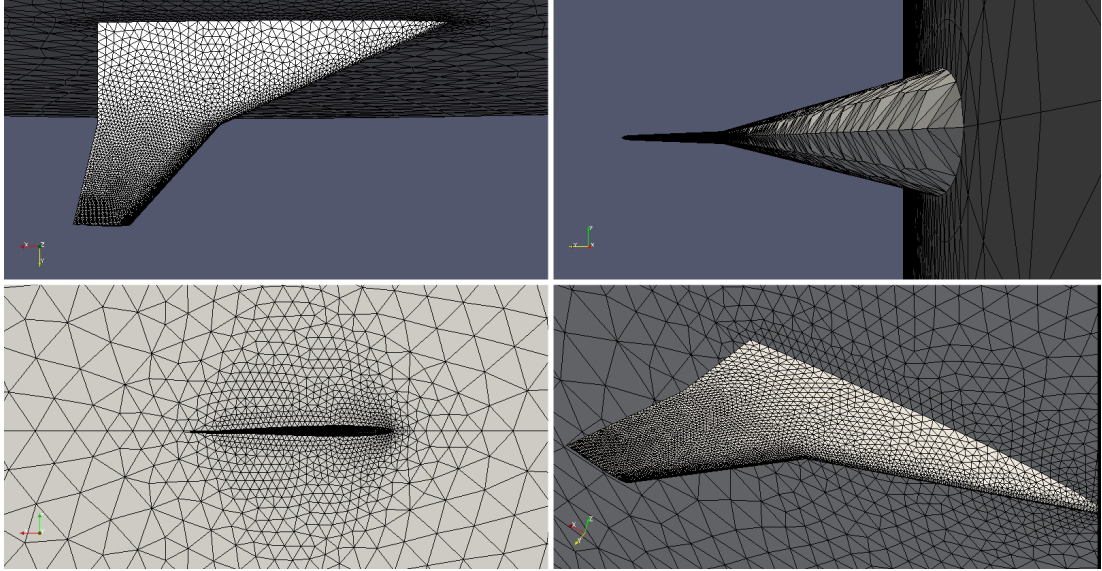


Figure 6.6: The aerodynamic mesh of the NEXST-2 wing model.

Table 6.2: Mesh details for the NEXST-2 wing model.

Nodes	37,791
Faces	384,489
Cells	187,566
Nodes on Wing	7,549

LCO Region was found at subsonic conditions (Mach 0.7-0.75) below the bending-torsion flutter boundary. In this region there was a small persistent response at a frequency above the fourth mode of the wing.

Each small dot in Figure 6.7 denotes conditions at which a simulation was run. The larger, lettered dots represent a condition that was studied in greater detail. These responses are discussed in Sections 6.2.2.1–6.2.2.5. Though the plot shows a definitive line around each region, the responses, especially near the edges of the regions, often gradually morph from damped response to flutter or from one

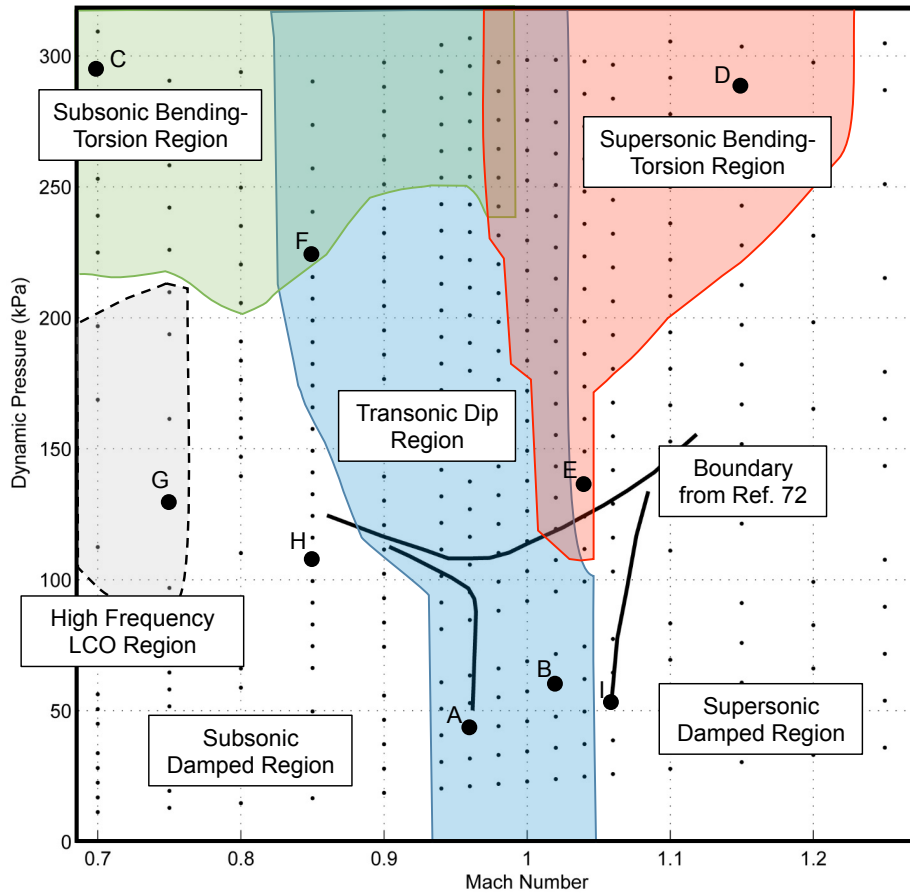


Figure 6.7: The flutter boundary from nonlinear simulations of the NEXST-2 wing.

type of flutter response to another. For many flow conditions, characteristics of multiple response types could be observed. In some cases, one response would dominate over other weaker responses, causing them to decay. In other cases, multiple responses existed together for many periods of oscillation.

The regions in Figure 6.7 are similar to the flutter boundary defined in Reference 72. The transonic dip region is at a Mach range approximately 0.02 lower than observed in the wind tunnel. It is believed that this shift is due to shock-boundary layer interaction which is not modeled by the inviscid Euler equations.

As will be shown later, the termination of the transonic dip occurs when the Mach number is increased to the point where the shock attaches itself to the wing trailing edge and stops moving as freely on the wing surface. Shock-boundary layer interactions have a tendency to slow the rearward movement of shocks with increasing Mach number, resulting in their reaching the trailing edge and terminating the transonic dip at higher Mach numbers [55].

Inaccurate model stiffness is believed to be the reason for differences in the bending-torsion flutter boundary. Though the present model is dynamically similar to the model used in the wind tunnel, export control laws prevented sharing of detailed information, including mass and stiffness. Only the publicly available data in References 72 and 73 could be used to assign the structural finite element properties. It is likely that the computational model is overly stiff, and a reduction in the stiffness would shift the bending-torsion flutter boundary to lower dynamic pressures. This seems especially likely based on the observations in Sections 6.2.2.1–6.2.2.5, where it is shown that deflection magnitude has a direct impact on the nature of the response.

6.2.2.1 Responses in the Transonic Dip

In the present simulations, a region of stable LCO with the primary response in the first structural mode was found at a narrow range of Mach numbers (0.94–1.04) and a wide range of dynamic pressures. The response is very similar to that described in Reference 72 and shown in Figure 6.2. The region is shaded blue and labeled as the *Transonic Dip Region* in Figure 6.7. In this section, the physical phenomena governing responses in the transonic dip are discussed.

Detailed investigation is included for two responses in the transonic dip (points *A* and *B* in Figure 6.7).

Researchers have long believed that the phenomenon known as the “transonic dip” is due to mixed subsonic-supersonic flow and moving shocks on the wing surface [2]. Recently, it was shown with great detail how the motion of shocks contributes to the energy exchange between the fluid and the structure [55]. Bendiksen observed that, for a high aspect ratio swept wing, the mean location of the shock on the wing has a direct effect on the nature of the response and determines the shape of the transonic dip region. Two specific conclusions were drawn:

1. The mean position of the shock must reach 40% to 50% of the wing chord before shock motion becomes destabilizing and the transonic dip is entered.
2. The transonic dip region abruptly ends once the shocks reach the trailing edge and stabilize the response.

With this in mind, it is useful to review two fundamental concepts of transonic flow over moving wings: shock motion and the structural washout effect.

At subsonic Mach numbers, as a wing approaches the critical Mach number, M_{cr} , regions of supersonic flow begin to form on the wing surface. The flow decelerates as it moves aft over the wing forming a shock wave nearly normal to the wing surface at the aft end of the supersonic region (Figure 6.8). The position of the shock wave is governed by the Mach number of the flow, the airfoil shape, and the local angle of attack. For a given airfoil, and in the absence of flow separation, the shock is further forward at lower Mach numbers and lower angles of attack (including negative angles) and further aft at higher Mach numbers and

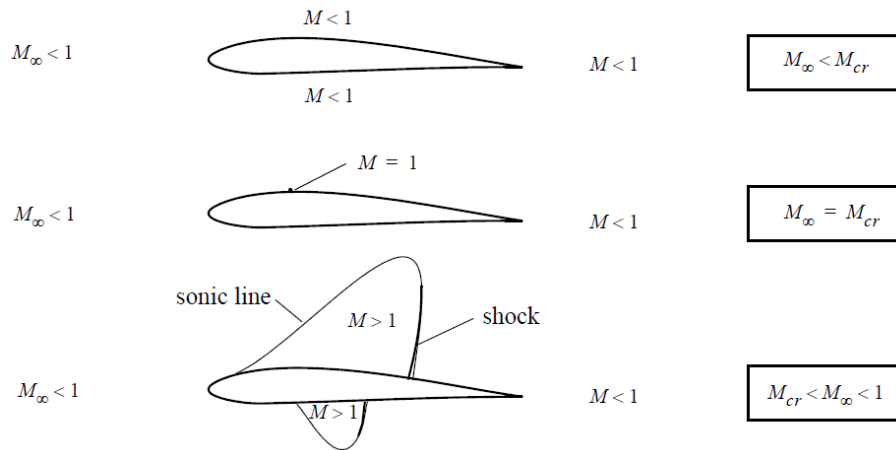


Figure 6.8: Transonic flow over an airfoil.

higher angles of attack.

Any wing deformation that alters the local angle of attack can change the airflow over the wing and cause the shock to move forward or aft. This type of deformation is often due to the structural washout effect (Figure 6.9) which couples bending and twist in swept wings (or wings with a swept elastic axis). When an aft swept wing bends upward, the angle of attack of individual stream-wise sections of the wing decreases. Conversely, when the wing bends down, the

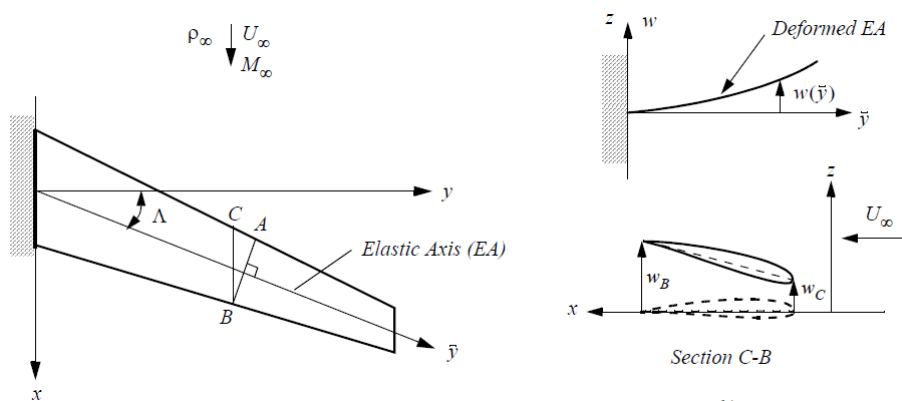


Figure 6.9: The structural washout effect for swept wings.

streamwise angle of attack is increased. The structural washout effect can interact with the mixed subsonic/supersonic flow in the transonic region resulting in the creation, destruction, or motion of shocks on the wing surface as the wing bends up and down and the local angles of attack change.

With the observations of Bendiksen [55] in mind, the distribution of supersonic flow on the wing surface for different Mach numbers was studied. Figure 6.10 shows the Mach distribution on the upper surface of the wing during steady flow at 0 degrees angle of attack for multiple Mach numbers. Mach 0.9 is the lowest Mach number for which a region of supersonic flow appears. At this condition a weak shock is located at approximately 20% of the chord, and the transonic dip only exists at dynamic pressures above 100 kPa. With a small increase of 0.04 in the steady Mach number, the supersonic flow on the wing expands inboard and aft. Additionally, the strength of the shock (indicated by the dark spot near the wing tip) increases. At this condition, the wing is in the transonic dip and limit cycle flutter in the first mode of the wing occurs even at very low dynamic pressure. As the Mach number is increased further, the supersonic region continues to expand and move further aft. The largest Mach number where the transonic dip region exists is Mach 1.04. At this Mach number, the shock is at 99% of the chord near the wing tip. With a small increase in Mach number to 1.06, the shock wave reaches the trailing edge, and the transonic dip region is terminated.

One may wonder why the transonic dip region moves to lower Mach numbers at higher dynamic pressures. This phenomenon can be explained by the larger wing deformation that occurs at higher dynamic pressures. At Mach numbers immediately below the transonic dip (Mach 0.85-0.9), when wing deformation is

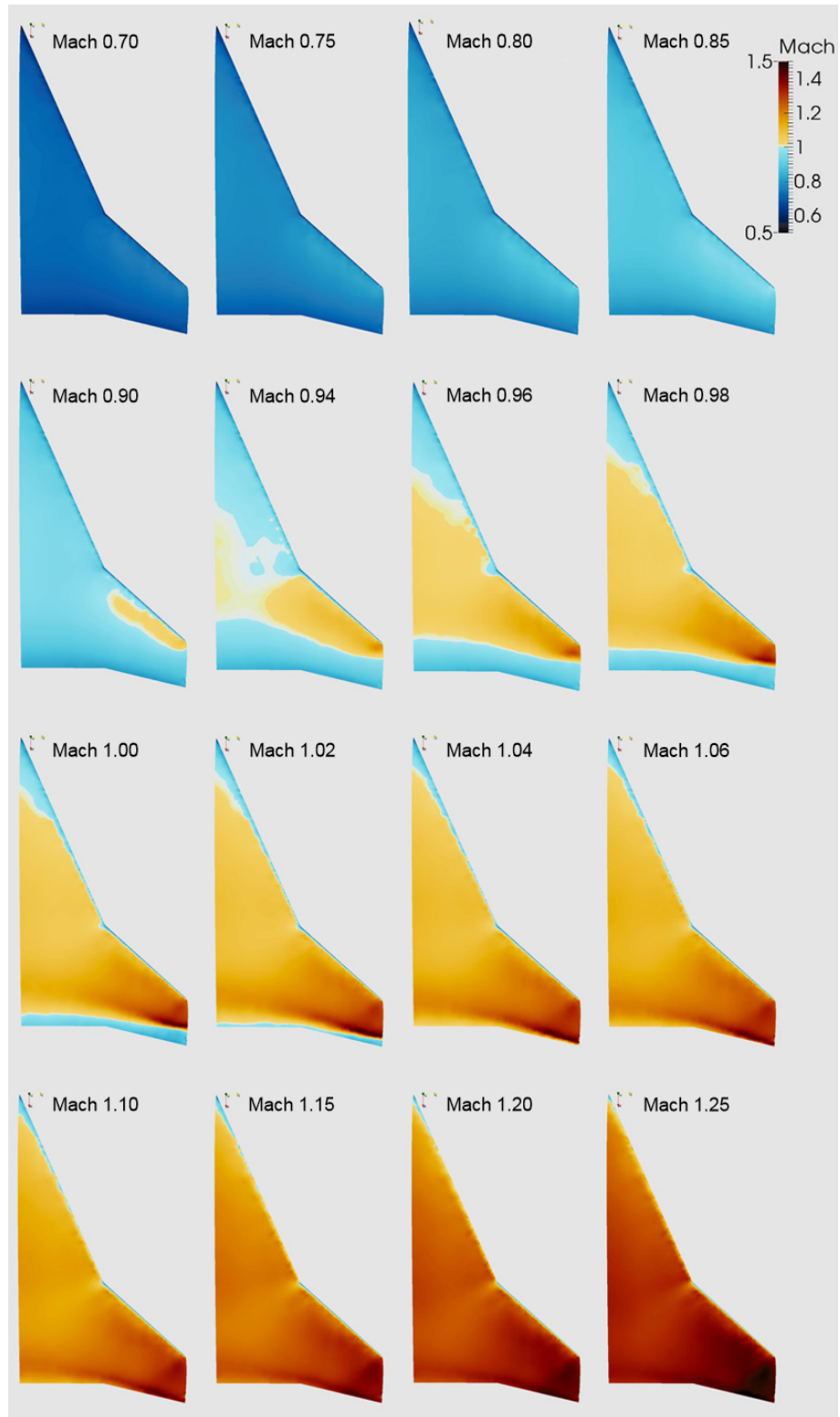


Figure 6.10: Mach distribution at a range of steady conditions for the NEXST-2 wing.

small to moderate, shocks are either not present or are located on the most forward part of the wing chord, and the initial velocity imparted to the wing damps out. As the dynamic pressure is increased, the flow causes larger deformation and larger angles of attack due to the structural washout effect. At some dynamic pressure, the deformation becomes large enough to move the shock aft for enough of the oscillation cycle to bring the wing into the transonic dip.

In the same manner, at Mach numbers near the right edge of the transonic dip, larger wing deformation results in increased local angles of attack and pushes the shock aft until it reaches the trailing edge. Shocks that reach the trailing edge along the full span of the wing have a tendency to stick despite changes in angle of attack [56]. This results in the shock spending larger portions of the cycle attached to the trailing edge. When this occurs, the wing is stabilized against the first mode response present in the transonic dip, but could be susceptible to supersonic bending-torsion flutter, which can rapidly diverge.

To better demonstrate these concepts, two sets of responses from the transonic dip region are presented. Flutter simulation responses for the NEXST-2 wing at Mach 0.96 with a dynamic pressure of 86.4 kPa are shown in Figures 6.11–6.18. This flow condition is labeled as point *A* in Figure 6.7. Wing tip time histories in Figure 6.11 show the slow growing response moving toward limit cycle. Responses of this type are typical in the transonic dip region, though the amplitude and growth rate of the LCO vary depending on the flow conditions.

The shock response spectrum (SRS) of the trailing edge tip motion is shown in Figure 6.12. The spectrum is nearly identical to the one measured in the wind tunnel (Figure 6.2) [72]. The largest response is in a frequency between the first

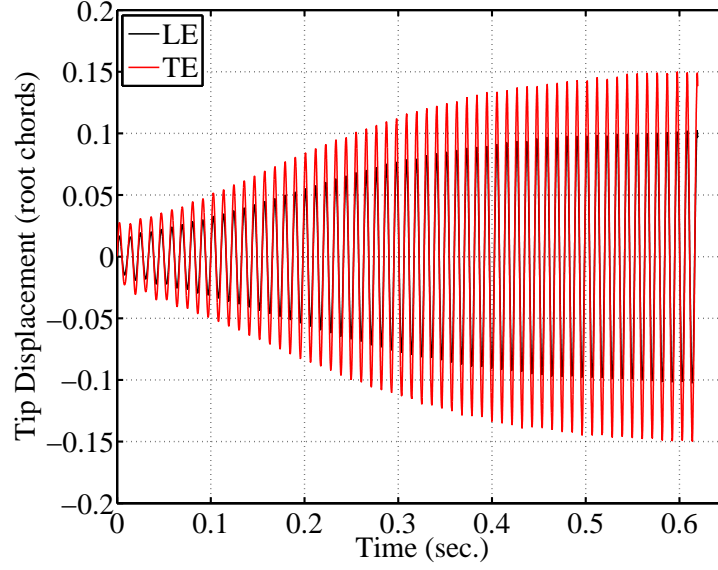


Figure 6.11: NEXST-2 wing tip time histories in the transonic dip region at Mach 0.96, $q = 86.4$ kPa.

and second modes. There is a secondary response at a frequency just above the fourth mode.

Though the SRS provides valuable information regarding the frequency of the response, it can not provide precise information as to which modes are participating. Because the models used in the simulation are in physical coordinates, a transformation to modal coordinates must be made to determine the modal composition of the response. This transformation is accomplished using Equation 6.1, where $\{q\}$ are the system modal responses (either displacements, velocities, or accelerations), $[\phi]$ are the mass orthonormalized system modes, $[M]$ is the system physical mass matrix, and $\{x\}$ are the system physical responses (either displacements, velocities, or accelerations). The most helpful metric for determining the modal composition of a response is the modal kinetic energy. This can be calcu-

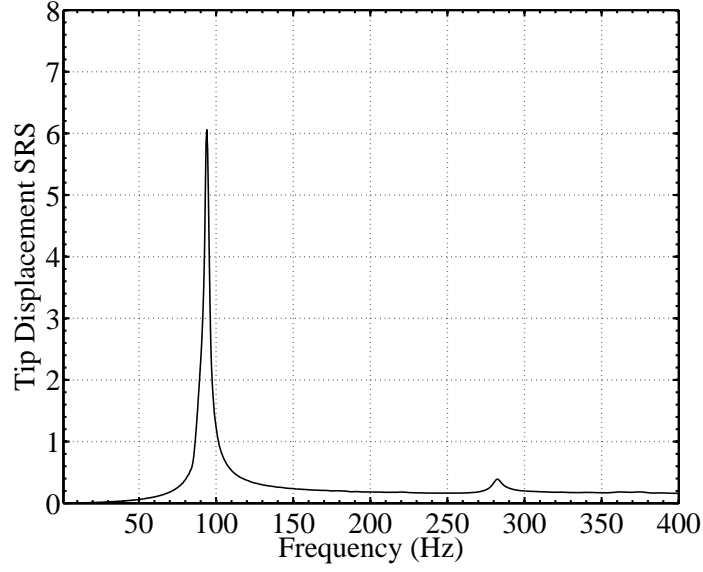


Figure 6.12: SRS of the NEXST-2 wing tip trailing edge at Mach 0.96, $q = 86.4$ kPa.

lated for each mode, i , with Equation 6.2 and is plotted for the first five modes of the wing in Figure 6.13. The plot shows that the response is almost exclusively in the first system mode, which is a bending mode. This type of response was identified in Reference 126 as a natural mode instability or *essentially single degree-of-freedom* (ESDOF) flutter. As the amplitude increases, the third mode begins to respond at a very small magnitude. This is the mode causing the high frequency peak in the SRS (Figure 6.12).

$$\{q\} = [\phi] [M] \{x\} \quad (6.1)$$

$$KE_i = \frac{1}{2} \dot{q}_i^2 \quad (6.2)$$

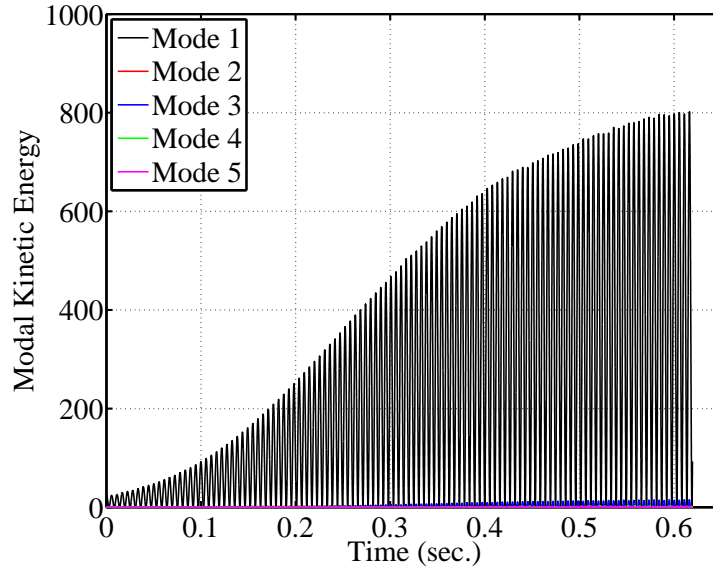


Figure 6.13: Modal kinetic energy of the NEXST-2 wing at Mach 0.96, $q = 86.4$ kPa.

In Figure 6.14, the aerodynamic work is compared to the total energy in the structure (kinetic plus strain). Throughout the simulation, the difference between the work and the energy is constant and equal to the initial energy of the system. Because the Direct Eulerian-Lagrangian coupling methodology accurately models the energy exchange between the fluid and the structure, the conservation laws are satisfied and energy is neither created nor destroyed.

Plots of the Mach, pressure coefficient, and aerodynamic power distributions on the wing surface are shown in Figures 6.15–6.17. Each figure shows ten time points during a full period of oscillation occurring after the wing has reached limit cycle amplitude. The deformed wing position at each of these time points is easily observed in Figure 6.18, which shows the wing from a location beyond the tip looking toward the root. In each of the figures, time (a) occurs when the tip

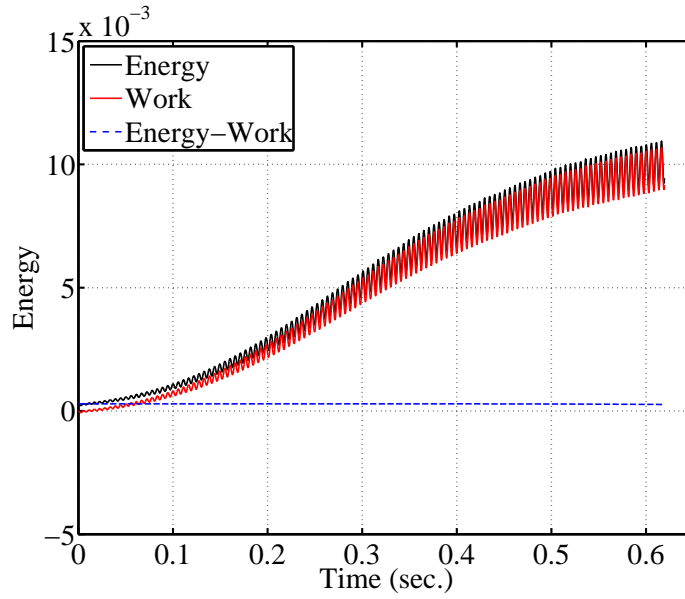


Figure 6.14: Aerodynamic work and wing total energy of the NEXST-2 wing at Mach 0.96, $q = 86.4$ kPa.

is at its maximum positive deformation and has just started moving downward. The images show how the changing deformation of the wing affects the flow field, shock motion, and energy transfer from the fluid to the structure. Because the wing is in a stable limit cycle, the net energy transfer between the fluid and the structure during one cycle is close to zero. The images reveal how the energy transfer is balanced over different areas of the wing during the cycle. Energy transfer, or aerodynamic work, is equal to the aerodynamic power integrated over time. The aerodynamic power distribution on the wing is shown in Figure 6.17. The convention used here is that positive power indicates the energy transfer is from the fluid to the wing. In most of the images only the top surface of the wing is visible, but because the wing is symmetric and is at an angle of attack of zero degrees, the response is also symmetric. That is, the flow field on

the upper surface during downward motion is identical to the flow field on the lower surface during upward motion. The images of the wing are generated at time points throughout the simulation and can be viewed as a 3-D interactive animation. Videos created from these animations are included as supplementary materials with this dissertation. Analysis of the animations has proven invaluable in understanding the responses discussed here.

At time (a) in Figures 6.15–6.18, the wing is at the top of the cycle just after it has started to move downward, and the structural washout effect results in a large negative angle of attack at the wing tip (Figure 6.18). Because of this, the only shock on the upper wing surface is weak and is located at the most inboard region, where the deflection is small (Figure 6.15). There is, however, a strong shock on the lower surface that can be seen in Figure 6.18. The corresponding pressure distribution (Figure 6.16) shows a downward force on the upper surface over the flexible part of the wing. An accompanying strong negative pressure pulling downward on the lower surface cannot be seen, but exists. The power plot for time (a) shows almost no energy transfer occurring (Figure 6.17). This is due to the slow velocity of the outboard wing while it is near the peak of its deformation.

As the wing tip accelerates downward in time points (b)–(e), the structural washout effect causes the angle of attack to increase, cross through zero, and become positive (Figure 6.18). As this happens, a strong shock begins to form on the upper surface of the wing (Figure 6.15), and the upper surface pressure becomes negative (Figure 6.16). The resulting energy transfer is negative (energy being extracted from the wing) over much of the wing (Figure 6.17). This is due

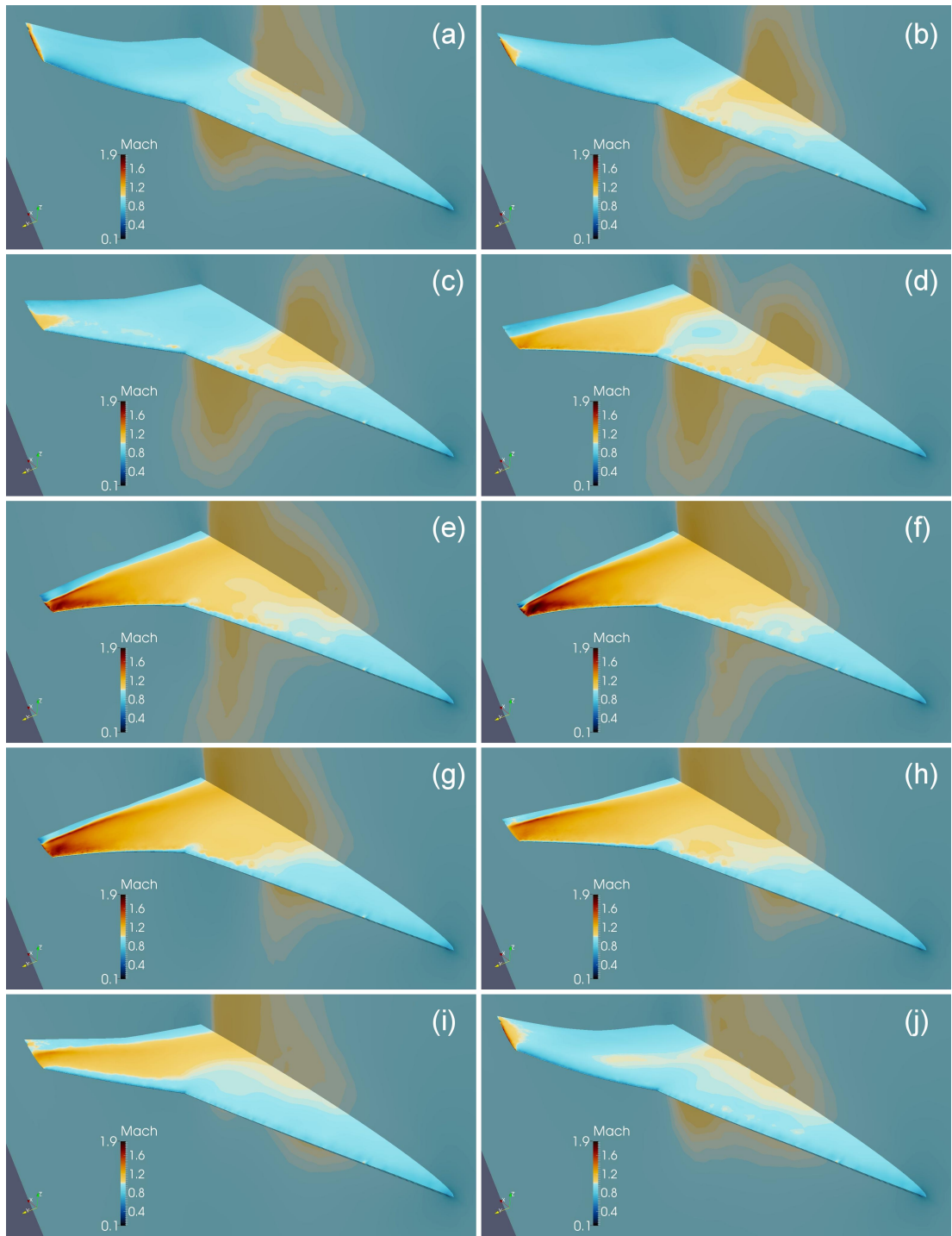


Figure 6.15: Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.96, $q = 86.4$ kPa.

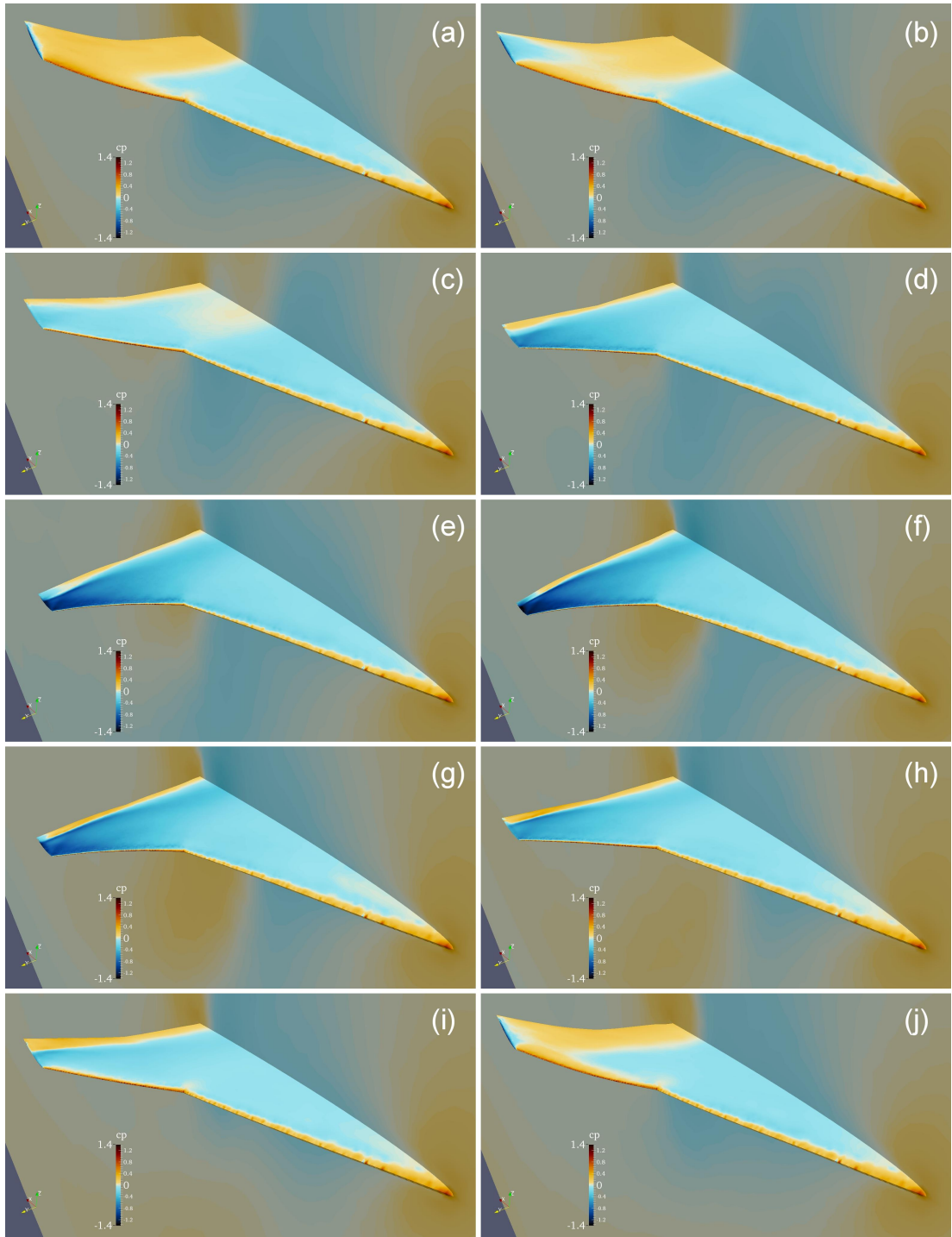


Figure 6.16: Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.96, $q = 86.4$ kPa.

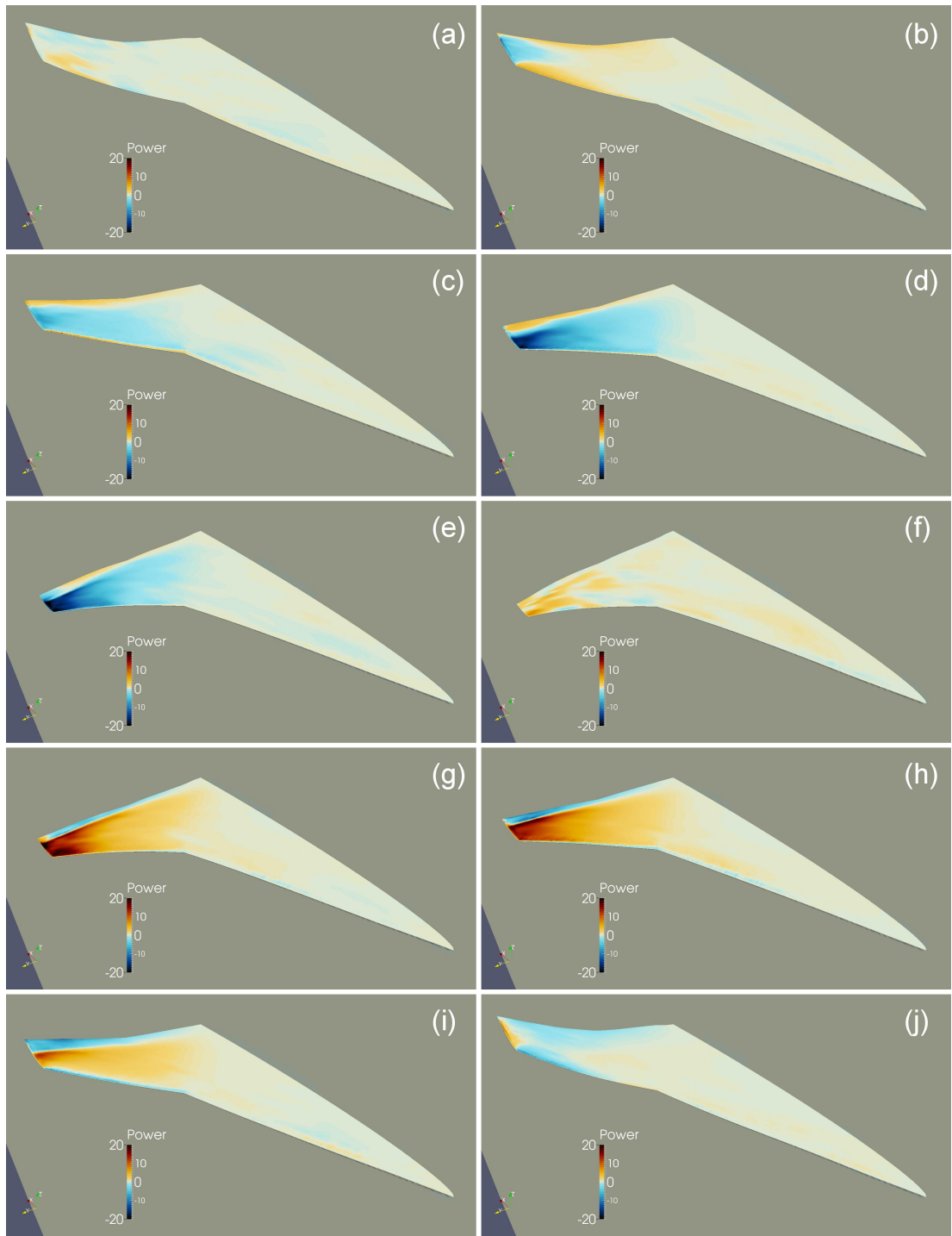


Figure 6.17: Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.96, $q = 86.4$ kPa.

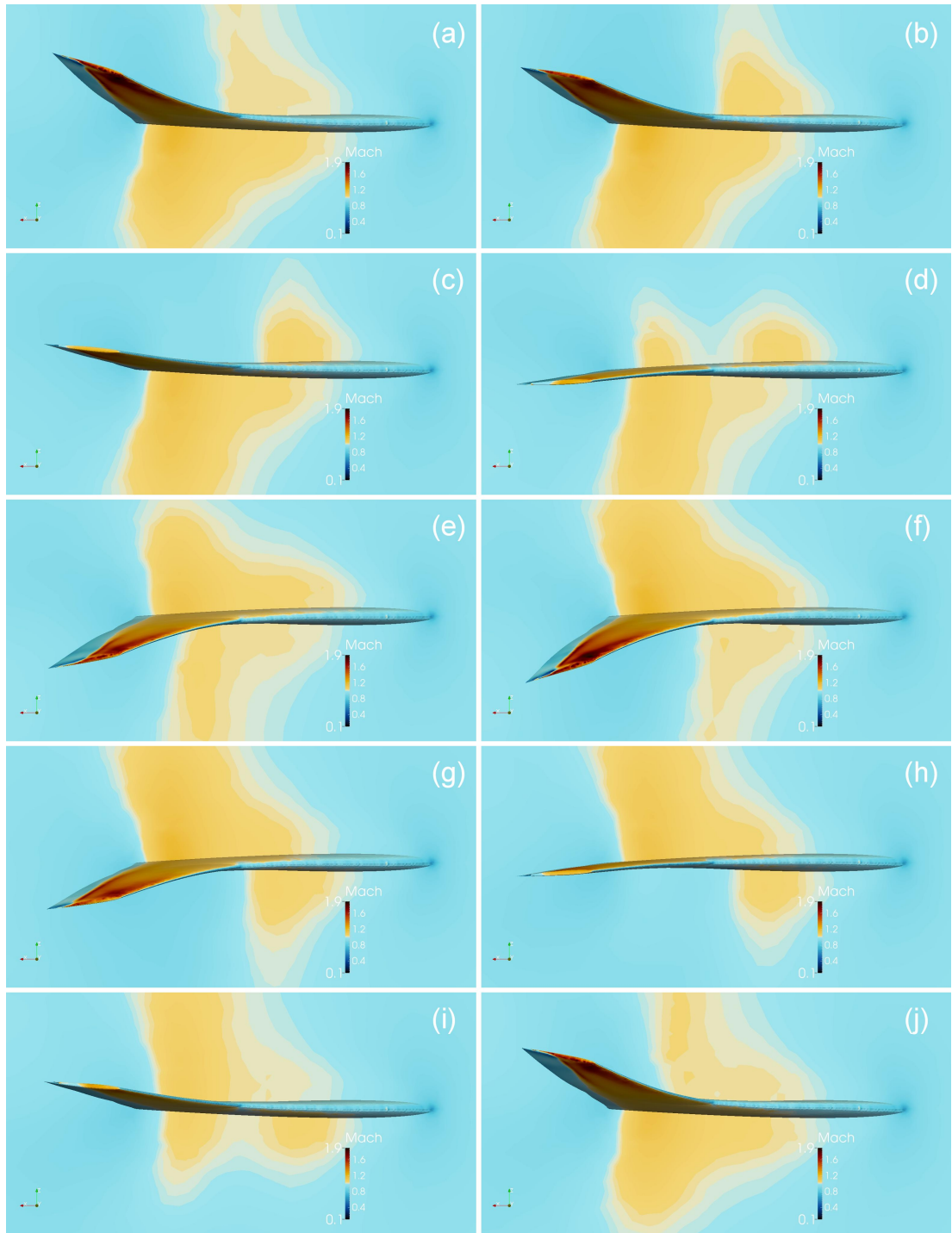


Figure 6.18: Wing deformation as viewed from beyond the NEXST-2 wing tip looking toward the root during a period of limit cycle flutter at Mach 0.96, $q = 86.4$ kPa.

to the strong suction pulling the wing up as it moves down.

At time (f), the wing tip has reached the bottom of the cycle. The upper surface now looks like the lower surface did at time (a). A strong negative pressure region associated with the supersonic flow begins pulling the tip up. As the wing moves upward through time points (f)–(j), the energy transfer is positive to the wing, the angle of attack decreases, and the shock weakens. The wing continues to extract energy from the flow until the upper surface shock disappears over most of the outboard wing and a positive pressure region forms at time (j). At this point, the wing tip has slowed down and is about to begin descending again. Because of the motion, creation, and destruction of shocks and the strong pressure gradients that occur, the energy transfer between the fluid and the structure is balanced. The resulting limit cycle can continue indefinitely.

The Mach 0.96 flow condition is at a low Mach number in the transonic dip region (Figure 6.7, point *A*). An improved understanding can be acquired by studying a second condition at a high Mach number in the transonic dip. Responses for the NEXST-2 wing at Mach 1.02 with a dynamic pressure of 59.7 kPa are shown in Figures 6.19–6.25. This flow condition is represented by point *B* in Figure 6.7.

Wing tip time histories (Figure 6.19), SRS (Figure 6.20), and modal kinetic energy plots (Figure 6.21) show ESDOF flutter in the first mode with magnitude growing slowly to LCO. The Mach 1.02 response is similar to the Mach 0.96 response.

By examining the Mach, pressure, and power distributions in Figures 6.22–

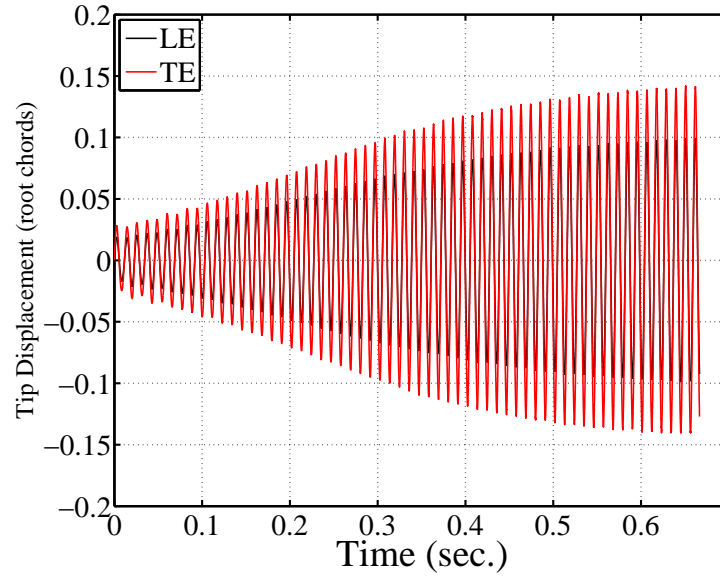


Figure 6.19: NEXST-2 wing tip time histories in the transonic dip region at Mach 1.02, $q = 59.7$ kPa.

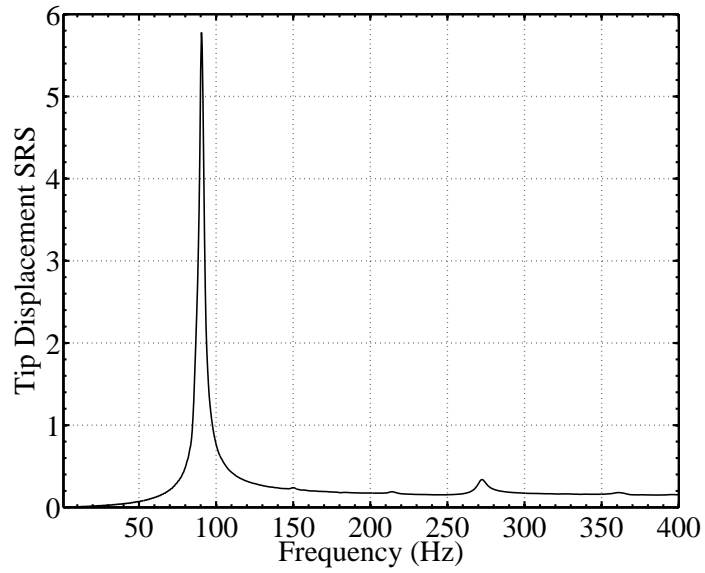


Figure 6.20: SRS of the NEXST-2 wing tip trailing edge at Mach 1.02, $q = 59.7$ kPa.

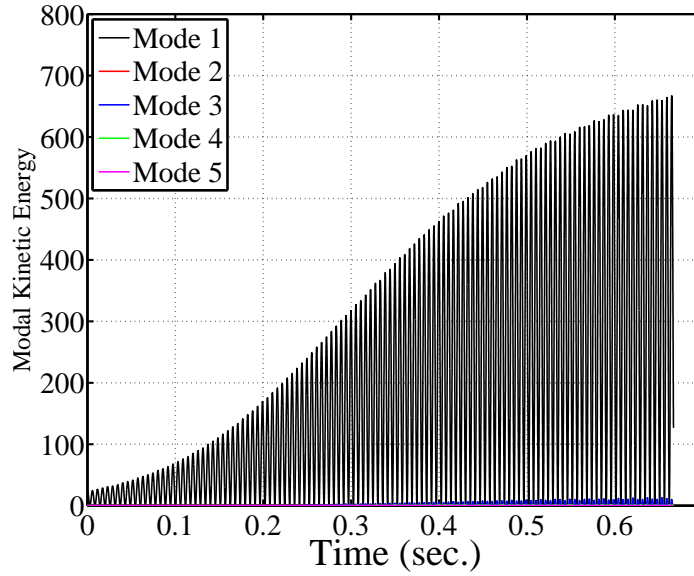


Figure 6.21: Modal kinetic energy of the NEXST-2 wing at Mach 1.02, $q = 59.7$ kPa.

6.25, the effect of increasing the Mach number is illustrated. The supersonic region and terminating shock persist on the wing for a larger portion of the oscillation than they did for the Mach 0.96 case. The shock is also located further aft, and reaches the trailing edge, but only for a small part of the oscillation cycle (times (f)–(h)). The power distribution oscillates between positive and negative on the outboard portion of the wing, with a thin sliver of the trailing edge affected by the shock moving over it. Because the shock does not attach itself to the wing trailing edge, it is free to move on the wing surface, albeit only over a very small part. It is this motion that keeps the wing in the transonic dip under these conditions. As will be shown later, a slight increase in Mach number or deformation pushes the shock further back on the wing, terminating the transonic dip response.

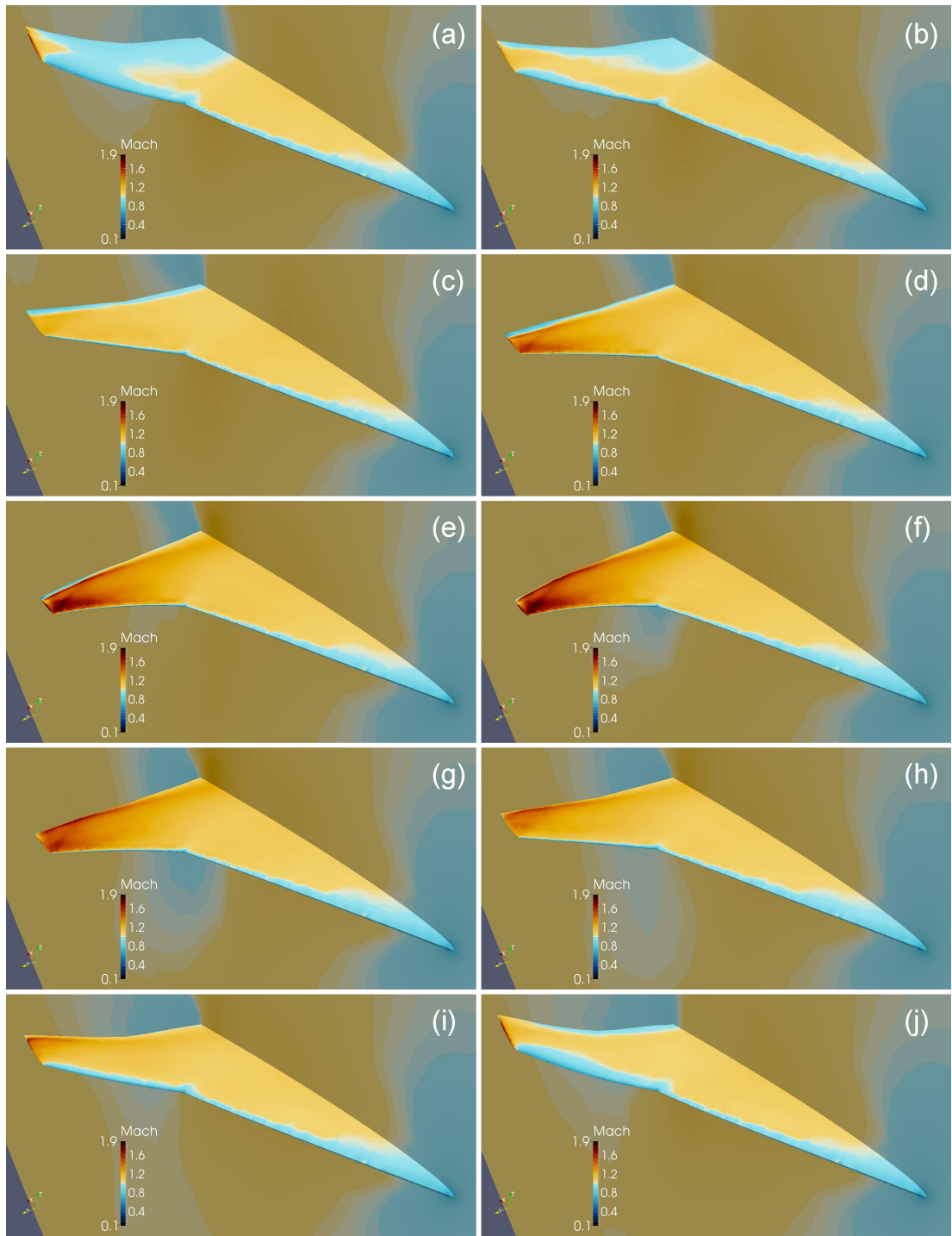


Figure 6.22: Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.02, $q = 59.7$ kPa.

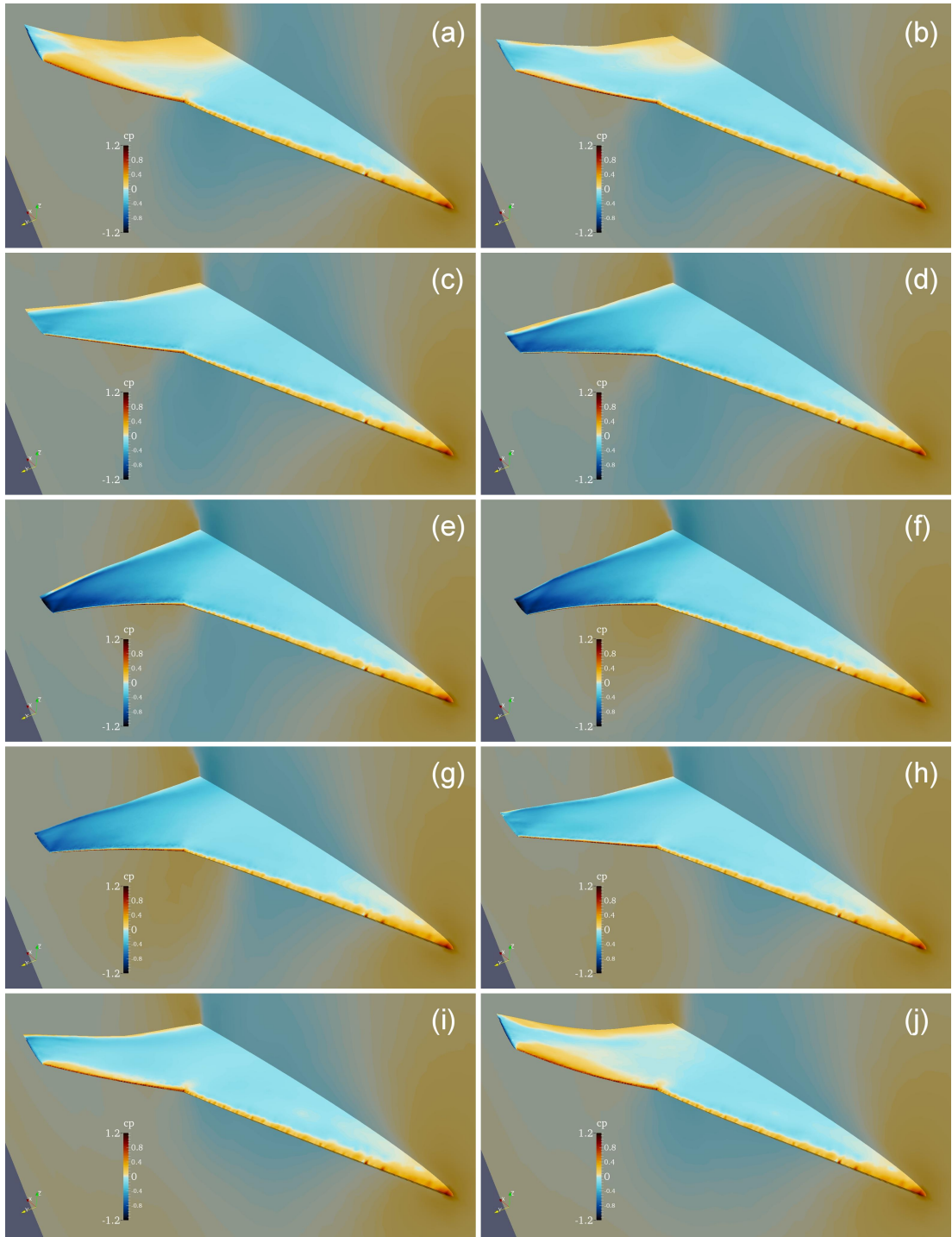


Figure 6.23: Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.02, $q = 59.7$ kPa.

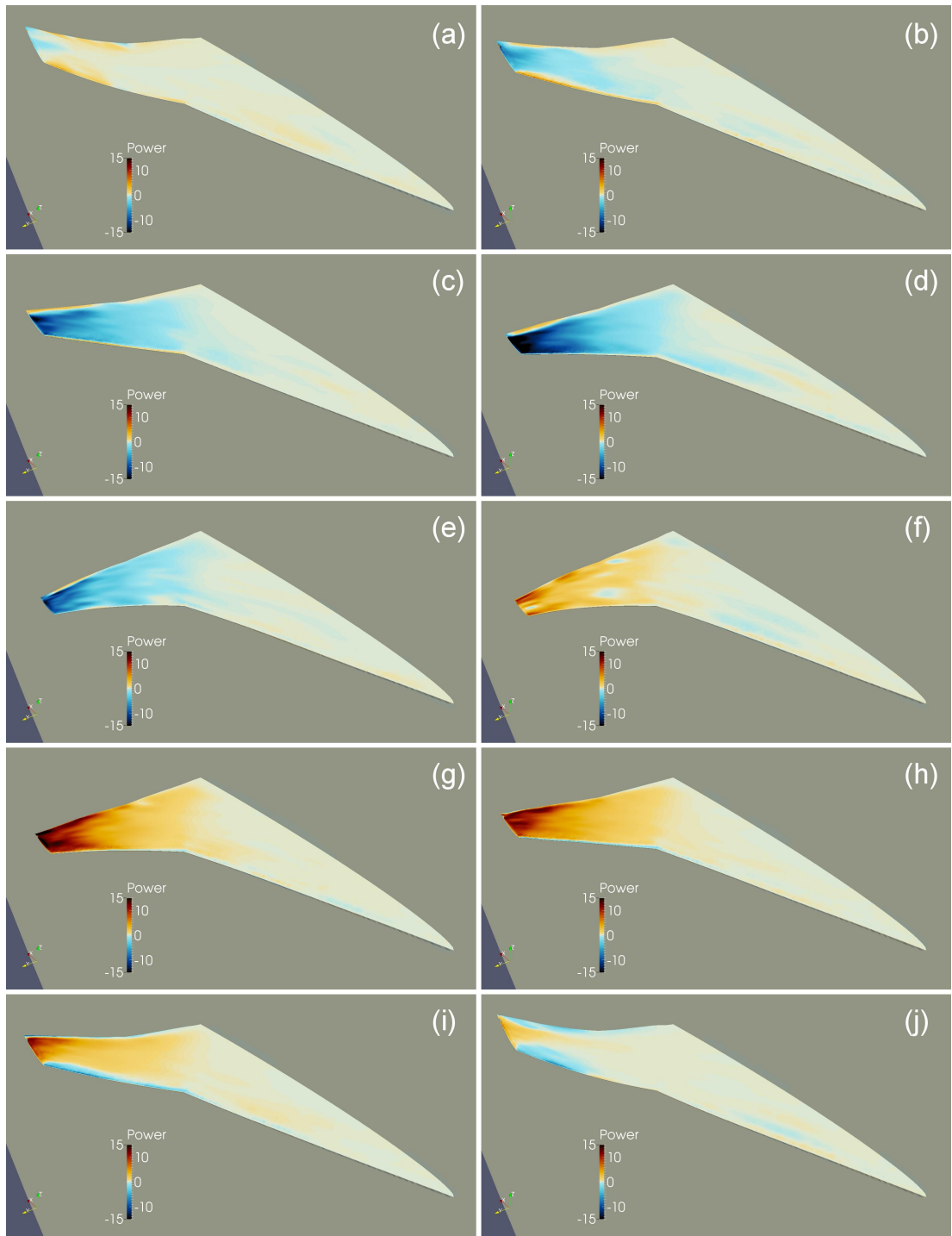


Figure 6.24: Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.02, $q = 59.7$ kPa.

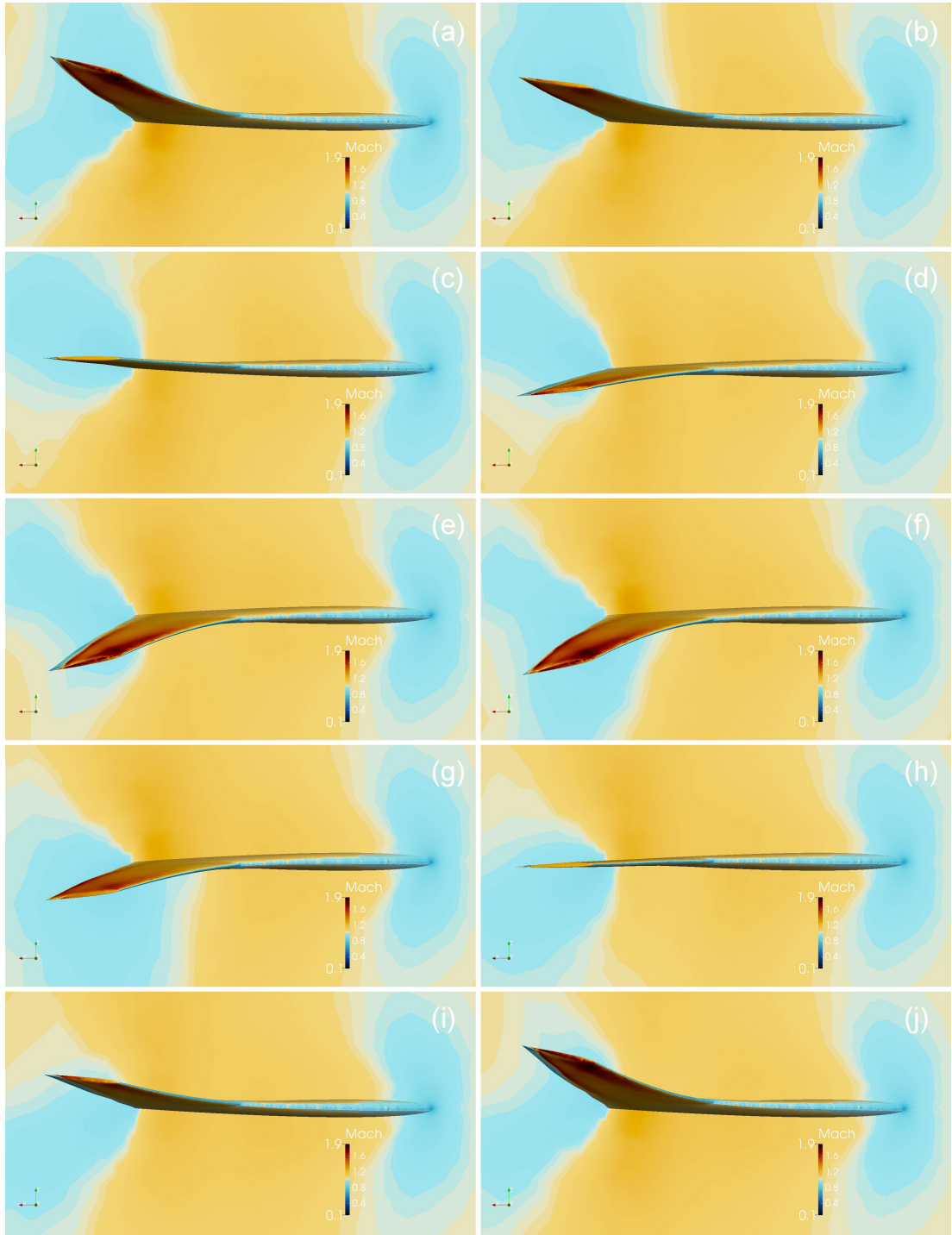


Figure 6.25: NEXST-2 wing deformation as viewed from beyond the wing tip looking toward the root during a period of limit cycle flutter at Mach 1.02, $q = 59.7$ kPa.

6.2.2.2 Subsonic Bending-Torsion Responses

Outside of the transonic dip region, the wing responds very differently to the moving flow. At Mach numbers below the transonic dip, the only regions of supersonic flow or shocks on the wing are near the leading edge, and for dynamic pressures below 200 kPa, the wing response is stable. At higher dynamic pressures, a subsonic bending-torsion flutter region exists. Responses typical of this region are shown in Figures 6.26–6.32. These responses were generated at Mach 0.70 with a dynamic pressure of 295.3 kPa, corresponding to point *C* in Figure 6.7.

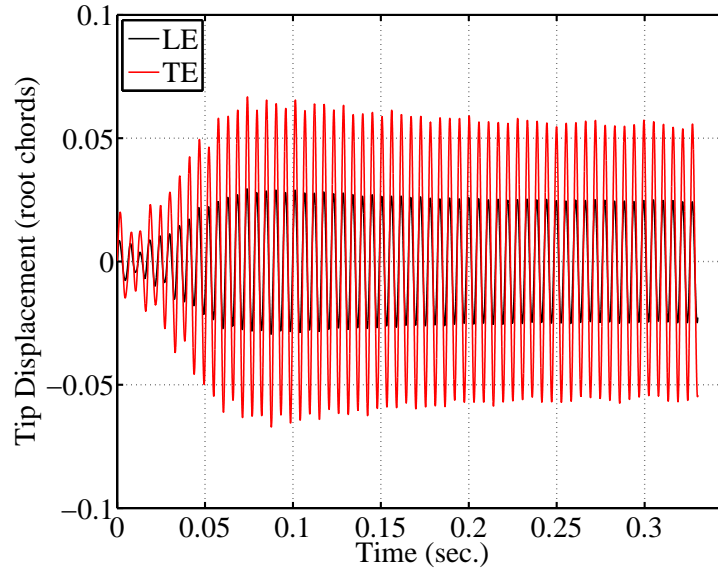


Figure 6.26: NEXST-2 wing tip time histories in the subsonic bending-torsion region at Mach 0.70, $q = 295.3$ kPa.

Wing tip time histories are plotted in Figure 6.26. The simulation is started from zero displacement with an initial velocity in the first mode, and the wing tip oscillations initially decrease in amplitude. After three cycles, the response

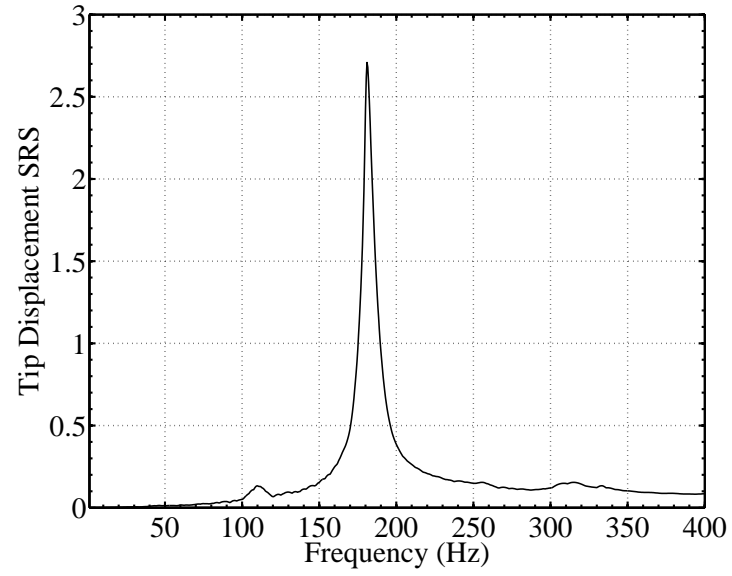


Figure 6.27: SRS of the NEXST-2 wing tip trailing edge at Mach 0.70, $q = 295.3$ kPa.

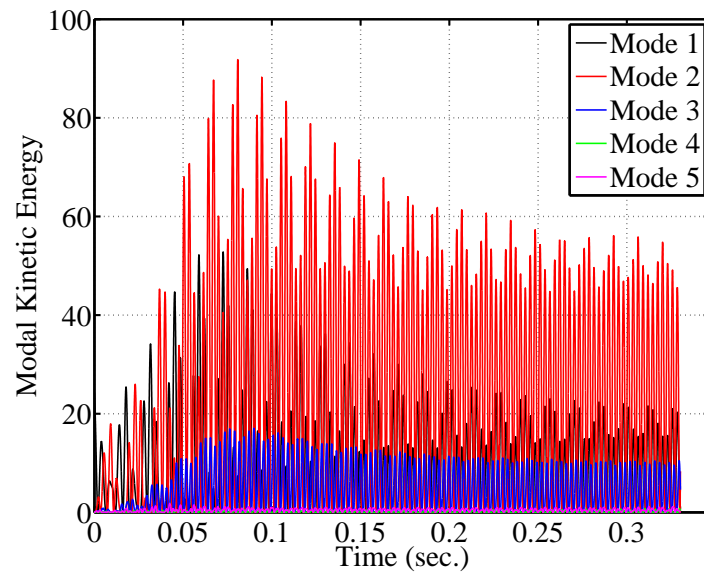


Figure 6.28: Modal kinetic energy of the NEXST-2 wing at Mach 0.70, $q = 295.3$ kPa.

begins to quickly grow at a rate nearly double what was seen for responses in the transonic dip region. Rapid growth like this is typical for bending-torsion flutter, and it often leads to destruction of the wing. After approximately ten cycles, the response magnitude stops growing and decreases slightly before settling at a stable LCO. The frequency of the response, shown in Figure 6.27, is just below the frequency of the third mode of the wing. Much like the wind tunnel test results shown in Figure 6.7, the wing responds at a higher frequency in the bending-torsion flutter region than it does in the transonic dip region. Examination of the modal kinetic energy in Figure 6.28 shows that the response is composed of multiple modes. The primary contributor is the second mode (second bending) with participation from the first (first bending) and third (first torsion) modes. Though the modes are identified as either bending or torsion modes, this is done for explanation only. The first four modes all contain both bending and torsion components on the outboard portion of the wing (Figure 6.5). Pictures of the wing deflection during one cycle are shown in Figures 6.29–6.32. The deflection is different than that observed in the transonic dip region, with the second and third modes contributing significantly more outboard wing twist than the first mode alone provides.

The nonlinear LCO response appears to be due to a mix of aerodynamic and structural nonlinearity. The flow field around the wing is fully subsonic with the exception of a small region of supersonic flow that forms near the wing leading edge during brief periods when the deflection and local angle of attack are maximized. The supersonic region can be seen in Figure 6.29(e)–(g). The aerodynamic power distribution in Figure 6.31 shows that the supersonic region results in large positive

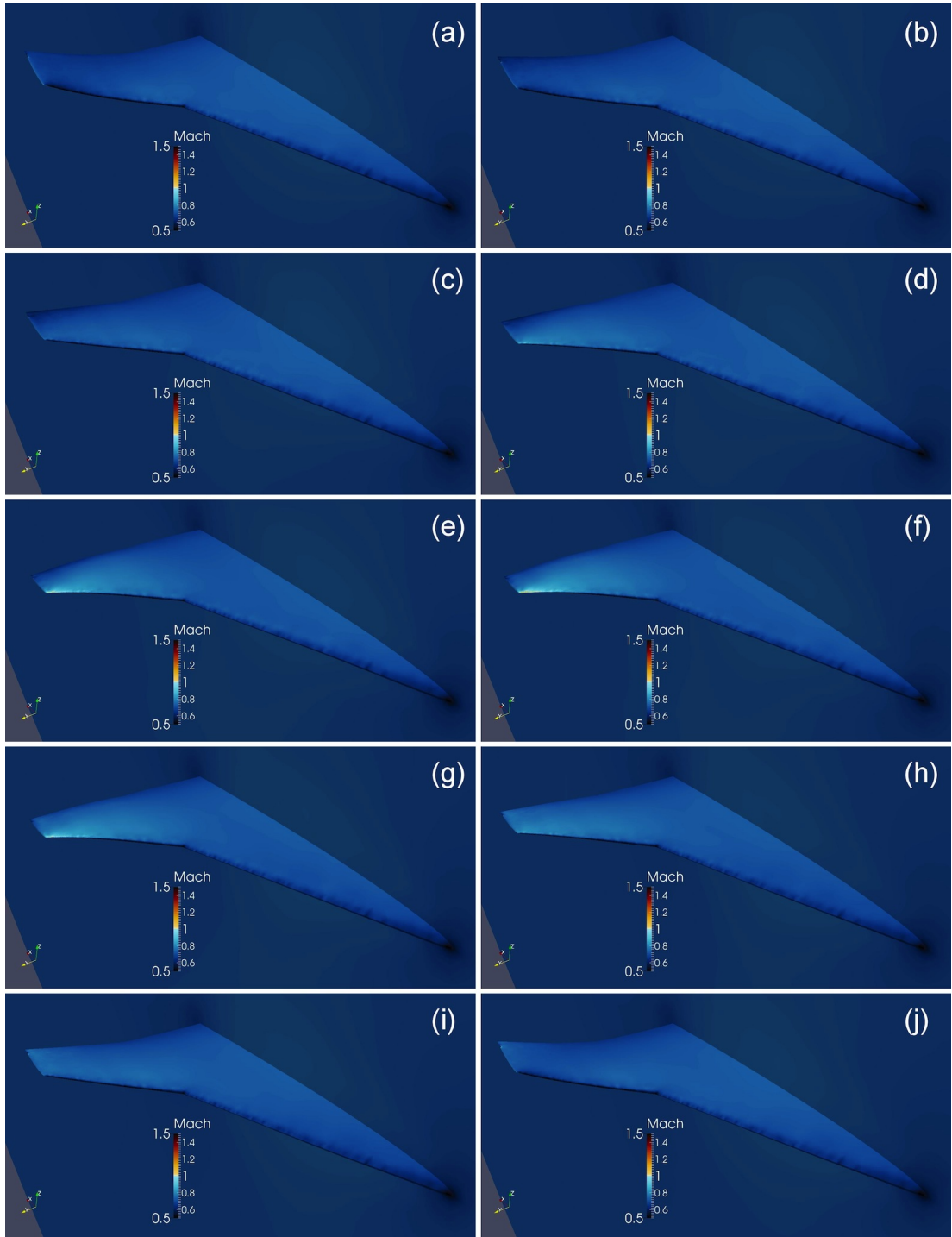


Figure 6.29: Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.7, $q = 295.3$ kPa.

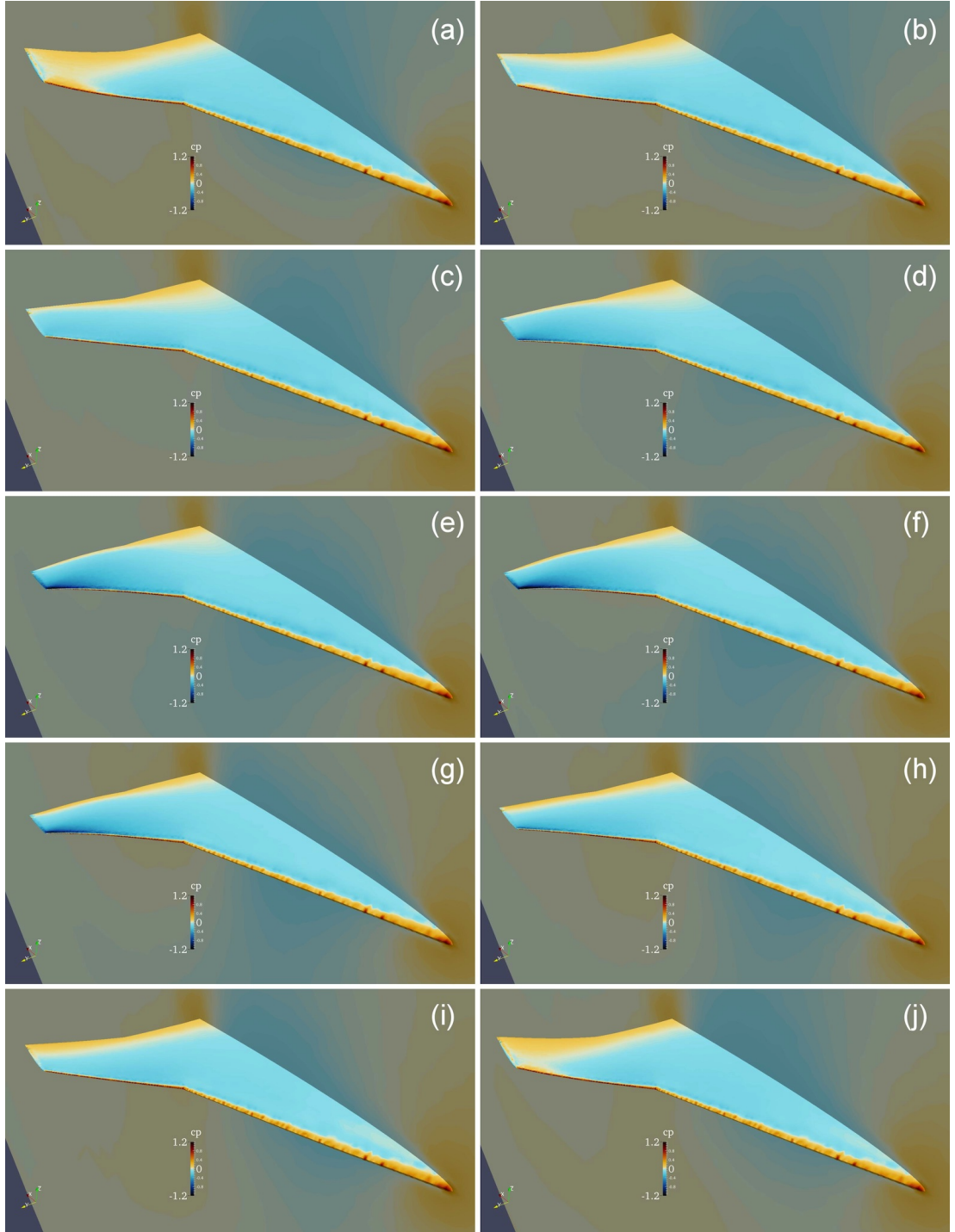


Figure 6.30: Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.7, $q = 295.3$ kPa.

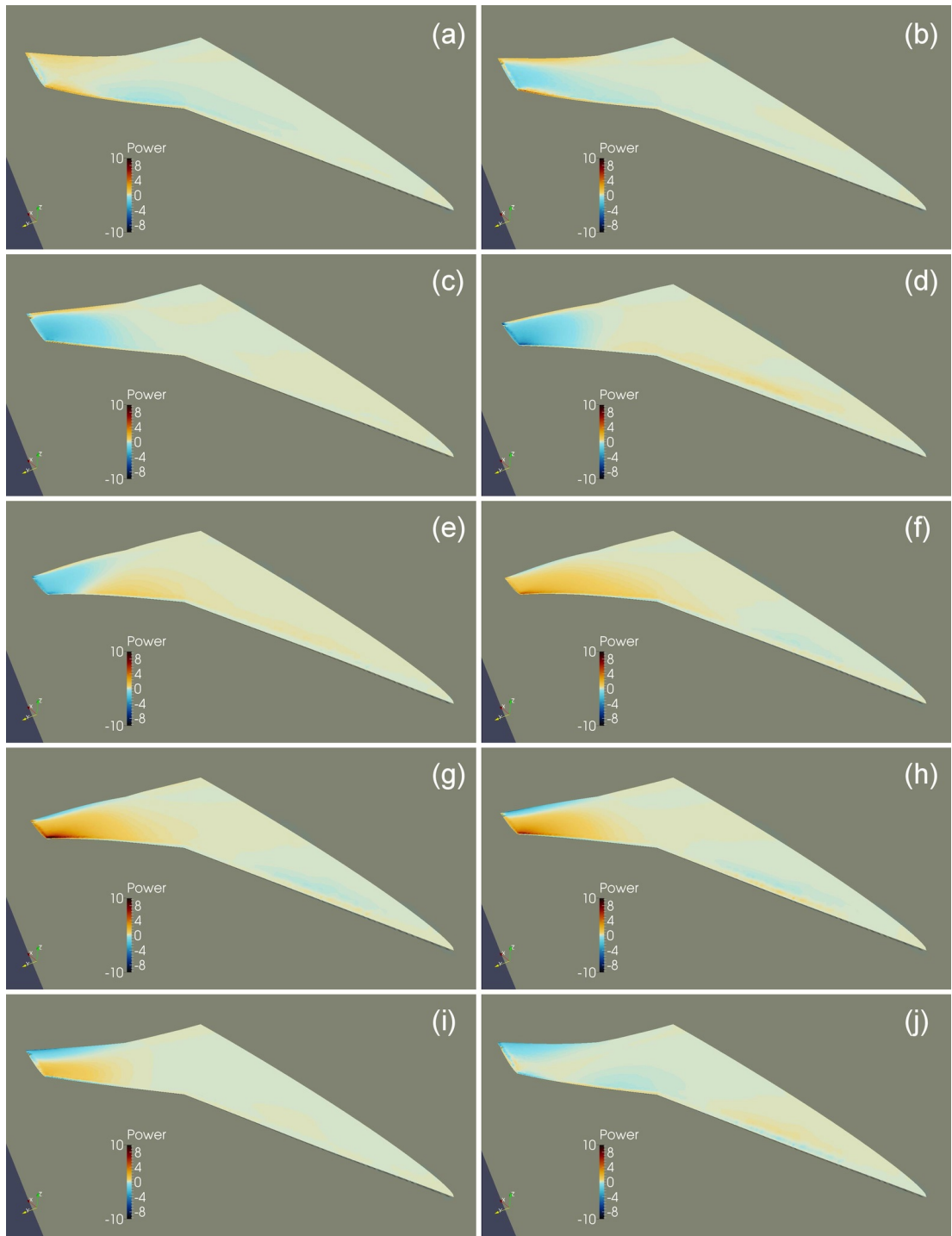


Figure 6.31: Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.7, $q = 295.3$ kPa.

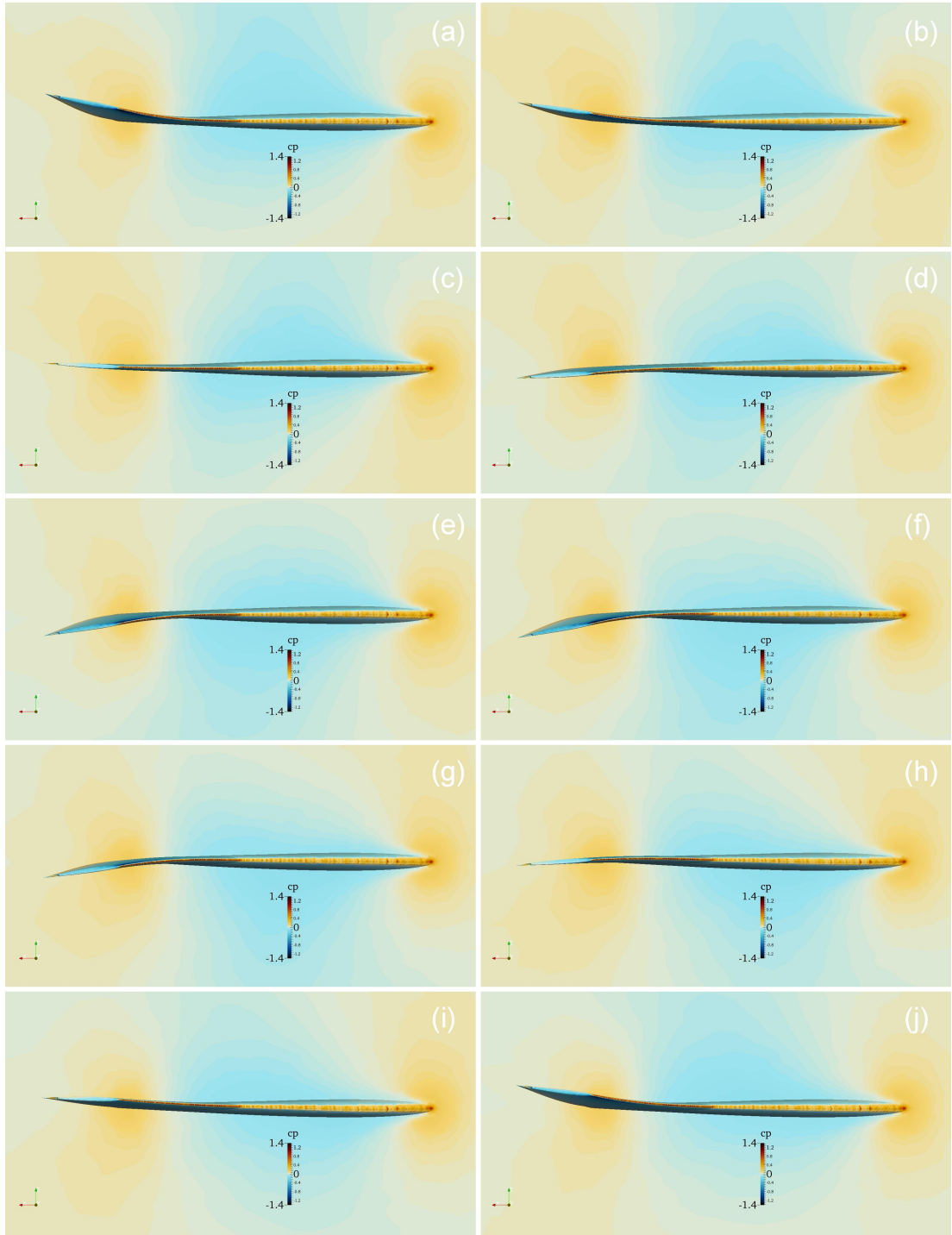


Figure 6.32: NEXST-2 wing deformation as viewed from beyond the wing tip looking toward the root during a period of limit cycle flutter at Mach 0.7, $q = 295.3$ kPa.

and negative energy transfer to the wing over a small area. Because the supersonic region is small, it is unlikely that it alone is the cause of the nonlinearity. It is believed that structural nonlinearity plays a large role as well. This seems reasonable due to the nature of the deflection. Because the deflection occurs over a small region of the wing, the local strains and the associated nonlinear effects are large.

The subsonic bending-torsion flutter region intersects with the transonic dip region at Mach numbers above 0.8. Responses from a simulation in this overlapping region are shown in Figures 6.33–6.39. The flow condition for these responses is Mach 0.85 with a dynamic pressure of 223.9 kPa, and is labeled as point F in Figure 6.7.

The wing tip time history in Figure 6.33 shows that the initial response magni-

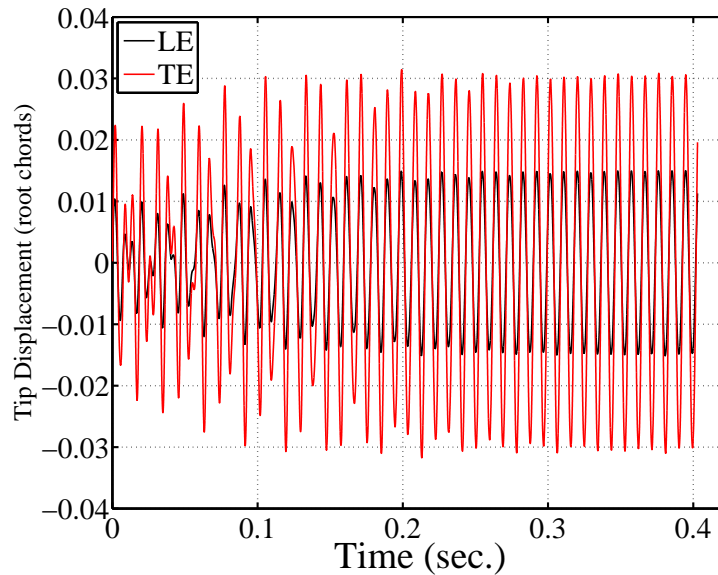


Figure 6.33: NEXST-2 wing tip time histories at Mach 0.85, $q = 223.9$ kPa.

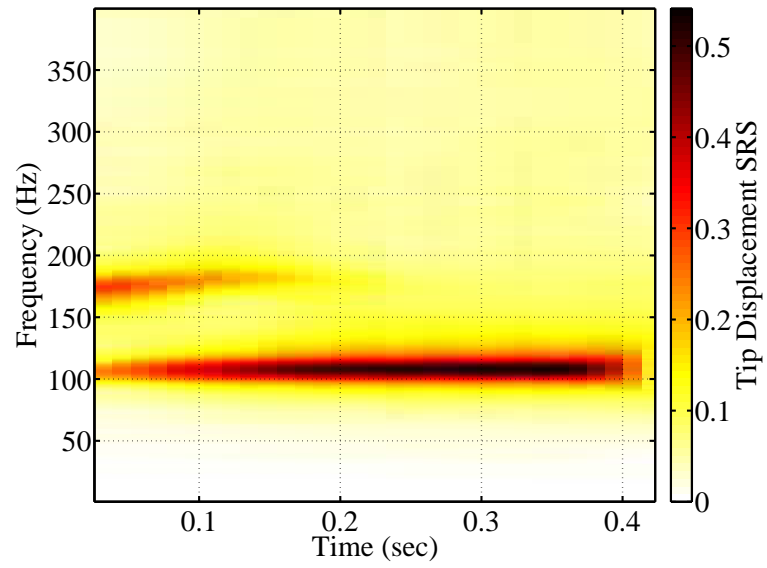


Figure 6.34: Waterfall SRS of the NEXST-2 wing tip trailing edge at Mach 0.85, $q = 223.9$ kPa.

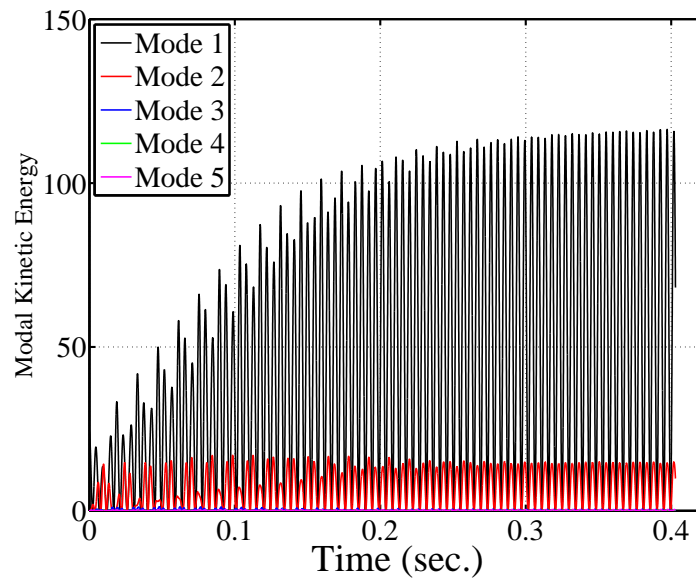


Figure 6.35: Modal kinetic energy of the NEXST-2 wing at Mach 0.85, $q = 223.9$ kPa.

tude varies from one oscillation cycle to the next. The variability in the response eventually subsides and a stable limit cycle is reached. A waterfall SRS of the wing tip trailing edge in Figure 6.34 shows how the frequency of the response changes over time. Initially, there are two distinct frequencies. It appears that the first mode response from the transonic dip region and the bending-torsion response from the subsonic bending-torsion region are both present at the start of the simulation. Based on Figure 6.34, one might be led to believe that the bending-torsion response decays, and only the transonic dip response remains, but the modal kinetic energy time histories show a different behavior (Figure 6.35). At the beginning of the simulation, the first and second mode participate in the response at a similar magnitude, but at different frequencies. As the response of the first mode grows, the response in the second mode does not decay, but its phase and frequency shift and lock in with the first mode. This behavior is typical in the areas where the response regions intersect.

Mach distribution on the wing during one period of LCO is shown in Figure 6.36. The wing deformation for this case is small compared to the other cases in the transonic dip region and is difficult to see from the viewpoint in Figure 6.36. Deformation can be seen more easily in Figure 6.39. The images show that a supersonic region exists on the wing only when the wing tip is either at the top or bottom of an oscillation (it is the bottom in this case since we are looking at the upper surface of the wing). Examination of pressure coefficient and aerodynamic power distributions in Figures 6.37 and 6.38 show that the largest negative and positive work is done in the region of the supersonic flow and the shock, indicating that the shock plays a large role in the energy transfer between the wing and the

flow.

These responses exemplify the shift of the transonic dip to lower Mach numbers at higher dynamic pressure. The increased wing twist caused by the participation of the second mode pushes the shock further aft on the wing than it is at lower dynamic pressures. In this case, the shock moves sufficiently aft for enough of the cycle to result in transonic dip responses.

6.2.2.3 Supersonic Bending-Torsion Responses

At Mach numbers above the transonic dip region, the flow in front of the wing is supersonic. Except in cases of large deformation at lower supersonic Mach numbers (1.04-1.06), the flow on the entire wing surface is supersonic. Even for moderate changes in angle of attack, the shock tends to stay attached to the trailing edge. This phenomenon was recently observed by Bendiksen as an effect of the stabilization law for transonic flutter [56]. At these lower supersonic Mach numbers, the wing is stable at lower dynamic pressures. A supersonic bending-torsion flutter region exists at higher dynamic pressures. The boundary of this region moves to much higher dynamic pressures as the Mach number is increased (Figure 6.7).

Responses typical of the supersonic bending-torsion flutter region are shown in Figures 6.40–6.46. These responses occur at Mach 1.15 with a dynamic pressure of 288.4 kPa. This condition is represented by point *D* in Figure 6.7. The figures show limit cycle flutter with nearly equal participation from the first (first bending) and second (second bending) modes of the wing. The third (first torsion) mode is also involved, but at a smaller magnitude. This response is similar

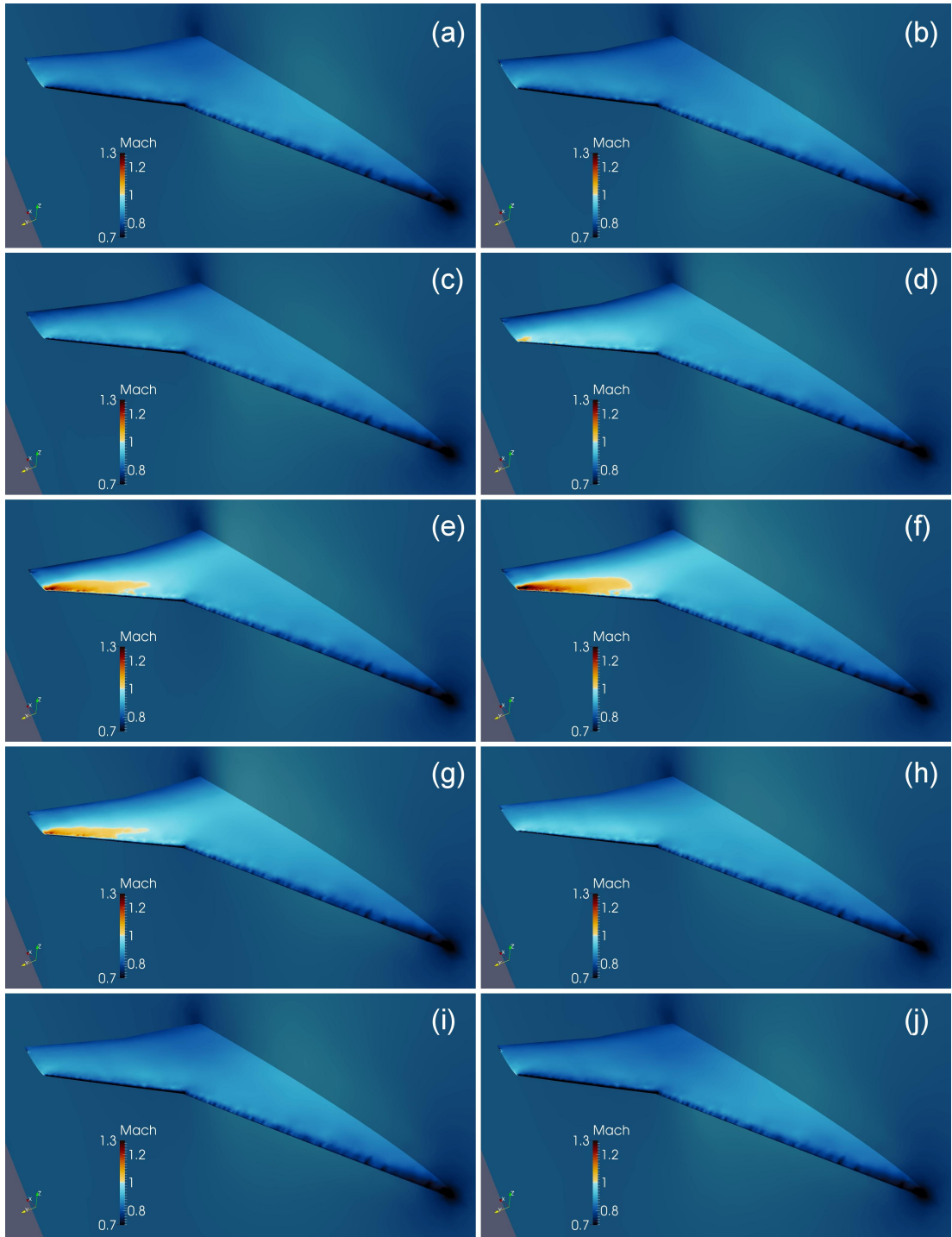


Figure 6.36: Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.85, $q = 223.9$ kPa.

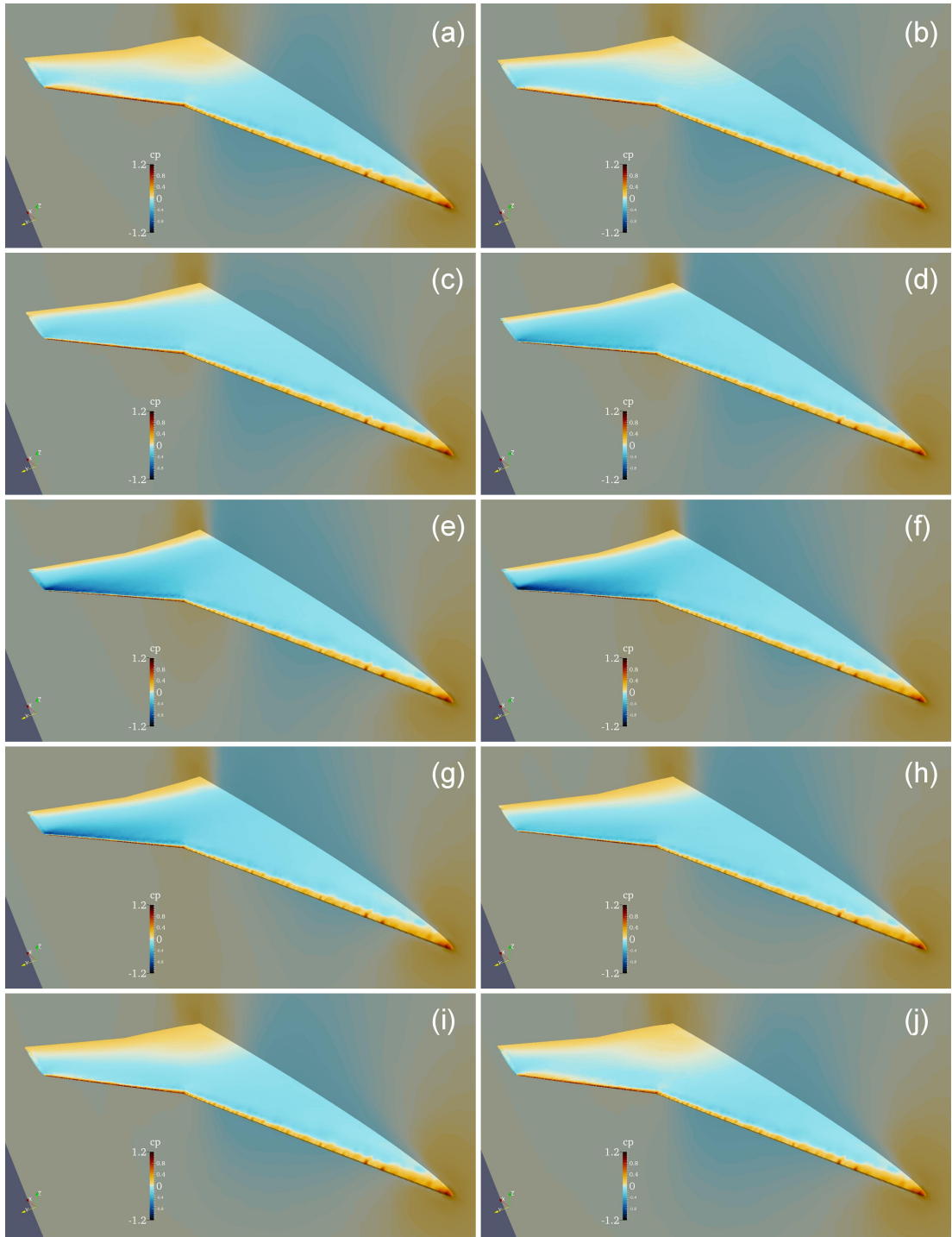


Figure 6.37: Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.85, $q = 223.9$ kPa.

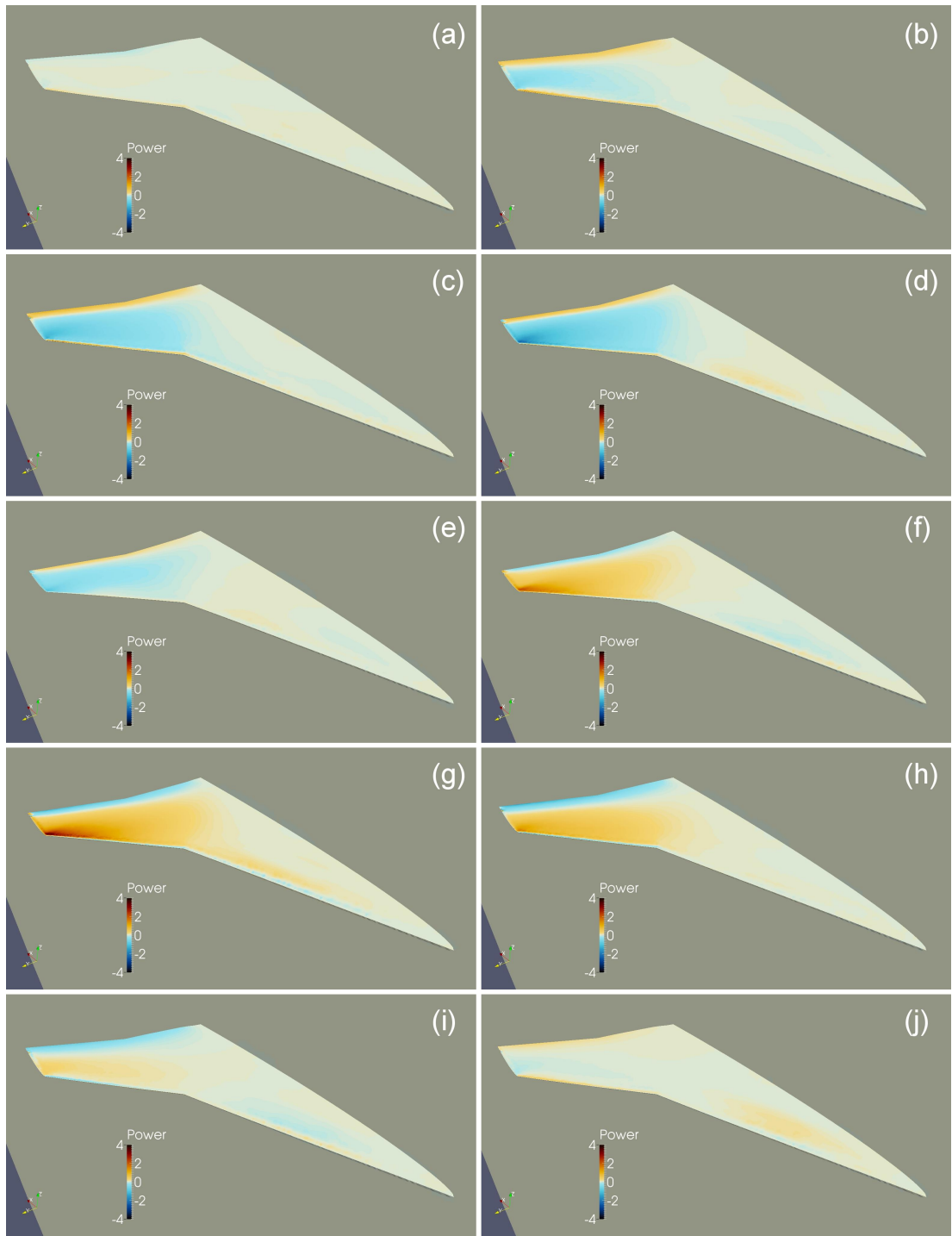


Figure 6.38: Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 0.85, $q = 223.9$ kPa.

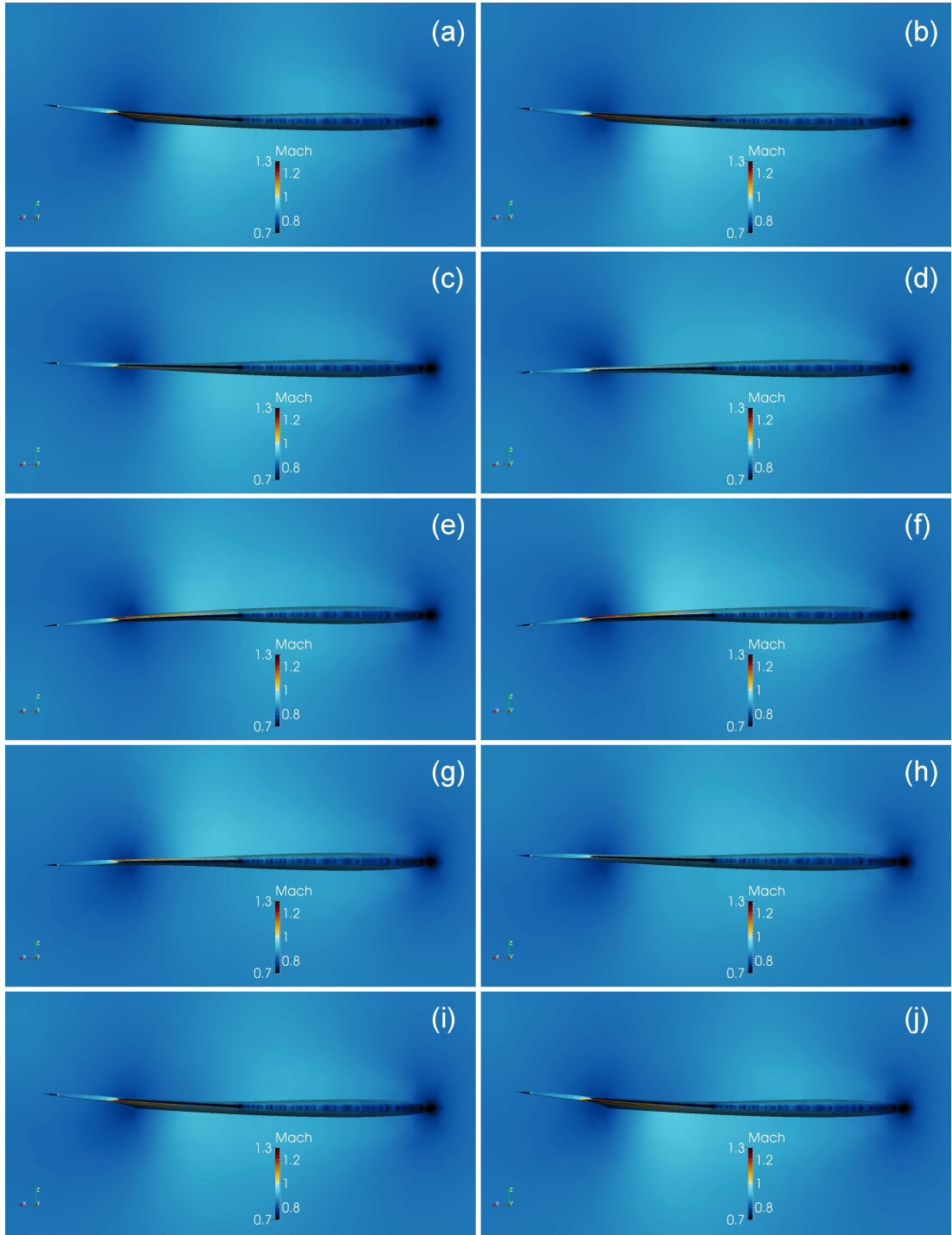


Figure 6.39: NEXST-2 wing deformation as viewed from beyond the wing tip looking toward the root during a period of limit cycle flutter at Mach 0.85, $q = 223.9$ kPa.

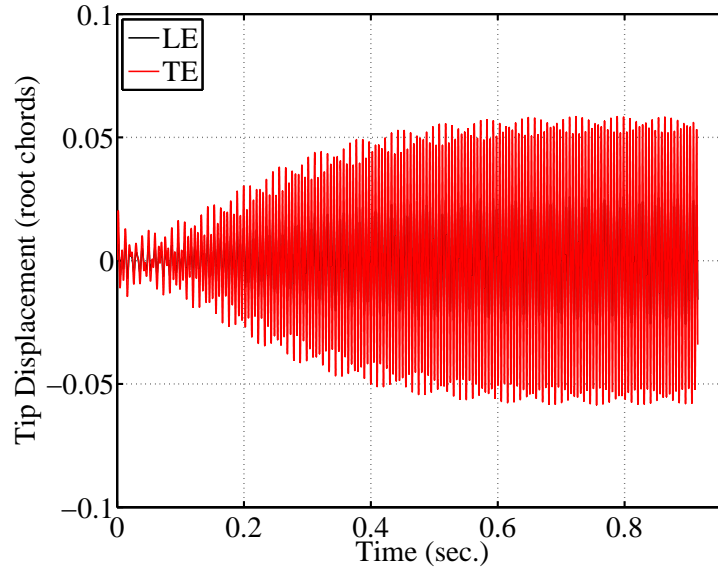


Figure 6.40: NEXST-2 wing tip time histories in the supersonic bending-torsion region at Mach 1.15, $q = 288.4$ kPa.

to that seen in the subsonic bending-torsion flutter region, where the first three modes participated in the response, though at different ratios. The frequency of the response is just below 200 Hz and slightly above the frequency of the third mode of the wing.

Mach, pressure, and power distributions are shown in Figures 6.43–6.45. The wing deformation can be seen more clearly in Figure 6.46. The time points shown represent one full cycle during LCO. The images show that, with the exception of the stagnation points on the leading edge, flow over the wing remains supersonic for the entire simulation. As a result, the pressure distribution on the wing is almost exclusively negative on both the top and bottom surfaces. Without the presence of some nonlinearity, flutter under these conditions typically results in increasing amplitudes without LCO. It is believed that nonlinear stiffening of the

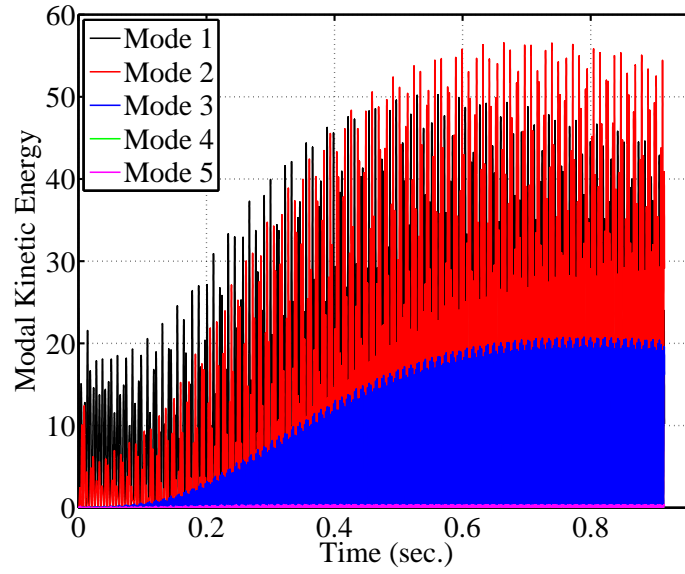


Figure 6.41: Modal kinetic energy of the NEXST-2 wing at Mach 1.15, $q = 288.4$ kPa.

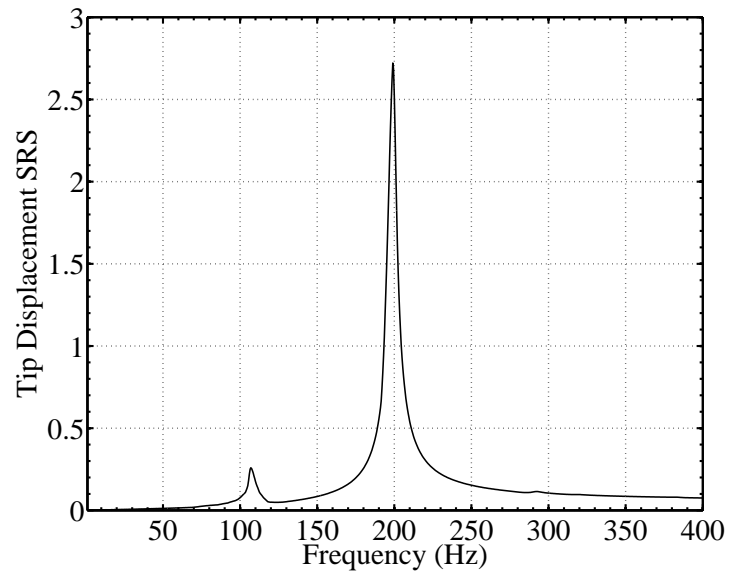


Figure 6.42: SRS of the NEXST-2 wing tip trailing edge at Mach 1.15, $q = 288.4$ kPa.

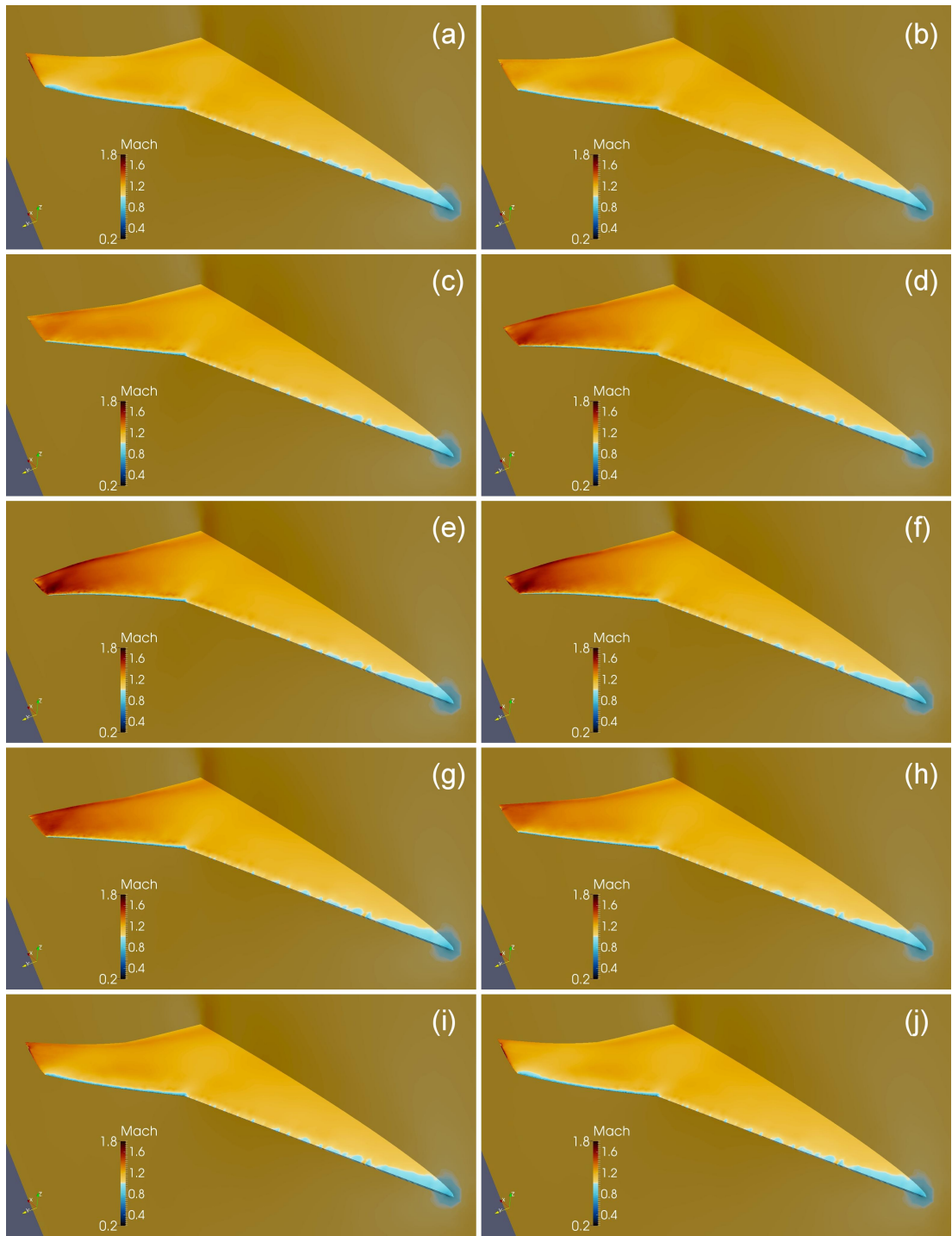


Figure 6.43: Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.15, $q = 288.4$ kPa.

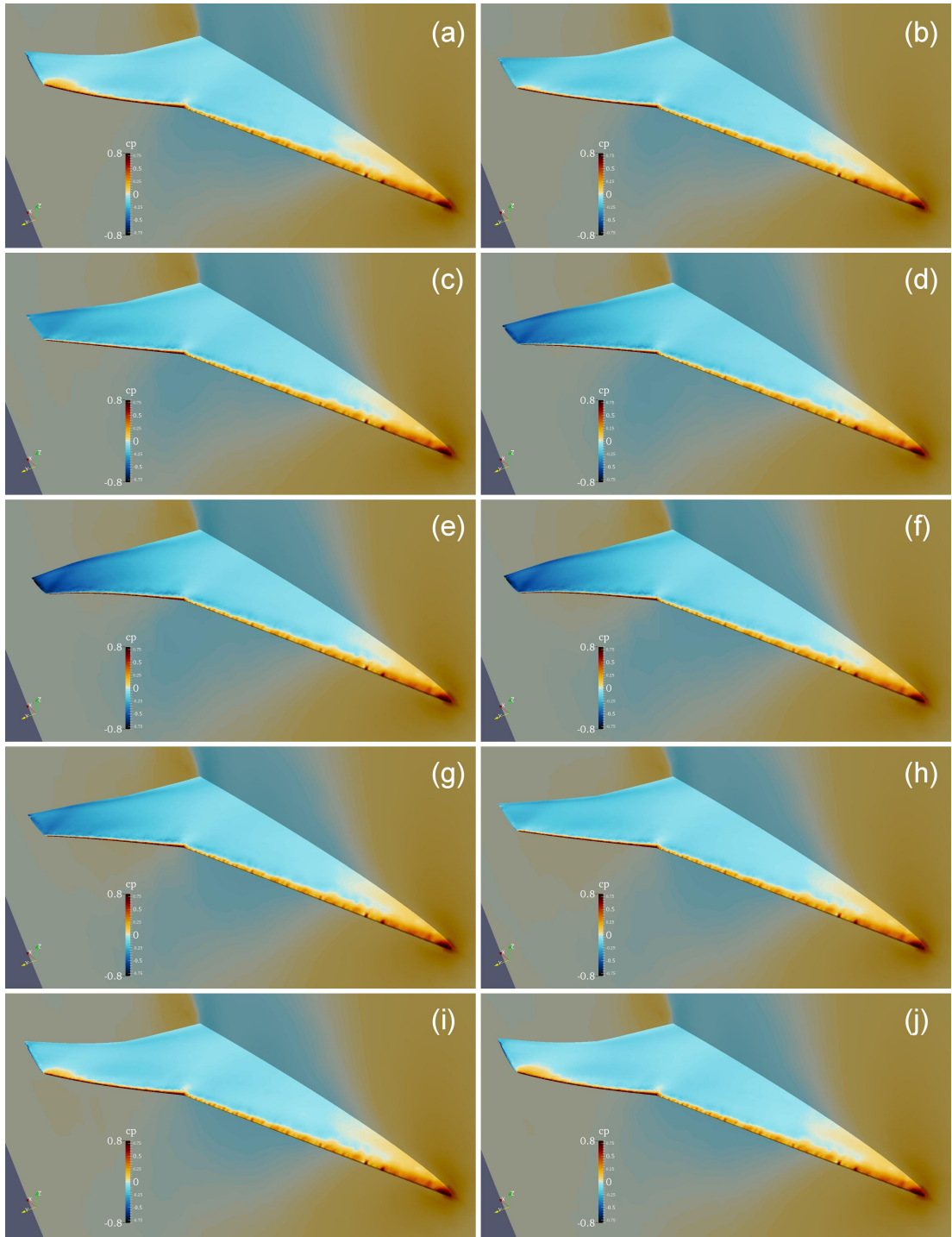


Figure 6.44: Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.15, $q = 288.4$ kPa.

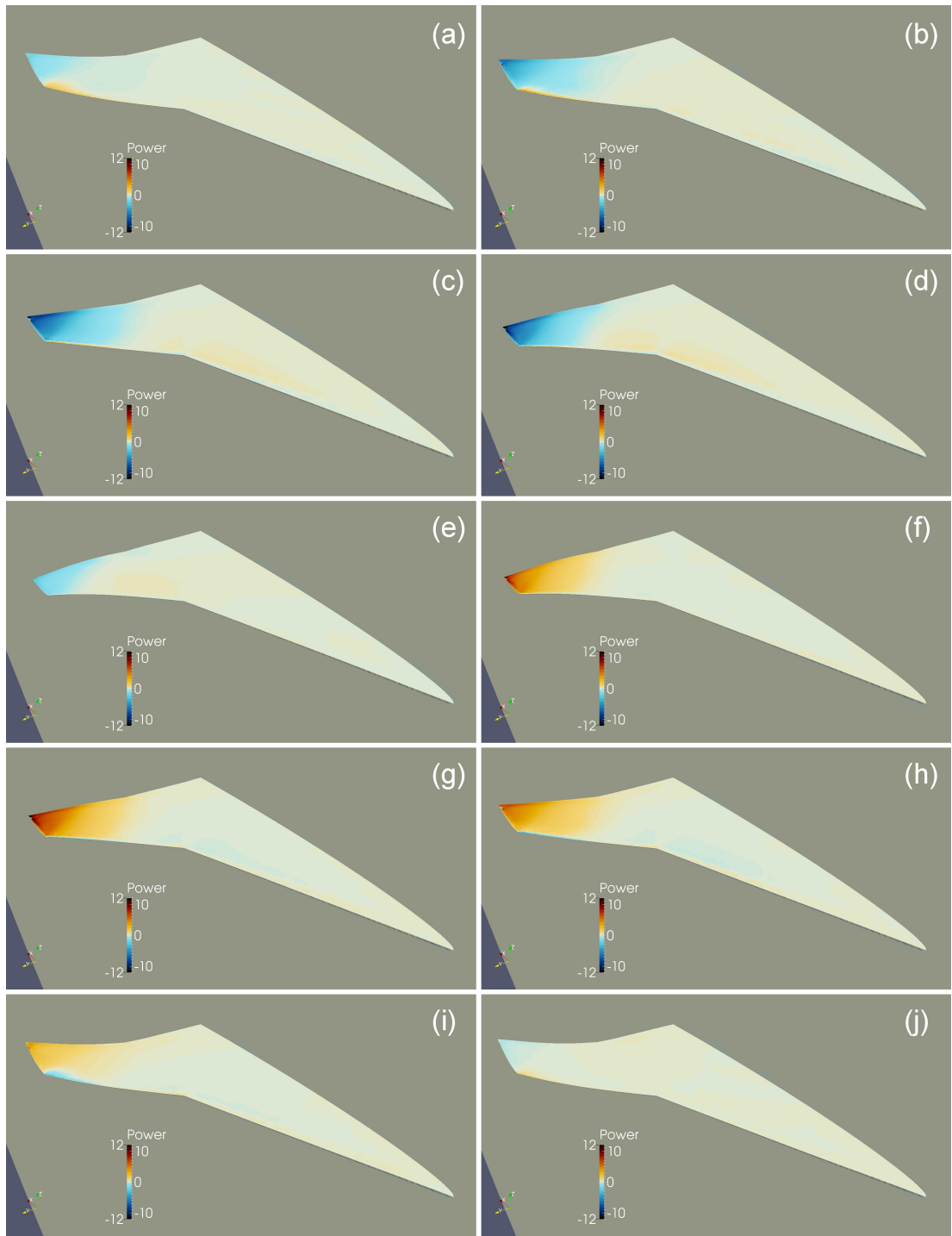


Figure 6.45: Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.15, $q = 288.4$ kPa.

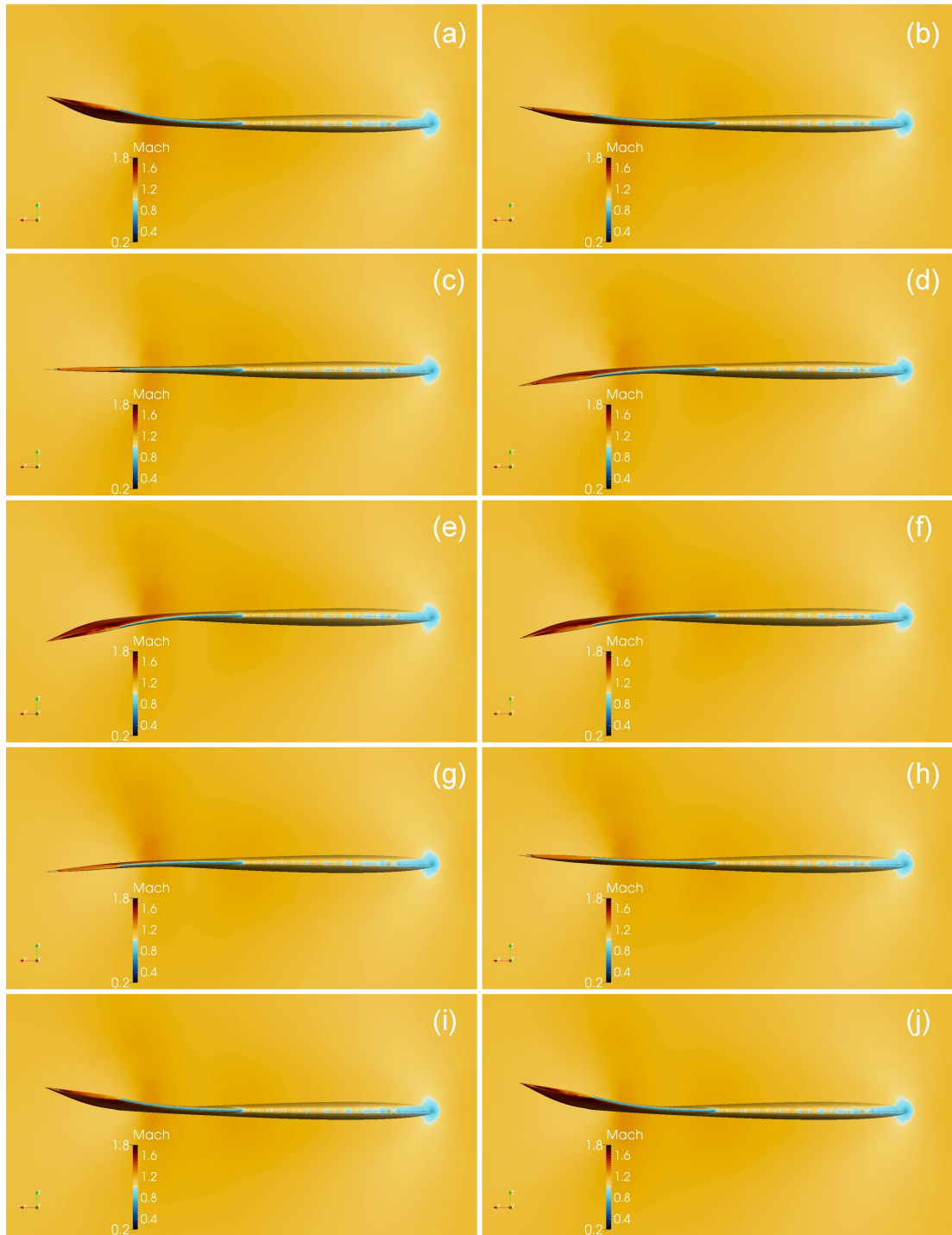


Figure 6.46: NEXST-2 wing deformation as viewed from beyond the wing tip looking toward the root during a period of limit cycle flutter at Mach 1.15, $q = 288.4$ kPa.

structure and consistent load vector application (follower forces) are responsible for the limited amplitudes seen in these results.

Previously, in Section 6.2.2.2, a response that exhibited characteristics of both the subsonic bending-torsion and transonic dip regions was discussed. In that case, the responses merged at a common frequency and existed together for the duration of the simulation. At the supersonic edge of the transonic dip, a very different behavior can be observed. Responses from the NEXST-2 wing at Mach 1.04 with a dynamic pressure of 136.6 kPa are shown in Figures 6.47–6.53. This flow condition is labeled in Figure 6.7 as point *E*.

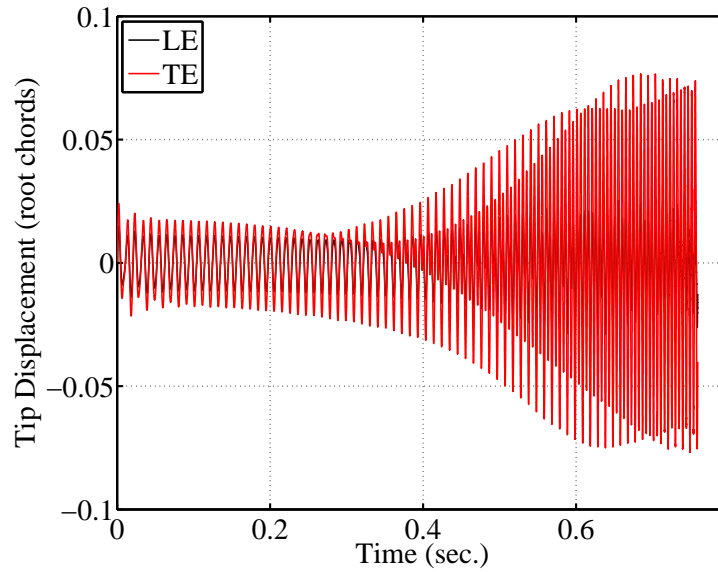


Figure 6.47: NEXST-2 wing tip time histories at Mach 1.04, $q = 136.6$ kPa.

For this flow condition, the initial response is a small amplitude LCO in the first mode of the wing (Figures 6.47 and 6.49). During this portion of the simulation, the shock is close to, but has not yet reached the trailing edge of the wing. The

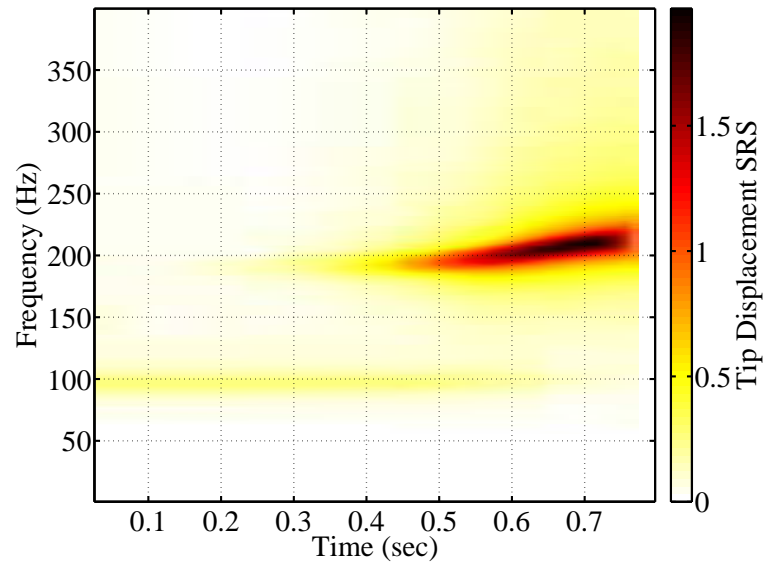


Figure 6.48: Waterfall SRS of the NEXST-2 wing tip trailing edge at Mach 1.04, $q = 136.6$ kPa.

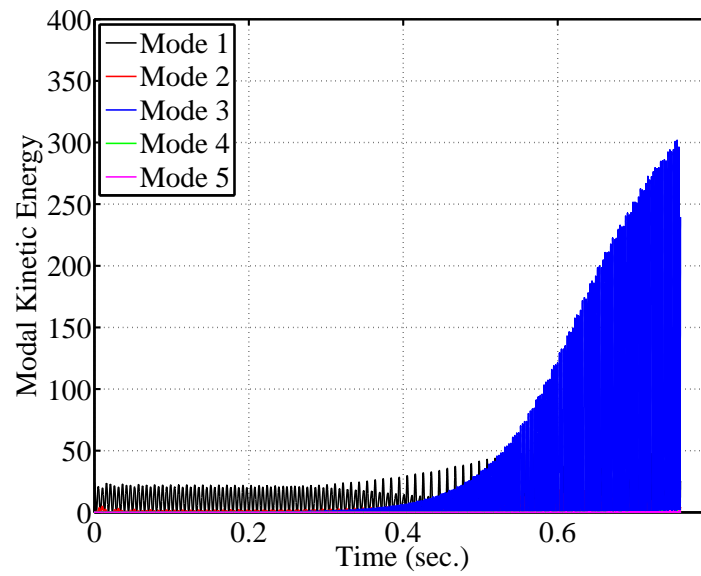


Figure 6.49: Modal kinetic energy of the NEXST-2 wing at Mach 1.04, $q = 136.6$ kPa.

response is identical to other responses in the transonic dip, but the deflections are much smaller. As the LCO amplitude slowly increases, the shock moves further aft on the wing. Eventually, the shock begins attaching to the trailing edge for part of the LCO cycle. This initiates a transition away from transonic dip type LCO to supersonic bending-torsion flutter. As the deformation increases, the shock spends a larger portion of each cycle attached to the trailing edge. Eventually, the wing is stabilized against the first mode, transonic dip response, but is suddenly under the correct conditions for supersonic bending-torsion flutter. At this point, the response of the third mode begins to rapidly increase, quickly surpassing the contribution of the first mode. The response grows so quickly and in such a manner that the fluid mesh cannot move properly to accommodate and the simulation crashes.

Mach distribution during a single period of the higher frequency LCO is shown in Figure 6.50. The images show that the shock moves away from the trailing edge for more than half the cycle. Despite this fact, the response is firmly of the supersonic bending-torsion nature, and no traces of the transonic-dip responses remain.

6.2.2.4 High Frequency Region Responses

At dynamic pressures below the subsonic bending-torsion flutter region at Mach 0.7 and 0.75, a high frequency limit cycle oscillation region was identified. The flow in this region is entirely subsonic, with no shocks forming on the wing. Responses typical of this region are shown in Figures 6.54–6.56. These results were generated at Mach 0.75 with a dynamic pressure of 129.1 kPa. This corresponds to point *G*

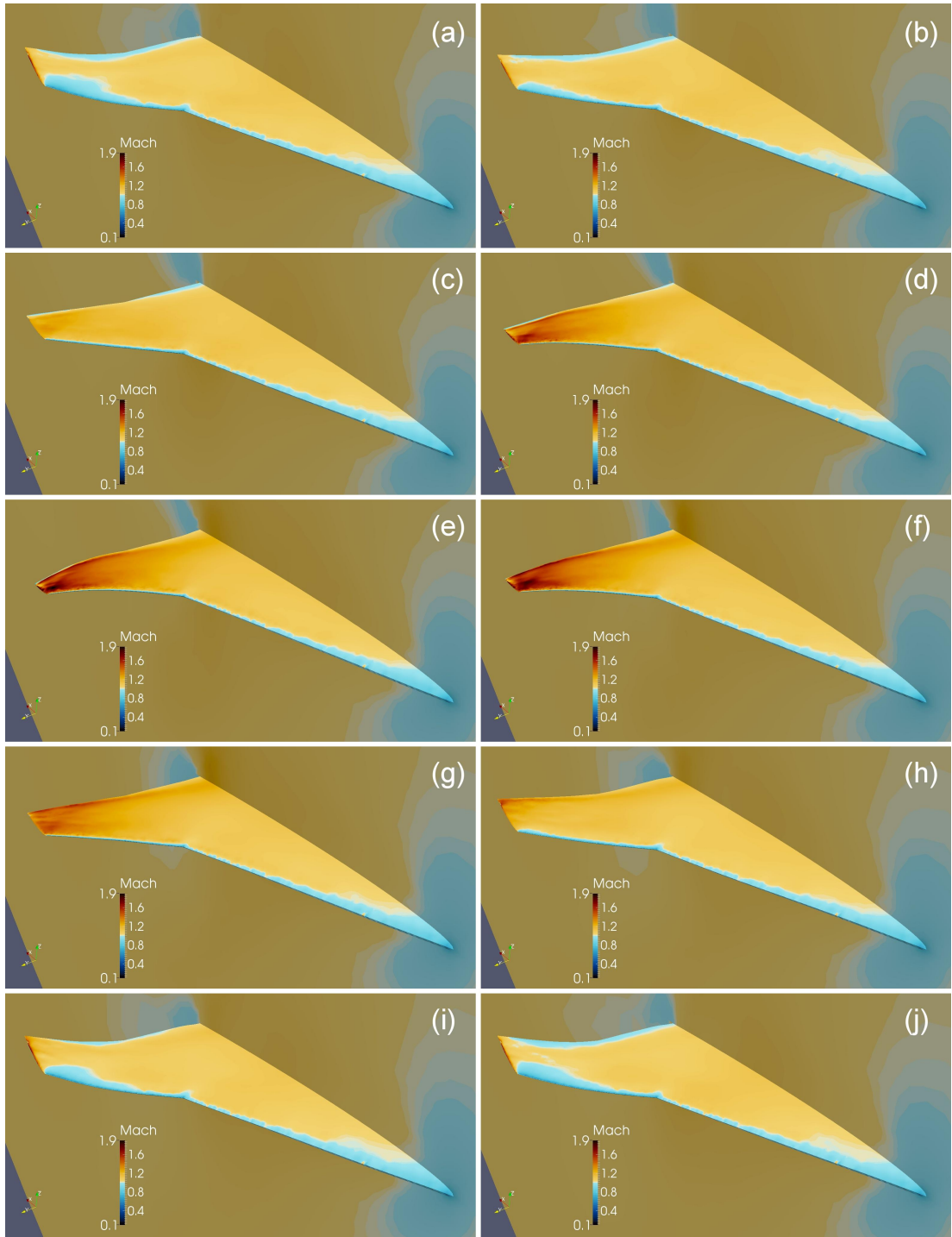


Figure 6.50: Mach distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.04, $q = 136.6$ kPa.

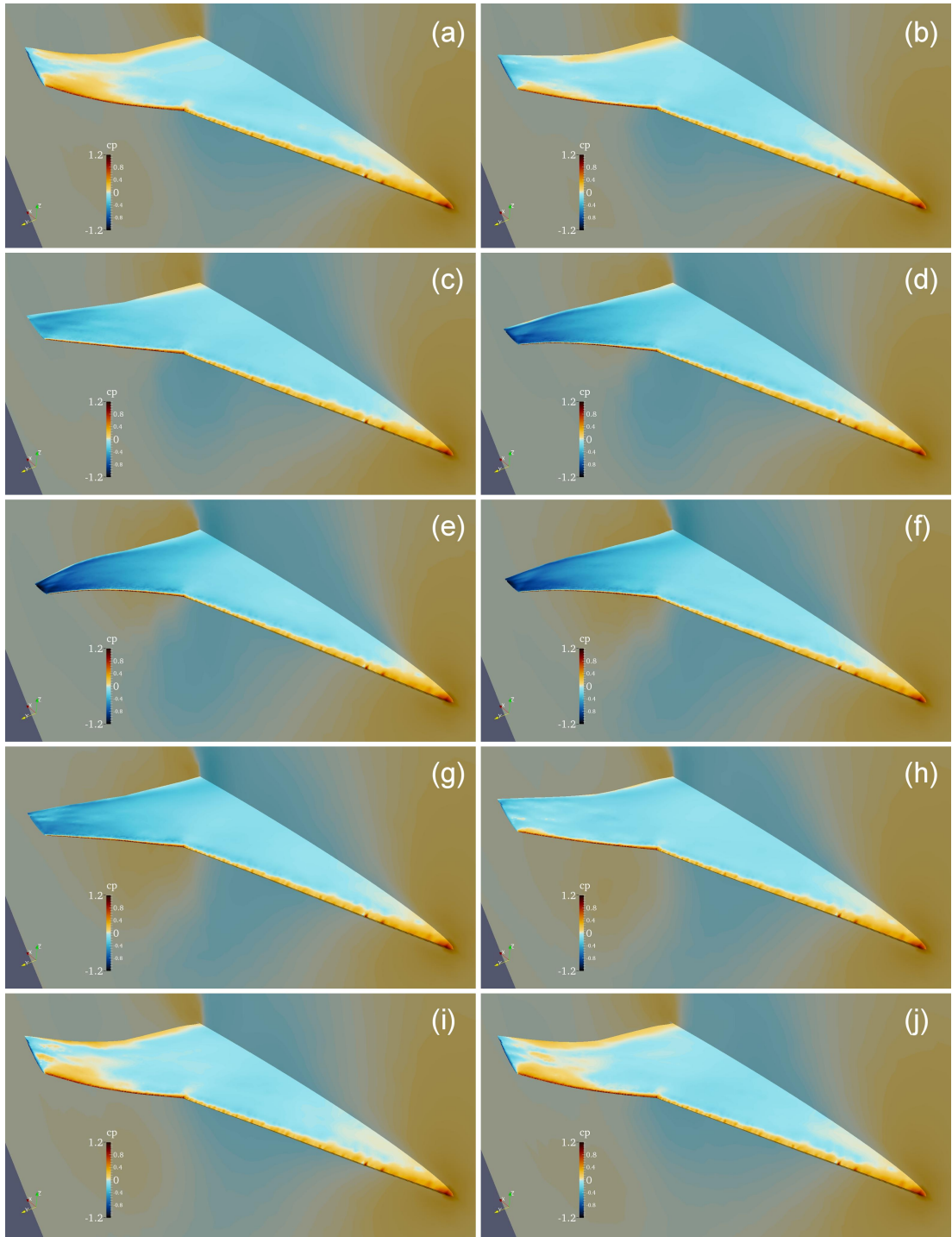


Figure 6.51: Pressure coefficient (C_p) distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.04, $q = 136.6$ kPa.

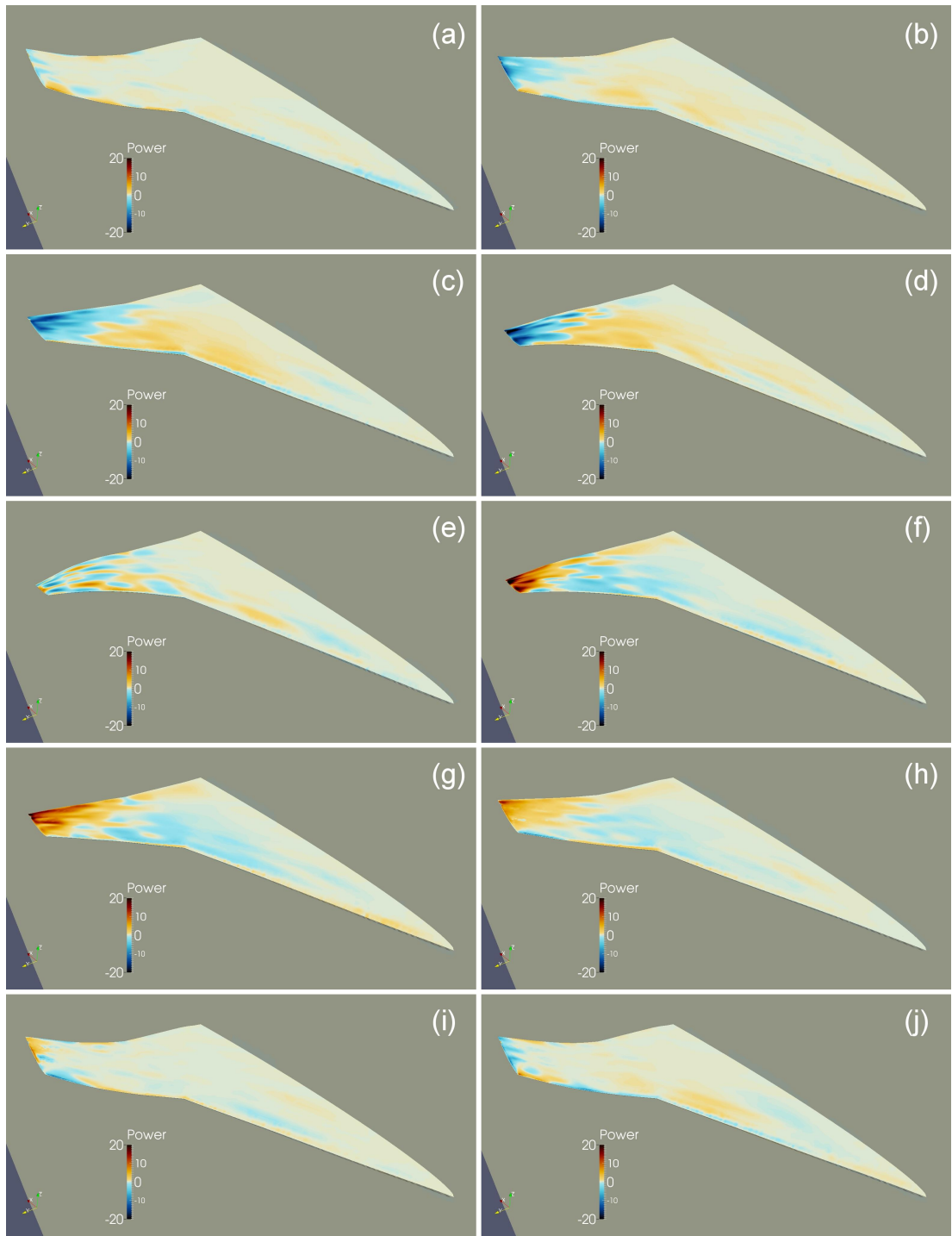


Figure 6.52: Aerodynamic power distribution on the NEXST-2 wing during a period of limit cycle flutter at Mach 1.04, $q = 136.6$ kPa.

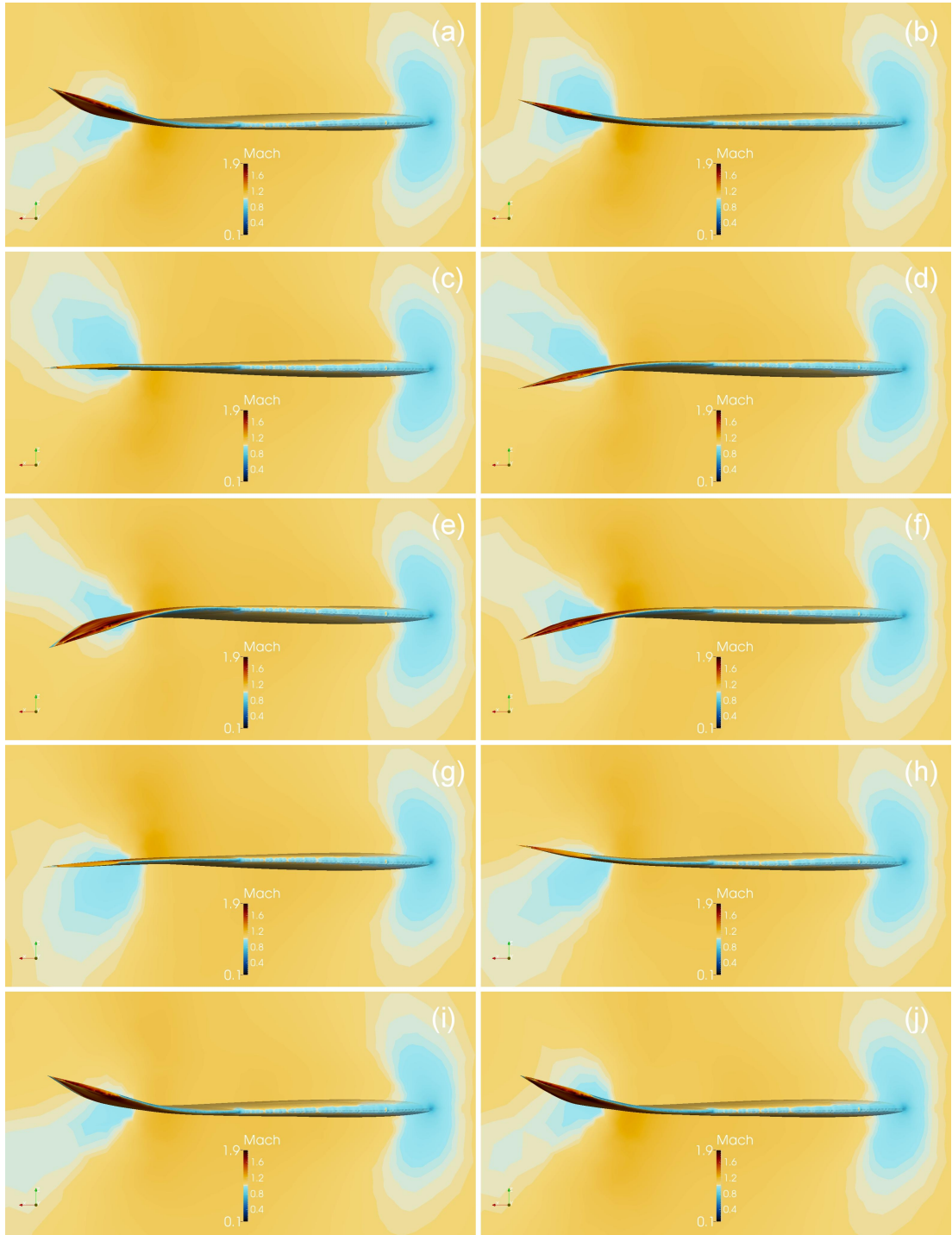


Figure 6.53: NEXST-2 wing deformation as viewed from beyond the wing tip looking toward the root during a period of limit cycle flutter at Mach 1.04, $q = 136.6$ kPa.

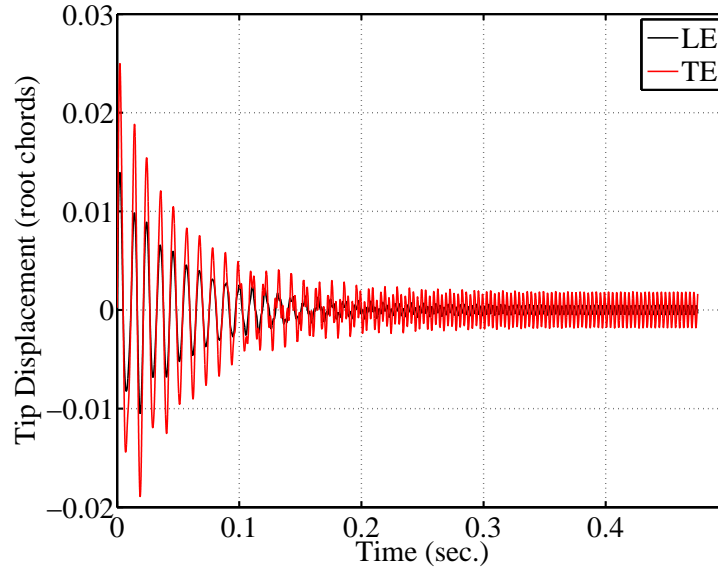


Figure 6.54: NEXST-2 wing tip time histories in the high frequency LCO region at Mach 0.75, $q = 129.1$ kPa.

in Figure 6.7.

Responses in the high frequency LCO region are much smaller and higher frequency than they are in other regions. In Figure 6.54, we can see that the initial velocity imparted to the first mode quickly damps out. This is followed by a brief period of transition before the frequency of the response triples, and the high frequency LCO begins. The frequencies of the different responses can be observed in the waterfall SRS in Figure 6.55. Figure 6.56 shows that the LCO response is mainly composed of vibrations in the fourth mode of the wing. As shown in Figure 6.5, the fourth mode has a significant amount of gain on the inboard leading edge. Animations of the aerodynamic power (included as supplementary material) show that the peak power transfer occurs at the inboard leading edge.

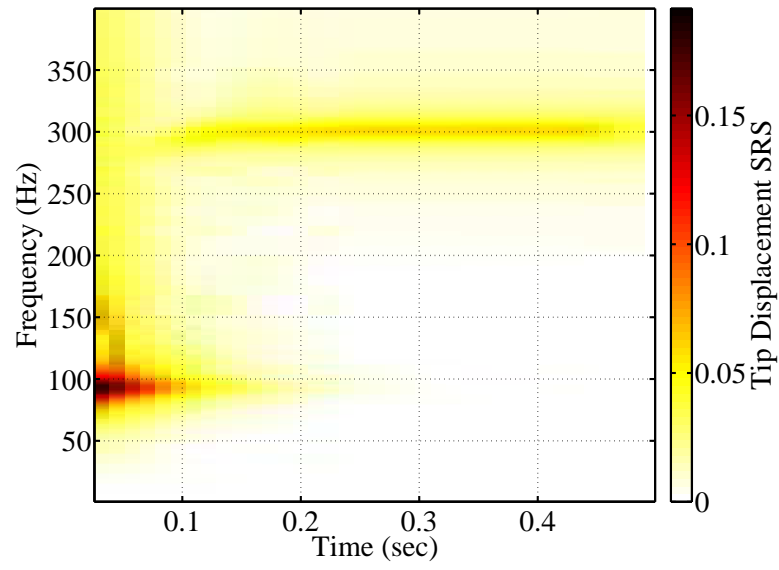


Figure 6.55: Waterfall SRS of the NEXST-2 wing tip trailing edge at Mach 0.75, $q = 129.1$ kPa.

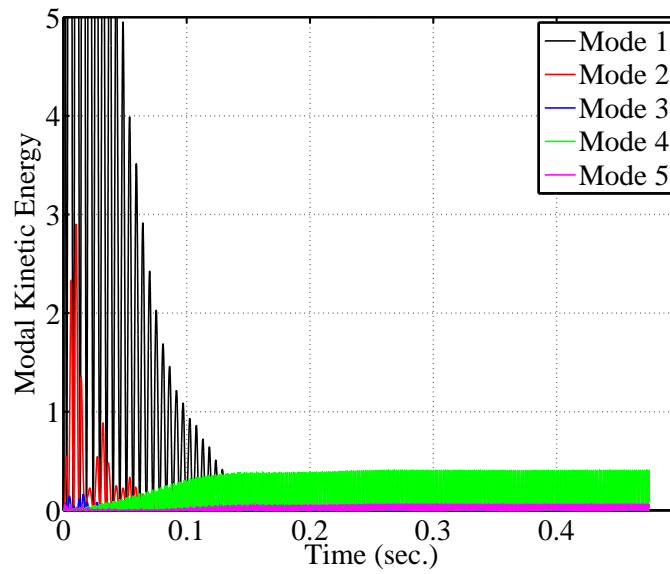


Figure 6.56: Modal kinetic energy of the NEXST-2 wing at Mach 0.75, $q = 129.1$ kPa.

Responses in the high frequency LCO region are so small in magnitude that they do not persist at higher dynamic pressures, where the dominant response is subsonic bending-torsion flutter in the first, second, and third modes. In fact, it is difficult to believe that this response would be observable at all in a wind tunnel where even a small amount of turbulence or structural damping would disrupt the LCO.

6.2.2.5 Damped Responses

At Mach numbers above and below the transonic dip region for lower dynamic pressures, responses damp out without the presence of any aeroelastic instability. By examining results at these conditions, a better understanding of the responses in the transonic dip and bending-torsion flutter regions can be gained. In this section two sets of damped results are briefly discussed. The first condition is at Mach 0.85 with a dynamic pressure of 107.8 kPa. The second condition is at Mach 1.06 with a dynamic pressure of 103.2 kPa. The points are labeled *H* and *I* in Figure 6.7.

Wing tip time histories for Mach 0.85 with a dynamic pressure of 107.8 kPa are shown in Figure 6.57. The response is in the first mode and damps out quickly. This condition is at the same Mach number as point *F* in Figure 6.7 which was discussed in Section 6.2.2.2. For that condition, the higher dynamic pressure resulted in deformation such that the supersonic region extended aft to a position that resulted in transonic dip type responses. In the present case, the dynamic pressure is not large enough to produce the deformation required to move the

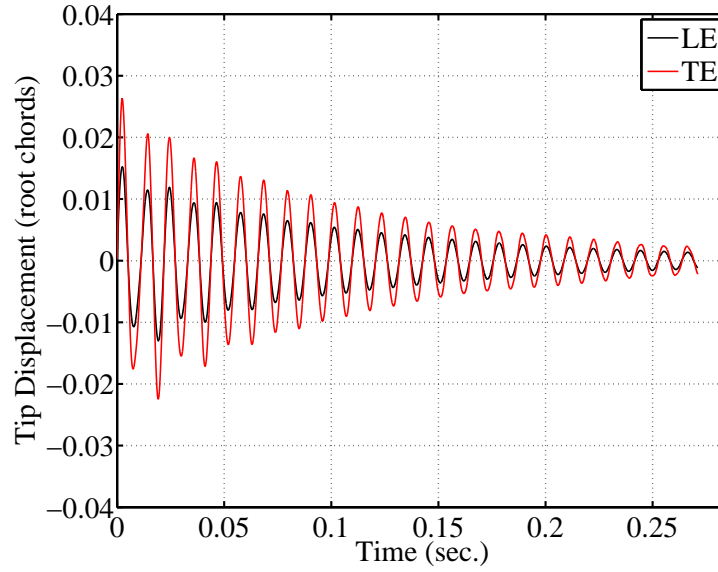


Figure 6.57: NEXST-2 wing tip time histories of a damped response at Mach 0.85, $q = 107.8$ kPa.

shock aft to a position that will result in an unstable response. This can be seen clearly in Figure 6.58 which shows the Mach distribution on the wing during one of the initial cycles of the simulation.

Results from a second damped response flow condition (Mach 1.06, $q = 103.2$ kPa) are shown in Figures 6.59 and 6.60. This condition is at a Mach number just above the transonic dip region, and the response damps out instead of increasing in amplitude.

The Mach distribution in Figure 6.59 shows that the entire wing remains in supersonic flow at every point during the cycle. When compared with the Mach distribution at Mach 1.02 and lower dynamic pressure shown previously in Figure 6.22, the differences are easily observed. At the lower Mach number, the shock is

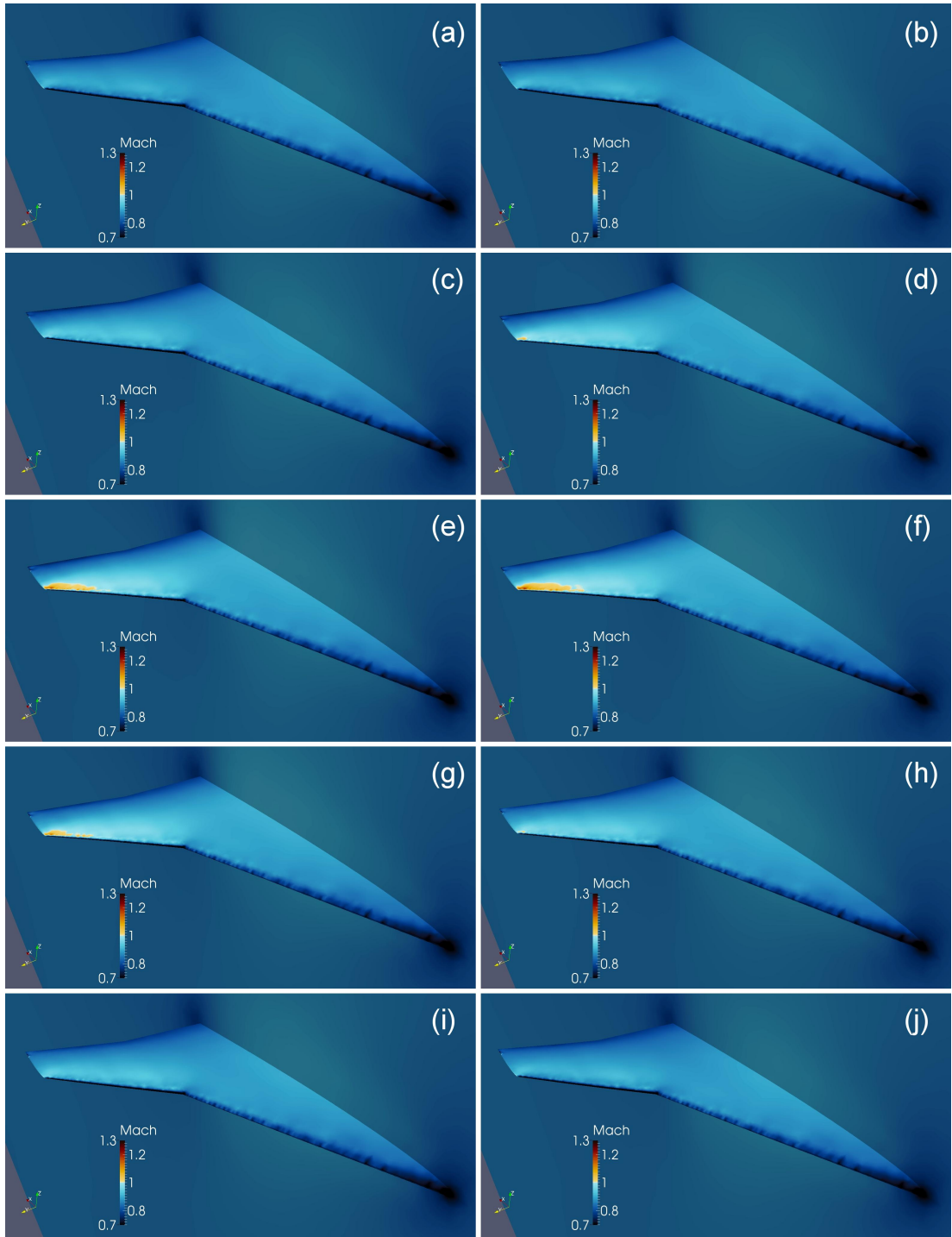


Figure 6.58: Mach distribution on the NEXST-2 wing during the second cycle of the simulation at Mach 0.85, $q = 107.8$ kPa.

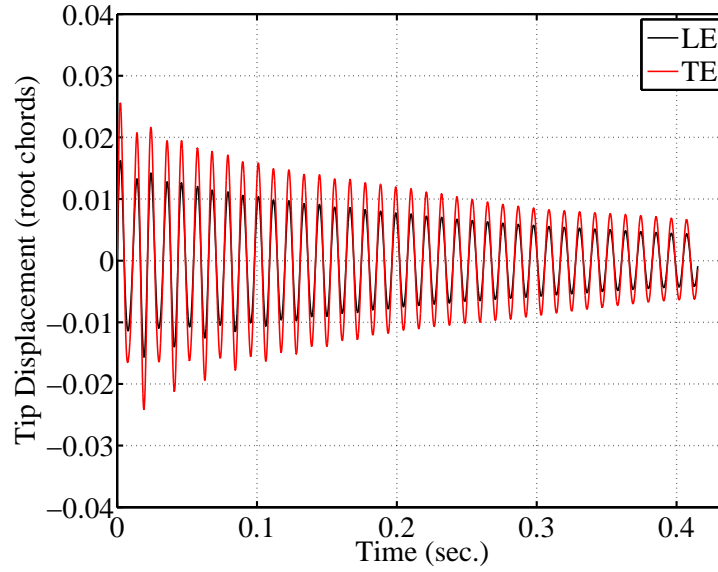


Figure 6.59: NEXST-2 wing tip time histories of a damped response at Mach 1.06, $q = 103.2$ kPa.

able to move forward on the wing when deformation occurs, resulting in the first mode LCO typical of the transonic dip. When the shock is firmly attached to the wing trailing edge, this type of response does not occur.

6.3 Generic Cranked High Speed Wing (HC-Wing)

A generic cranked high speed wing (the *HC-Wing*) has been developed and studied using the methods discussed in Chapters 2–4. The present work is an extension of the work of Bendiksen who previously studied the outboard section of the wing (identified as the *H-Wing*) using the Direct Eulerian-Lagrangian method [56, 126]. He observed angle of attack and Mach number dependent transonic LCO in the second bending mode of the wing. Additionally, and somewhat counterintuitively,

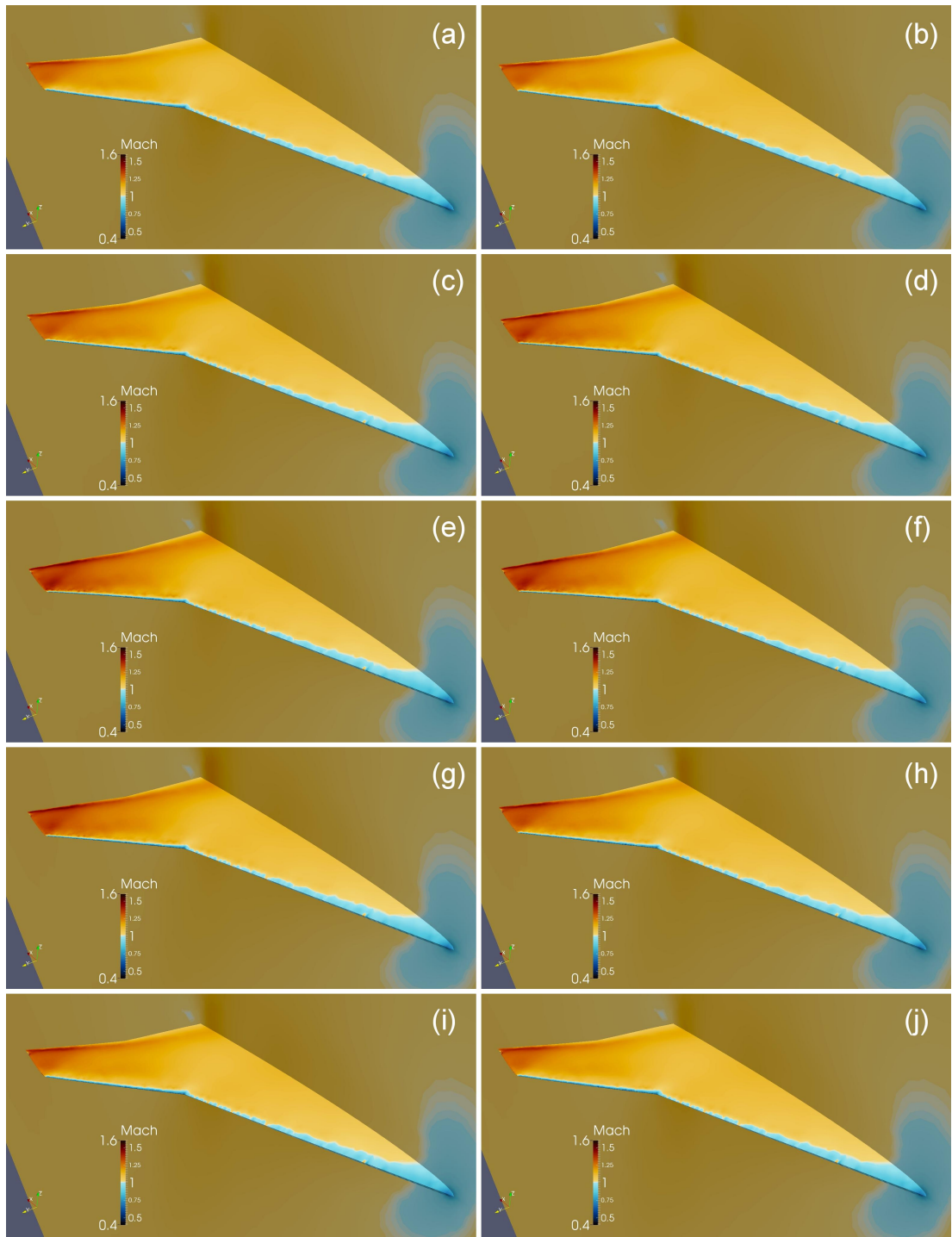


Figure 6.60: Mach distribution on the NEXST-2 wing during the second cycle of the simulation at Mach 1.06, $q = 103.2$ kPa.

the LCO amplitude increased as the wing was made stiffer.

In the present research, the wing was studied both with and without the inboard sections and, similar to Reference 126, showed angle of attack and Mach number dependent transonic limit cycle flutter in the second bending mode of the wing. The addition of a rigid inboard section shifted the onset of LCO to higher angles of attack and Mach numbers due to aerodynamic interaction between the inboard the outboard sections. When compliance was added to the inboard section, no instability was found. The softening of just a small portion of the inboard section stabilized the response at all conditions tested.

6.3.1 H-Wing Aeroelastic Model

To establish a baseline for comparisons of the cranked HC-Wing results, aeroelastic models of the H-Wing were built based on the work of Bendiksen [56, 126]. The planform and aerodynamic mesh of the present implementation of the H-Wing are shown in Figure 6.61. Details of the aerodynamic mesh and a comparison to the mesh used in Reference 126 are provided in Table 6.3. The aerodynamic mesh of the present model is not as dense as the one used in Reference 126 away from the wing, but the two meshes have a similar density on the wing. The current mesh is seen as an improvement over the previous model, as the mesh from Reference 126 was not generated for the H-Wing, but was mapped from another wing with a different planform.

Two structural models of the H-Wing were created using two different constant thickness aluminum plates in the same shape as the wing. The thick model had a plate thickness to root chord ratio, h/c_{root} , of 0.0050 meters, and the thin model

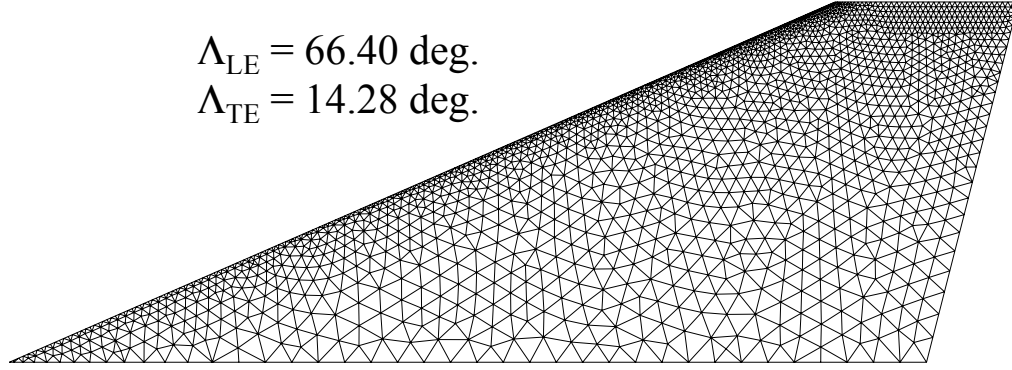


Figure 6.61: Planform and fluid mesh of the H-Wing.

Table 6.3: Mesh details for the H-Wing model.

	Present	Ref. 126
Nodes	29,080	63,700
Faces	302,150	669,202
Cells	148,110	328,652
Nodes on Wing	7,658	9,263

had an h/c_{root} of 0.0045 meters. Ninety-six triangular finite elements were used to discretize the wing (Figure 6.62). The mode shapes of the H-Wing model are shown in Figure 6.63. Details of the natural frequencies of the wing are shown in Table 6.4.

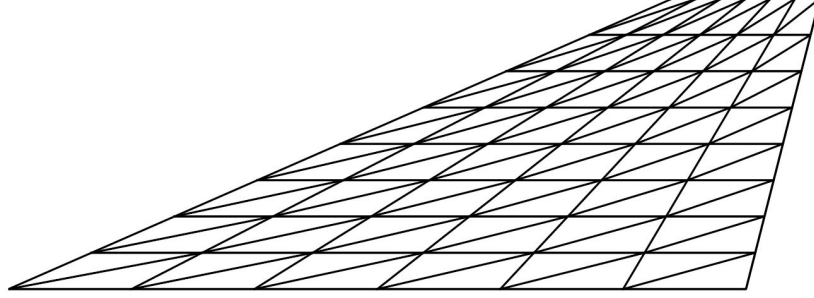


Figure 6.62: The structural finite element discretization of the H-Wing.

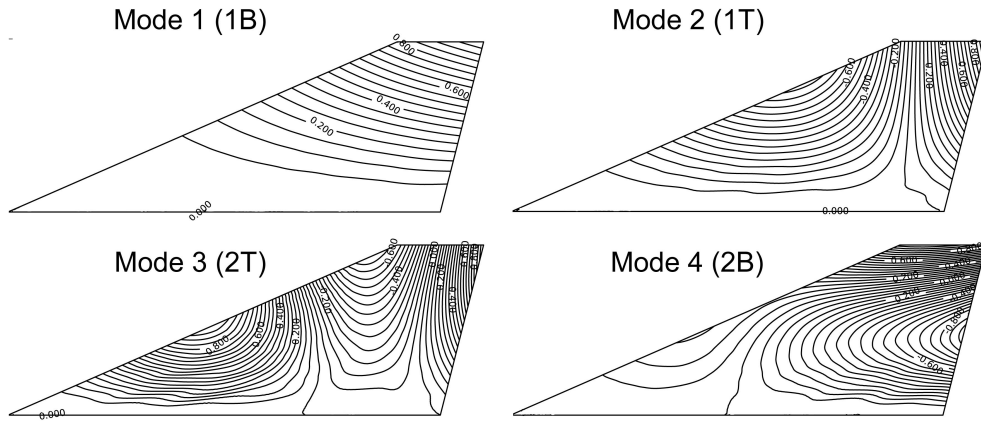


Figure 6.63: The first four normal modes of the H-Wing.

6.3.2 H-Wing Aeroelastic Results

Flutter simulations for the H-Wing were carried out at a variety of conditions for both the thick and thin structural models. A Mach sweep was performed at 7.5 degrees root angle of attack (α), and an angle of attack sweep was performed at

Table 6.4: Natural frequencies of the H-Wing structural finite element model.

Mode	Description	$h/c_{root} = 0.0050$ (Thick)	$h/c_{root} = 0.0045$ (Thin)
1	1st bending	31.4 Hz	28.2 Hz
2	1st torsion	77.5 Hz	69.7 Hz
3	2nd torsion	143.1 Hz	128.8 Hz
4	2nd bending	177.9 Hz	160.1 Hz

Mach 0.99. All simulations were started from a converged steady solution with sea level air density and a small initial velocity in the fourth mode of the wing. This mode was identified as the primary LCO contributor in Reference 126.

Much like the results from Reference 126, the present set of computational solutions revealed LCO that was dependent on both angle of attack and Mach number. The frequency of the response was near the fourth mode of the wing. LCO amplitudes of the thick model were larger than they were for the thin model. At some conditions (usually higher Mach numbers and angles of attack) a “buzz-like” instability was observed. This response was also noted in Reference 126. The LCO amplitudes do not match the previously reported ones exactly, but the differences are within reason considering the different aerodynamic meshes used in the analyses. The LCO amplitude at a range of Mach numbers for $\alpha = 7.5$ degrees is shown in Figure 6.64. The LCO amplitude versus root angle of attack for Mach 0.99 is shown in Figure 6.65. In both plots, the lighter symbols and dashed lines indicate that the buzz-like instability is affecting the results at that condition. The LCO amplitude at these points is a best estimate, as the participation of high frequency modes results in limit cycle amplitudes that do not always remain

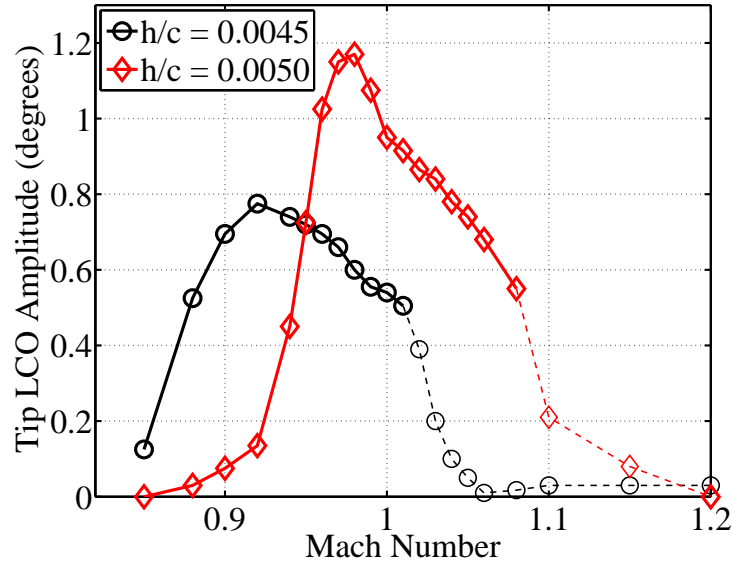


Figure 6.64: LCO amplitude of the tip angle of attack vs. Mach number for the H-Wing at $\alpha = 7.5$ degrees.

constant. Results from the H-Wing will be used as a baseline for comparisons with the HC-Wing results.

Figures 6.66–6.68 show the response of the thick H-Wing model at Mach 0.99, $\alpha = 8.0$ degrees. Wing tip angle of attack time histories in Figure 6.66 show the response growing to LCO. Modal kinetic energy time histories (Figure 6.67) show that response is primarily in the third and fourth modes of the wing (second torsion and second bending). SRS of the tip angle of attack shows that the response is at 185 Hz, which is slightly higher than the frequency of the fourth mode of the wing (Figure 6.68).

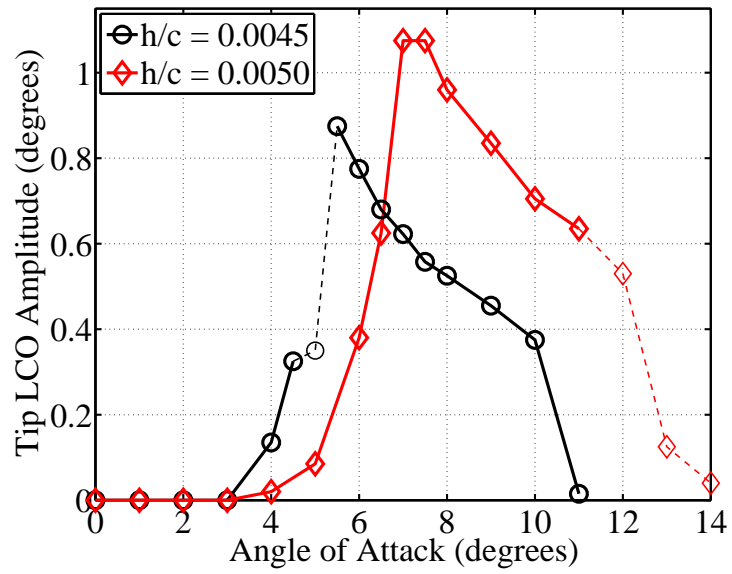


Figure 6.65: LCO amplitude of the tip angle of attack vs. root angle of attack for the H-Wing at Mach 0.99.

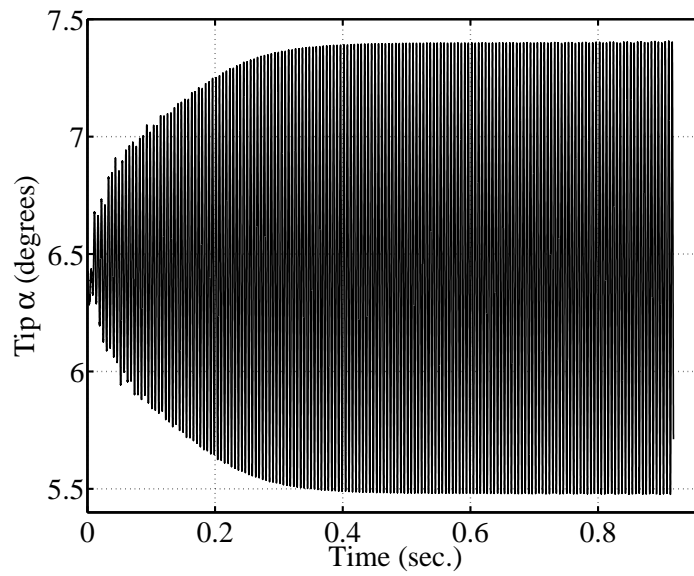


Figure 6.66: Wing tip time histories of the thick H-Wing at Mach 0.99, $\alpha = 8.0$ degrees.

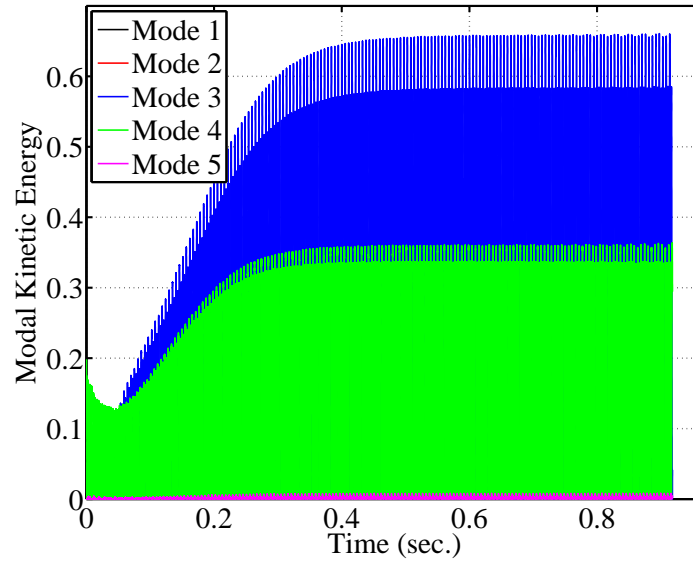


Figure 6.67: Modal kinetic energy of the thick H-Wing at Mach 0.99, $\alpha = 8.0$ degrees.

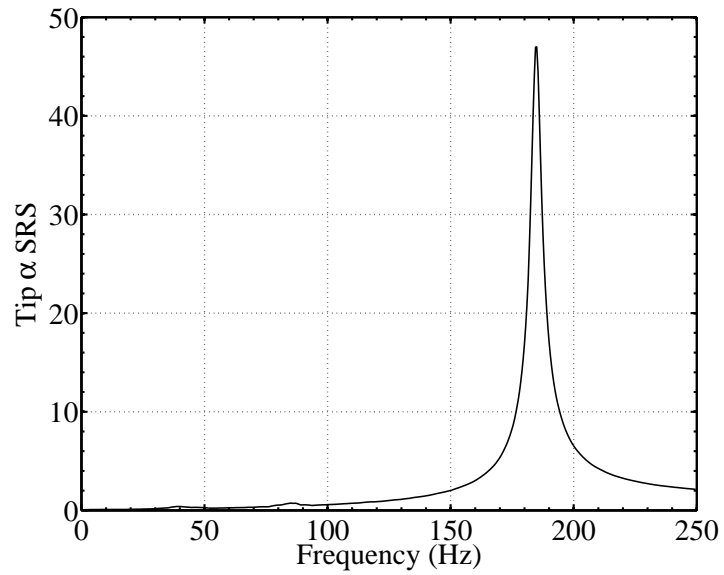


Figure 6.68: SRS of the thick H-Wing at Mach 0.99, $\alpha = 8.0$ degrees.

6.3.3 HC-Wing Aeroelastic Model

The HC-Wing represents a generic high speed wing with high leading edge sweep. The planform, shown in Figure 6.69, is composed of the H-Wing from Section 6.3.1 attached to an inboard section with 74 degree leading edge sweep. The span of the inboard section is half of the H-Wing root chord.

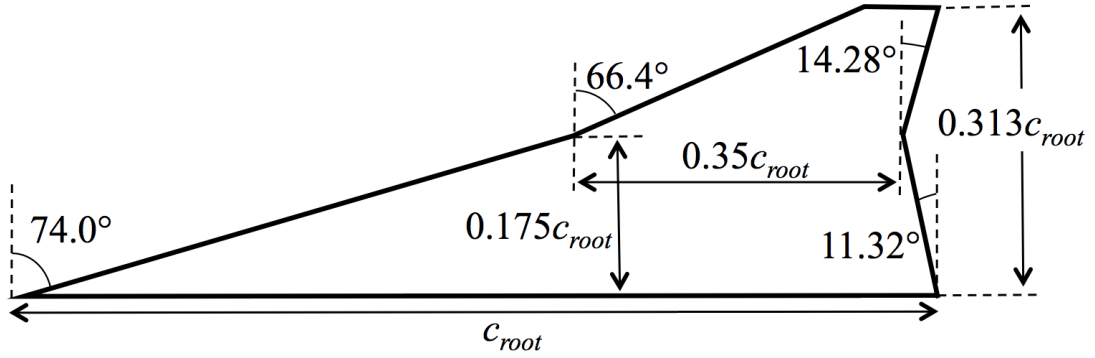


Figure 6.69: Planform of the HC-Wing model.

The fluid mesh for the HC-Wing is shown in Figure 6.70. This mesh was generated using the tools described in Chapter 2. Details of the mesh are provided in Table 6.5.

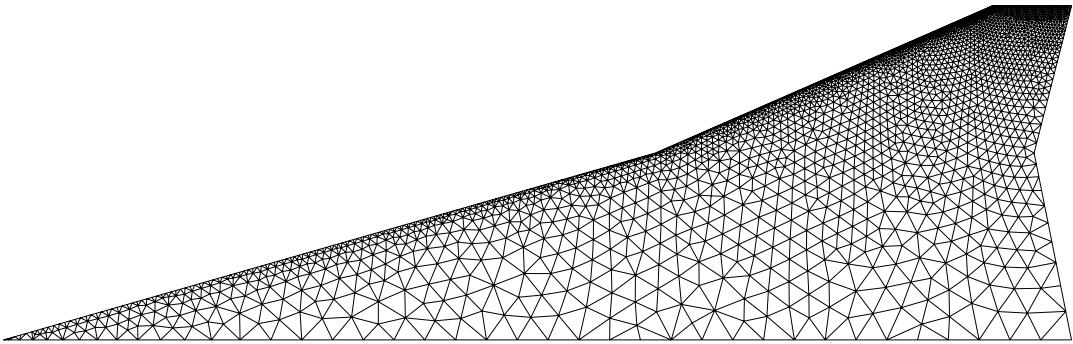


Figure 6.70: Fluid mesh of the HC-Wing model.

Table 6.5: Mesh details for the HC-Wing model.

Nodes	35,103
Faces	363,415
Cells	177,972
Nodes on Wing	5,549

Three different structural models were created for the HC-Wing. The first two models were built by attaching the H-Wing models (thick and thin) from Section 6.3.1 to a rigid inboard section. The structural finite element model, mode shapes, and natural frequencies of the models with the rigid inboard section are unchanged from the H-Wing. For the third model, some flexibility was added to the inboard section, and the thick outer section from the H-Wing was attached. Seventy-five percent of the inboard section was left rigid to prevent high frequency modes from corrupting the simulation. The finite element discretization and thickness distribution of the model with the flexible inboard section is shown in Figure 6.71. The addition of the flexible inboard section changes the mode shapes and natural frequencies of the model. These are shown in Figure 6.72 and Table 6.6. The natural frequencies of the H-Wing are included in Table 6.6 to show the effect of the added flexibility.

6.3.4 HC-Wing Aeroelastic Results

Flutter simulations for the HC-Wing were completed at a range of conditions for all three structural models. A Mach sweep was performed at $\alpha = 7.5$ degrees, and an angle of attack sweep was performed at Mach 0.99. All simulations were

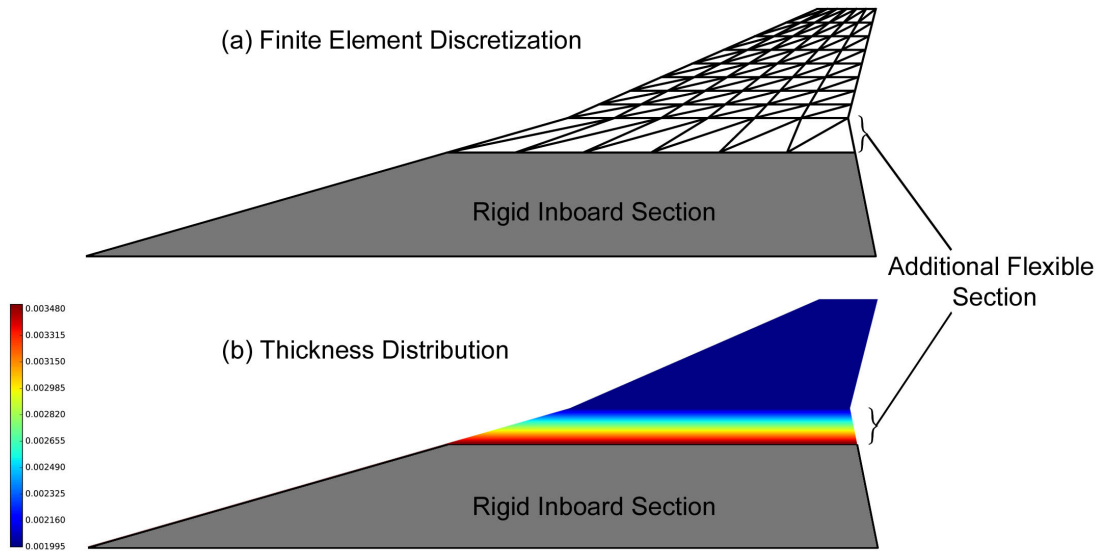


Figure 6.71: Structural model of the HC-Wing with flexible inboard section (a) finite element discretization and (b) thickness distribution.

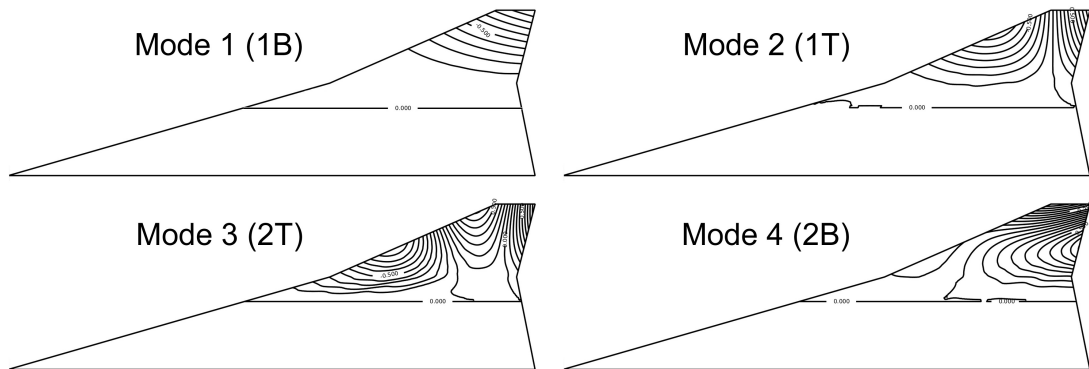


Figure 6.72: The first four normal modes of the HC-Wing with flexible inboard section.

Table 6.6: Natural frequencies of the HC-Wing with flexible inboard section.

Mode Number	Description	Flexible HC-Wing	Thick H-Wing
1	1st bending	25.1 Hz	31.4 Hz
2	1st torsion	60.0 Hz	77.5 Hz
3	2nd torsion	102.8 Hz	143.1 Hz
4	2nd bending	130.7 Hz	177.9 Hz

started from a converged steady solution with sea level air density and a small initial velocity in the fourth mode of the wing.

Similar to the H-Wing results, the HC-Wing with the fully rigid inboard section exhibited angle of attack and Mach number dependent LCO at a frequency near the fourth mode of the wing. Also like the H-Wing, LCO amplitudes were larger for the thick model. Additionally, the same buzz-like instability was present at certain conditions. For the HC-Wing with the flexible inboard section, no instability was found. The softening of just a small portion of the inboard section stabilized the response at all conditions tested. Comparisons of the responses of the thick and thin HC-Wings with rigid inboard sections are shown in Figures 6.73 and 6.74.

Response comparisons between the H-Wing and the HC-Wing are shown in Figures 6.75–6.78. The addition of the highly swept inboard section of the HC-Wing results in the LCO response shifting to higher Mach numbers and angles of attack.

Figures 6.79–6.81 show the response of the thick HC-Wing at Mach 1.01, $\alpha = 7.5$ degrees. Similar to the H-Wing, the response is coupled bending-torsion

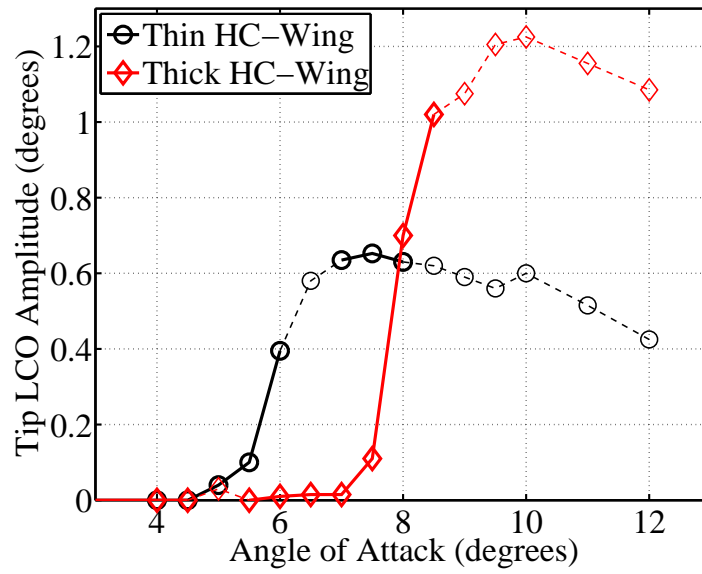


Figure 6.73: LCO amplitude of the tip angle of attack vs. root angle of attack for the HC-Wing at Mach 0.99.

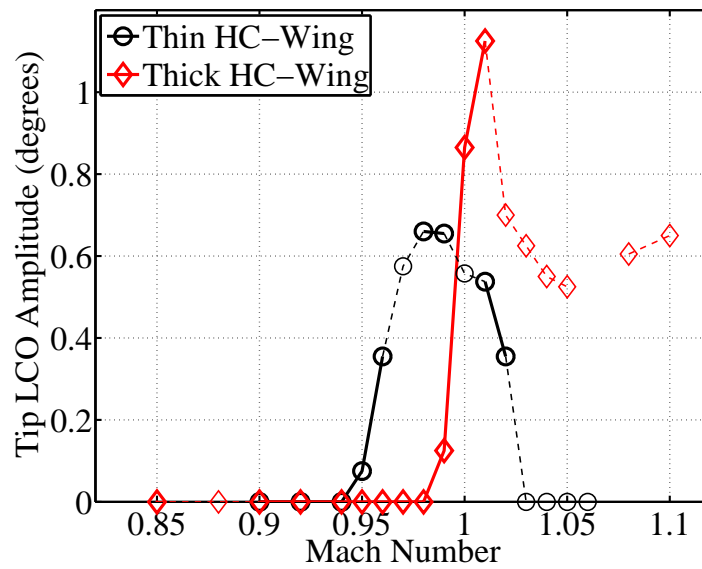


Figure 6.74: LCO amplitude of the tip angle of attack vs. Mach number for the HC-Wing at $\alpha = 7.5$ degrees.

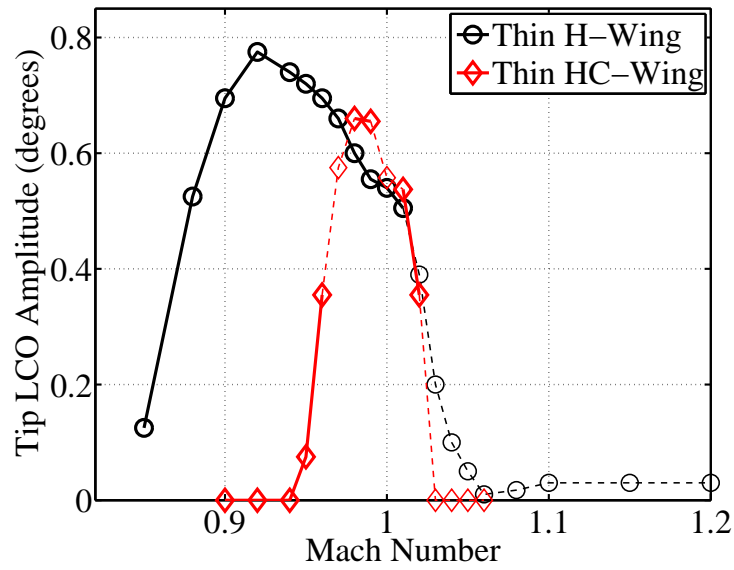


Figure 6.75: LCO amplitude of the tip angle of attack vs. Mach number for the thin H-Wing and thin HC-Wing at $\alpha = 7.5$ degrees.

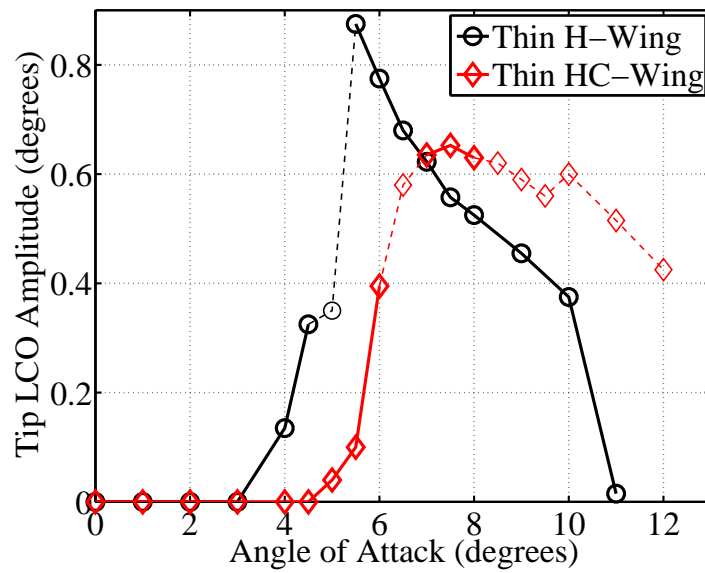


Figure 6.76: LCO amplitude of the tip angle of attack vs. root angle of attack for the thin H-Wing and thin HC-Wing at Mach 0.99.

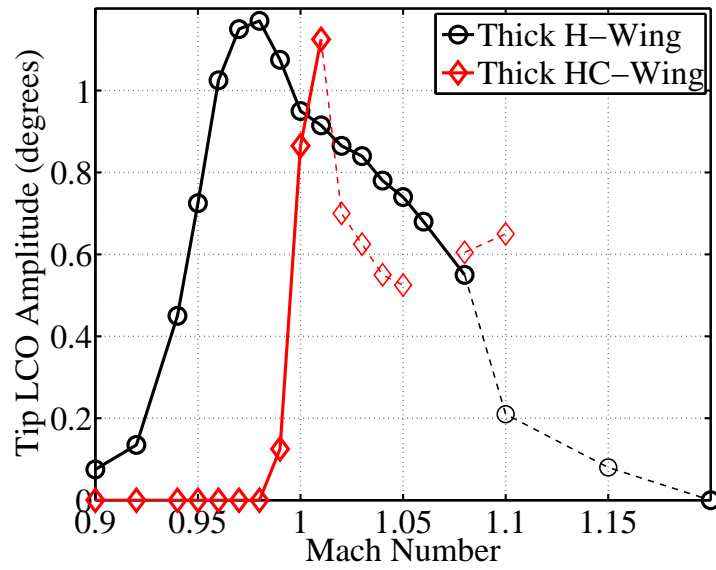


Figure 6.77: LCO amplitude of the tip angle of attack vs. Mach number for the thick H-Wing and thick HC-Wing at $\alpha = 7.5$ degrees.

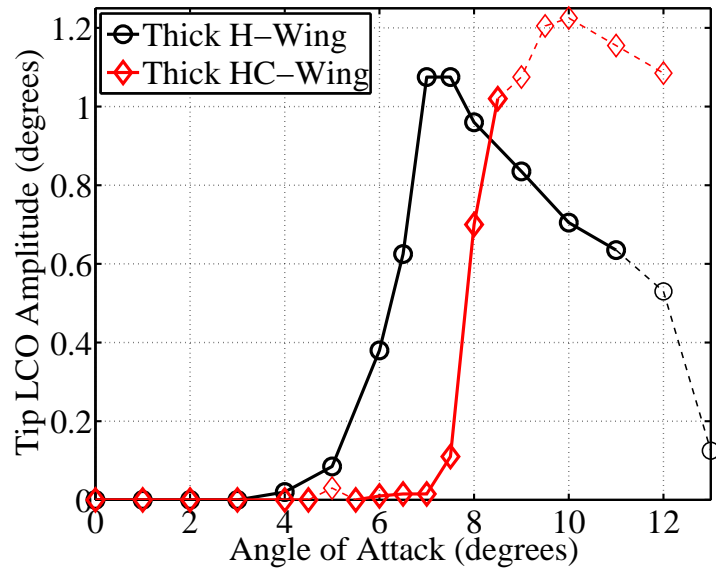


Figure 6.78: LCO amplitude of the tip angle of attack vs. root angle of attack for the thick H-Wing and thick HC-Wing at Mach 0.99.

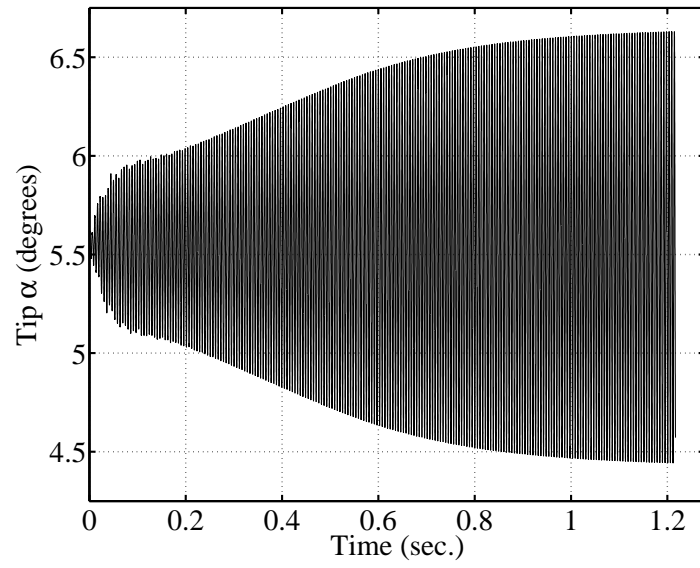


Figure 6.79: Wing tip time histories of the thick HC-Wing at Mach 1.01, $\alpha = 7.5$ degrees.

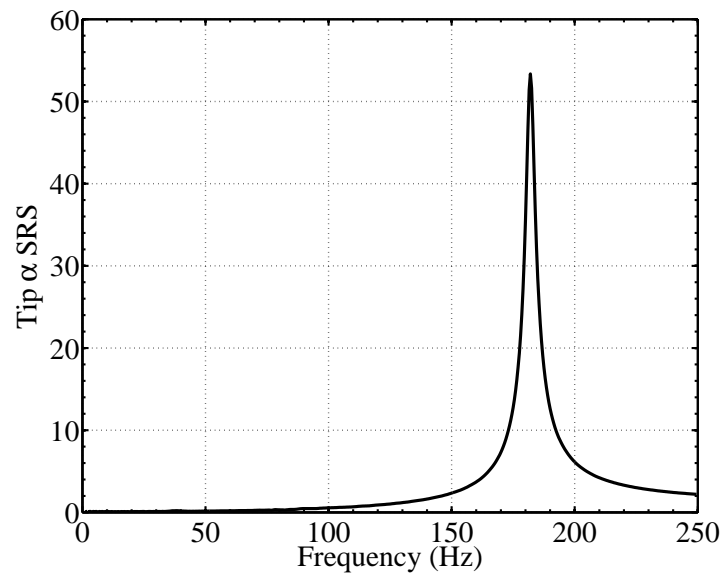


Figure 6.80: SRS of the thick HC-Wing at Mach 1.01, $\alpha = 7.5$ degrees

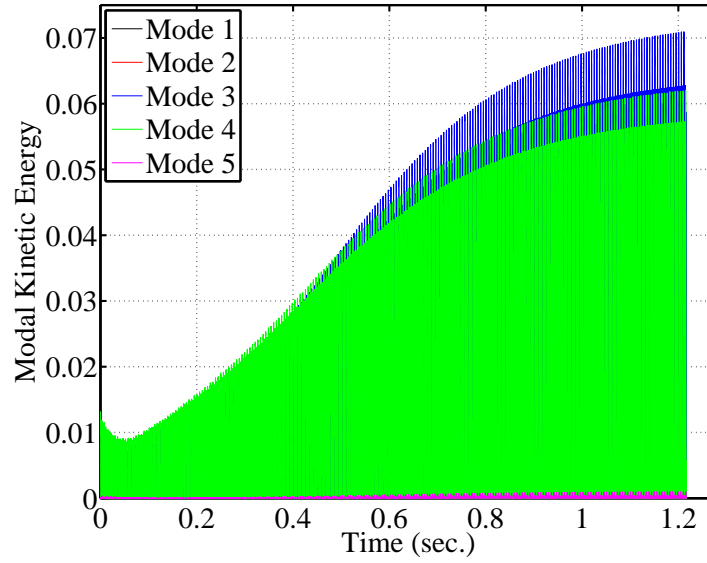


Figure 6.81: Modal kinetic energy of the thick HC-Wing at Mach 1.01, $\alpha = 7.5$ degrees.

LCO in the third (second torsion) and fourth (second bending) modes of the wing. This response is typical, with the exception of the high frequency buzz, of all of the observed LCO. The SRS frequency is unchanged from the H-Wing. Based on the similarities between the responses of the H-Wing and HC-Wing, it appears that the physical mechanism driving the behavior is the same for both wings. The differences are a result of the altered flow field on the HC-Wing caused by the highly swept inboard section.

To better understand the relationship between the LCO response, Mach number, and wing angle of attack, the Mach distribution on both the thin H-Wing and thin HC-Wing upper surface is shown in Figures 6.82–6.85. In these figures, the wing is deformed under a static aeroelastic load.

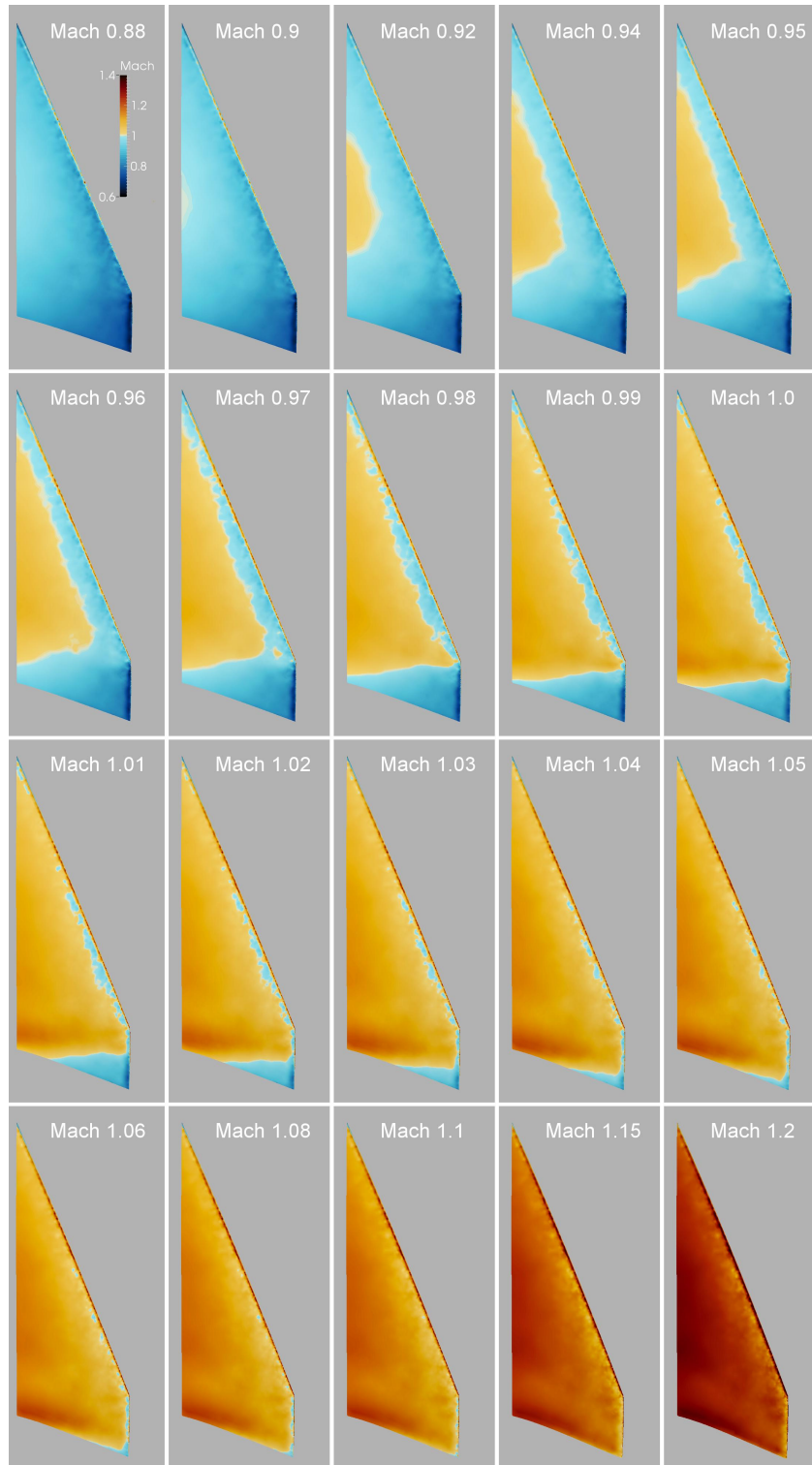


Figure 6.82: Mach distribution on the thin H-Wing upper surface at a range of free stream Mach numbers for $\alpha = 7.5$ degrees.

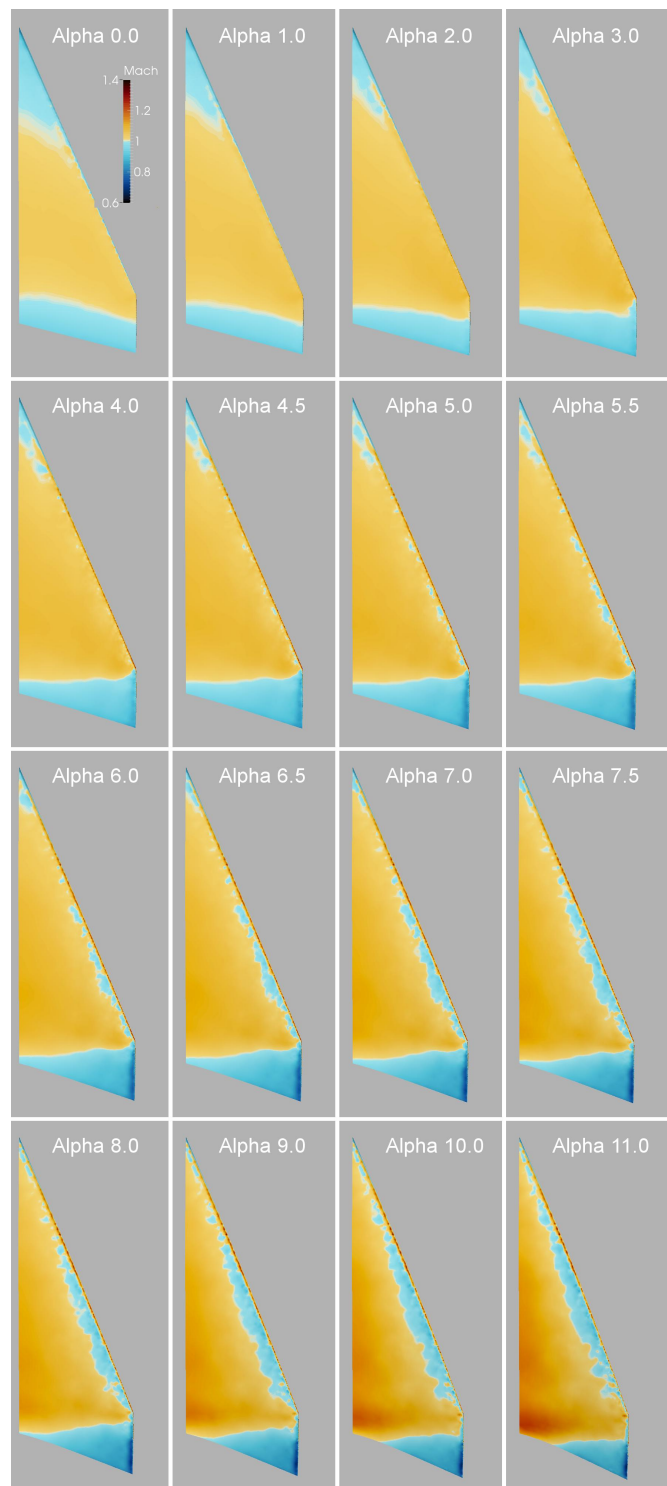


Figure 6.83: Mach distribution on the thin H-Wing upper surface at a range of root angles of attack for a free stream Mach number of 0.99.

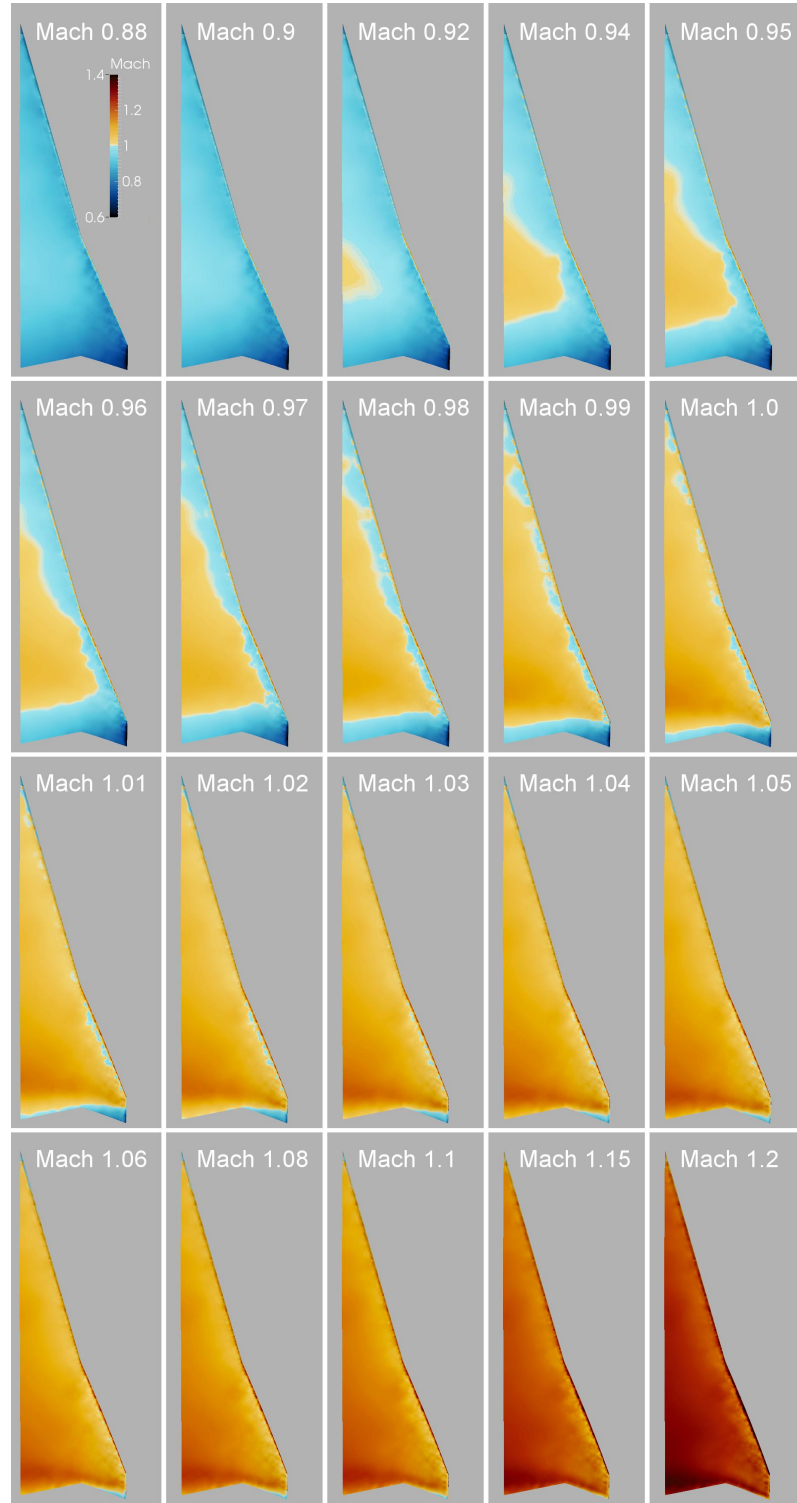


Figure 6.84: Mach distribution on the thin HC-Wing upper surface at a range of free stream Mach numbers for $\alpha = 7.5$ degrees.

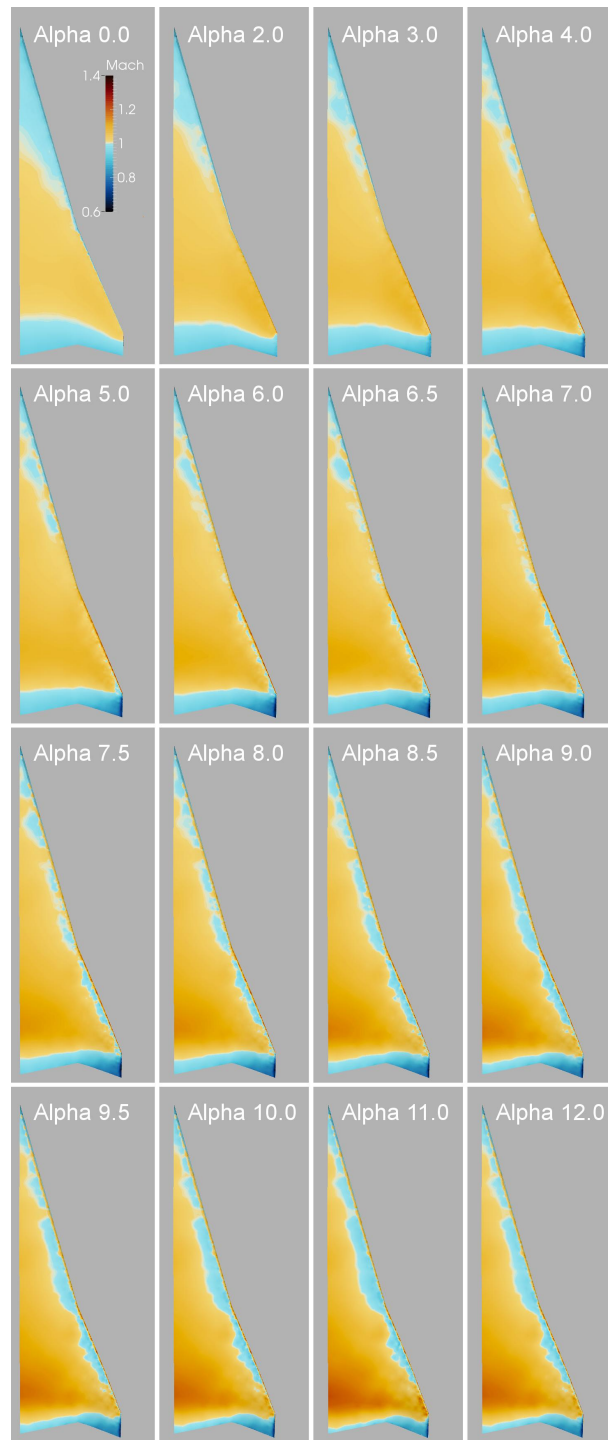


Figure 6.85: Mach distribution on the thin HC-Wing upper surface at a range of root angles of attack for a free stream Mach number of 0.99.

Comparison of Figures 6.82 and 6.84 shows that the presence of the inboard section on the HC-Wing causes the formation of a supersonic flow region on the flexible outboard portion of the wing to occur at higher Mach numbers relative to the H-Wing. For the H-Wing (Figure 6.82), the supersonic region begins forming at Mach 0.92, but for the HC-Wing (Figure 6.84), the supersonic region does not reach a comparable position on the wing until Mach 0.96. This difference in flow conditions on the wing surface results in the LCO of the HC-Wing shifting to higher Mach numbers.

There is a similar explanation for the LCO response of the HC-Wing occurring at higher angles of attack relative to the H-Wing. Figures 6.83 and 6.85 show both wings at a range of root angles of attack for a free stream Mach number of 0.99. The largest LCO response for the H-Wing occurs at 5.5 degrees root angle of attack. The Mach distribution plots show that this corresponds to the condition where the supersonic region on the upper surface of the wing has moved away from the leading edge. As the angle of attack increases, the supersonic region moves further from the leading edge, and the LCO response amplitude decreases. For the HC-Wing, a similar Mach distribution near the wing leading edge does not occur until 7 degrees root angle of attack. This corresponds to the largest LCO amplitude for the HC-Wing.

Figure 6.63 shows that the third and fourth modes of the wing (modes that participate in the LCO) both have a significant amount of gain near the leading edge. It is believed that the LCO behavior observed in the simulations is a result of the transition from subsonic to supersonic flow oscillating in the region of high structural gain near the wing leading edge. The transition only exists in the

correct location on the wing for a narrow range of Mach numbers and angles of attack.

This theory is supported by Figures 6.86 and 6.87, which show the Mach and aerodynamic power distribution on the upper surface of the thick H-Wing during a full period of LCO at Mach 0.99, $\alpha = 8.0$ degrees. At this condition, the thick H-Wing exhibited LCO with a tip amplitude of 0.96 degrees. The deformation is difficult to see in the images but is more visible in the supplementary videos. Figure 6.87 shows that the largest contribution to the energy transfer between the wing and the air occurs at the wing leading edge, especially near the tip. For this reason, the flow conditions along the leading edge are vital to determining the stability of the response.

The Mach distribution on the thick H-Wing upper surface during a period of damped oscillation at Mach 0.99, $\alpha = 3.0$ degrees is shown in Figure 6.88. This is a condition for which the response was stable. Note that the supersonic region covers the entire leading edge of the wing and does not move during the oscillation. This provides further support to the theory that the flow on the leading edge plays an important role in the flutter behavior of the H-Wing and HC-Wing.

Mach distribution plots for the flexible HC-Wing (Figures 6.89 and 6.90) do not show significant differences from the rigid inboard HC-Wing (Figures 6.84 and 6.85). It is believed that the change in modes caused by the additional flexibility (Figure 6.72) results in the aerodynamic forces acting at locations with lower structural gain. The change is enough to result in a net loss of energy during an oscillation cycle instead of a net gain, and the response is stable for all conditions tested.

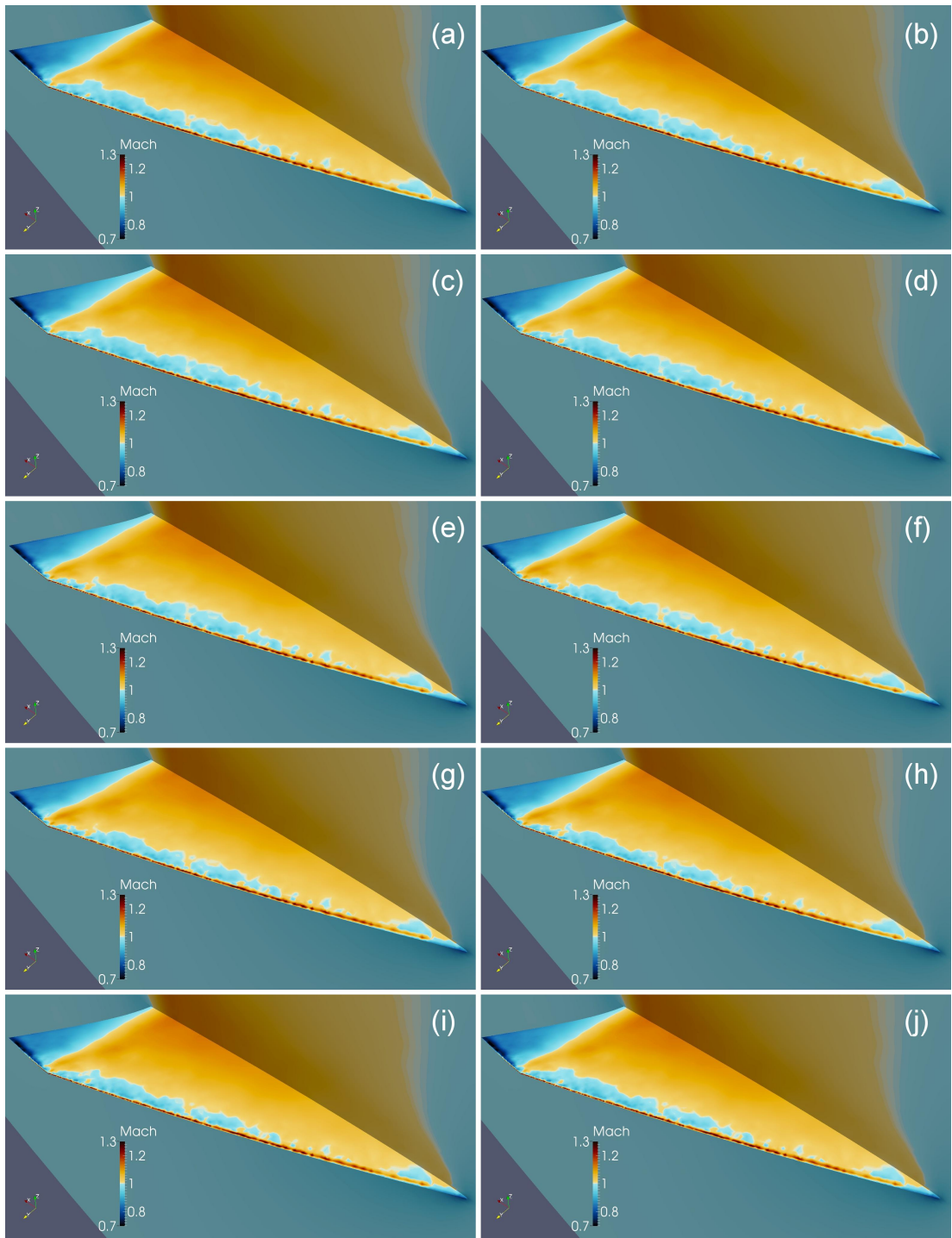


Figure 6.86: Mach distribution on the thick H-Wing during a period of limit cycle flutter at Mach 0.99, $\alpha = 8.0$ degrees.

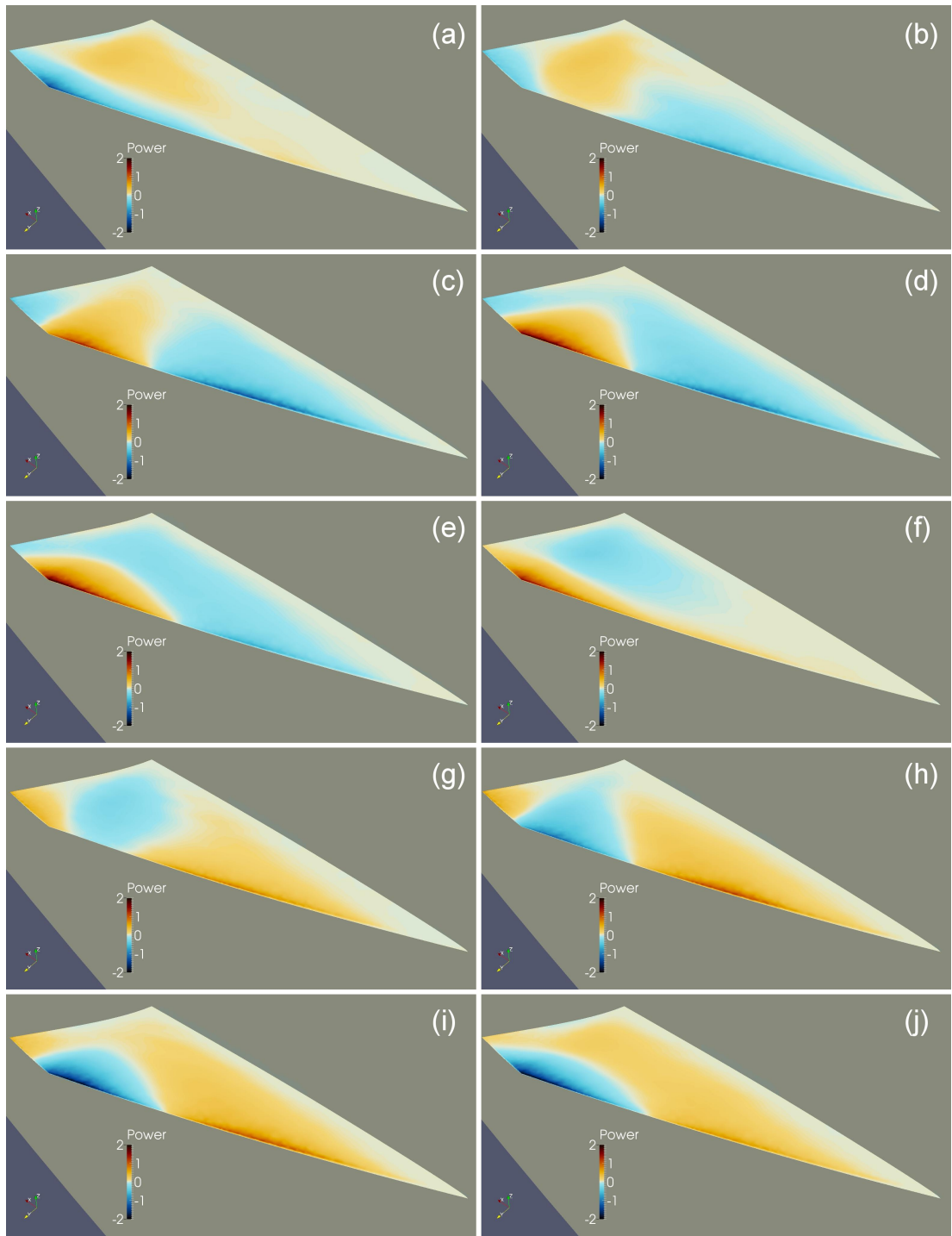


Figure 6.87: Aerodynamic power distribution on the thick H-Wing during a period of limit cycle flutter at Mach 0.99, $\alpha = 8.0$ degrees.

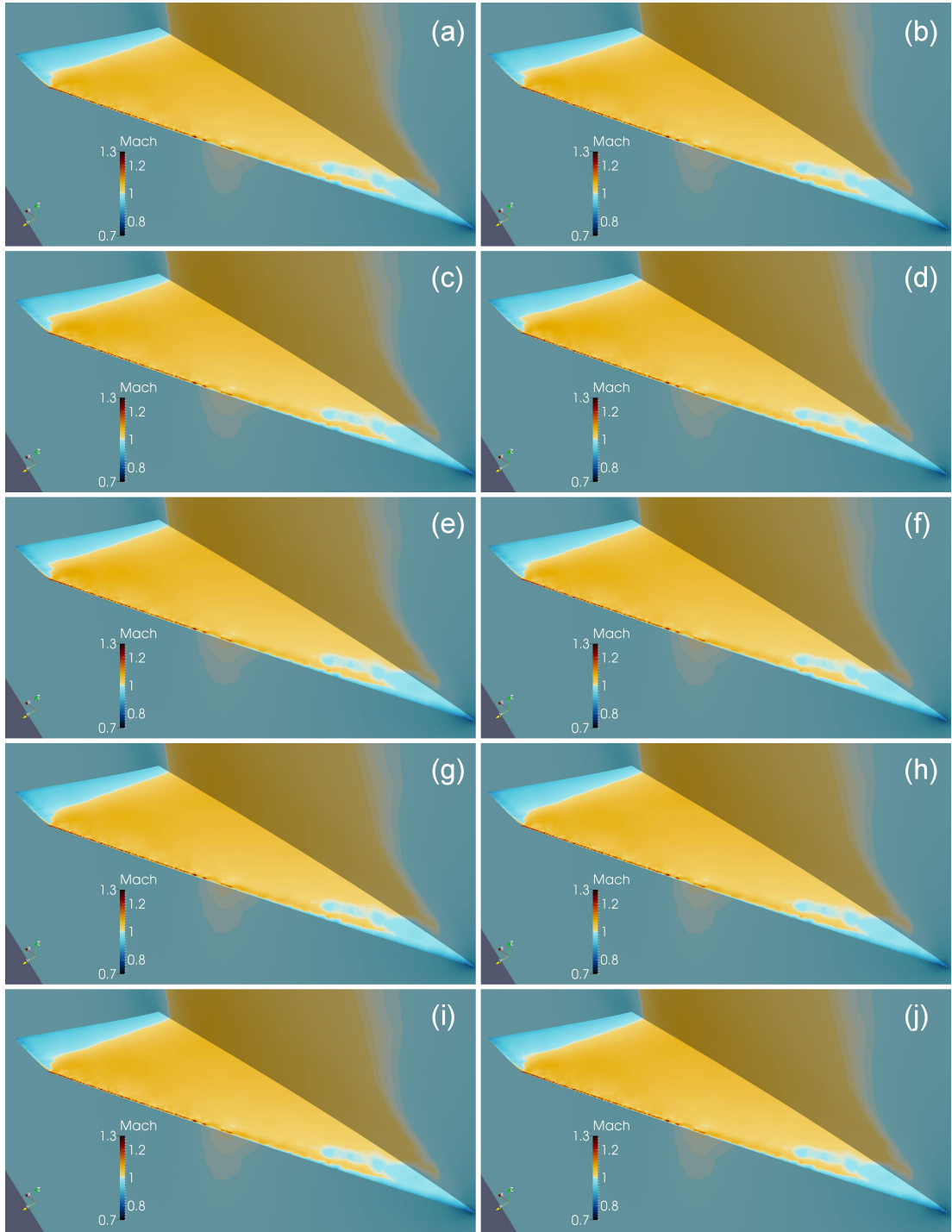


Figure 6.88: Mach distribution on the thick H-Wing during a period of damped oscillation at Mach 0.99, $\alpha = 3.0$ degrees.

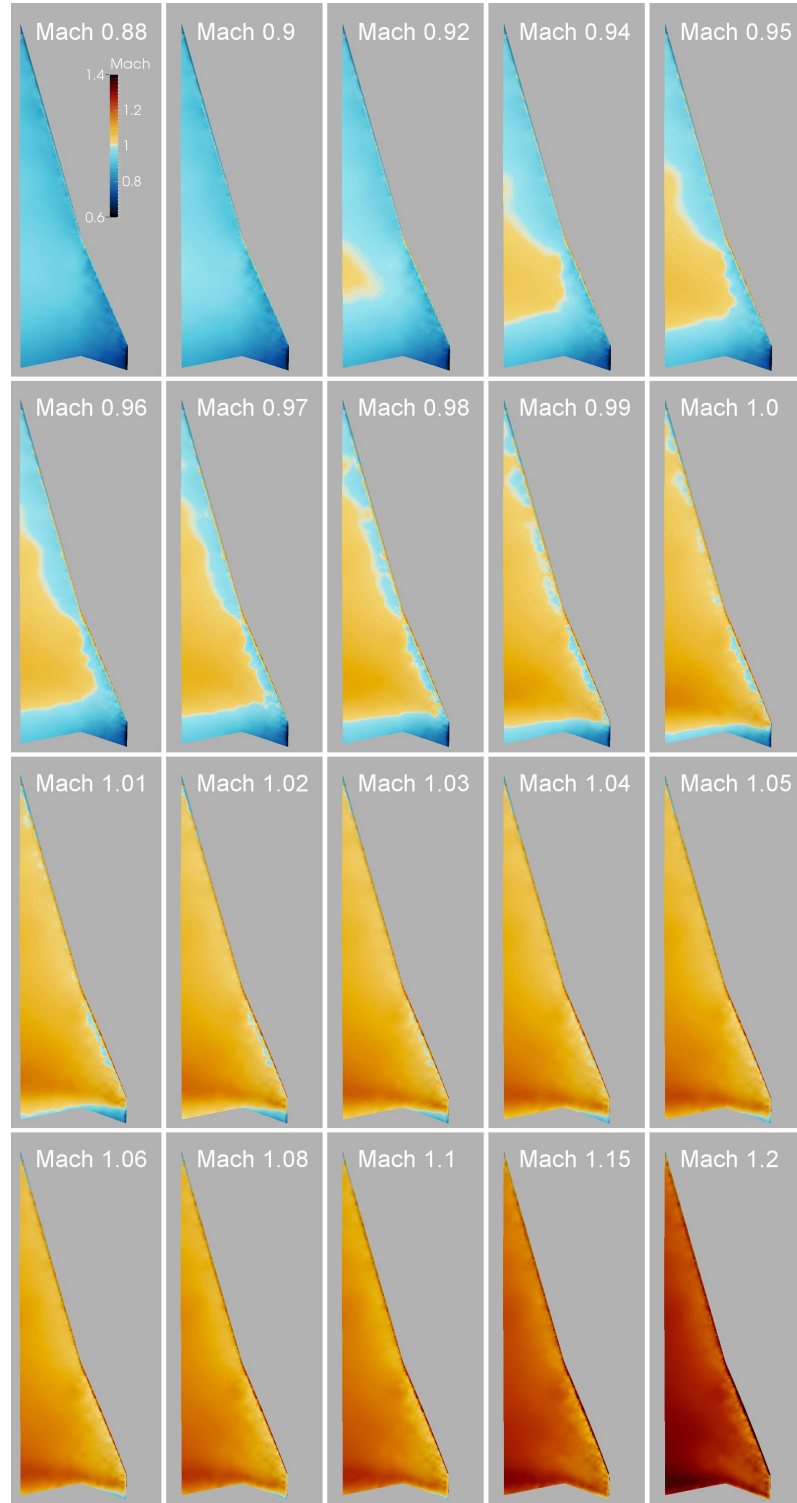


Figure 6.89: Mach distribution on the flexible inboard HC-Wing upper surface over a range of free stream Mach numbers for $\alpha = 7.5$ degrees.

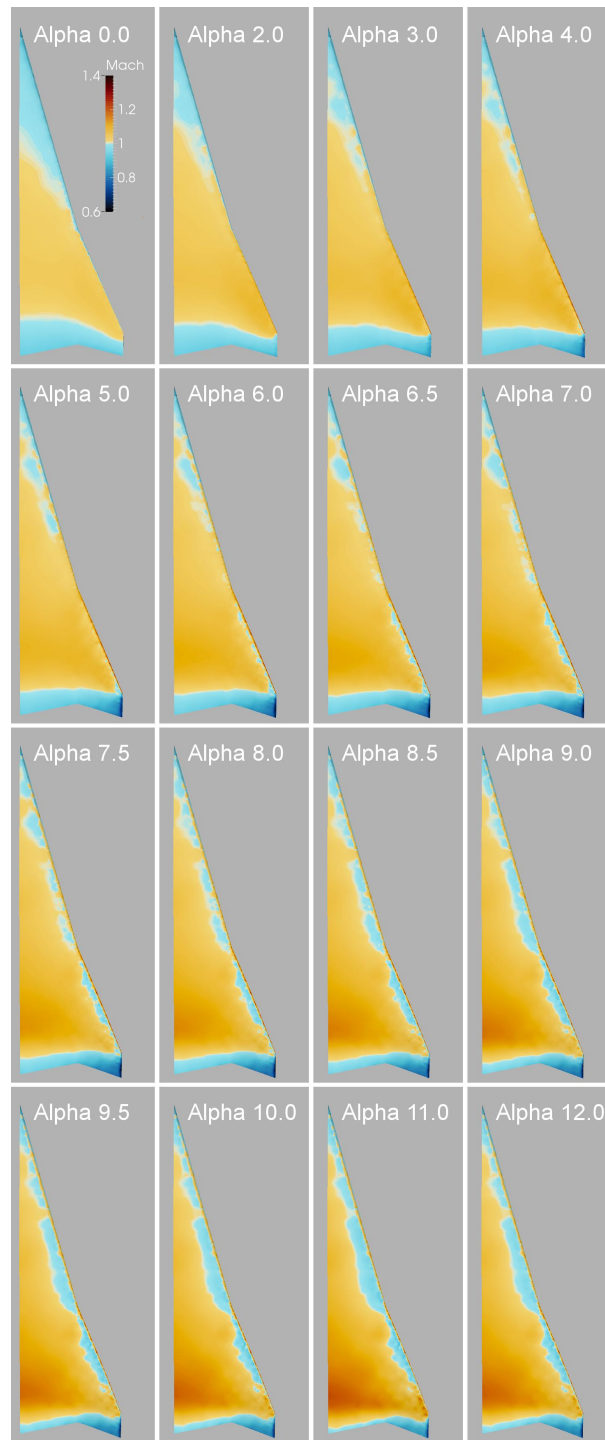


Figure 6.90: Mach distribution on the flexible inboard HC-Wing upper surface for a range of root angles of attack for a free stream Mach number of 0.99.

CHAPTER 7

Conclusions and Recommendations for Future Research

7.1 Introduction

A three-dimensional computational aeroelastic simulation for cranked, highly-swept wing planforms has been developed. The Direct Eulerian-Lagrangian coupling method has been applied to problems of this type for the first time. The simulation has accurately modeled transonic flutter behavior observed in wind tunnel tests. Results from these simulations have revealed previously undocumented aeroelastic behavior.

7.2 Conclusions

The main conclusions that may be drawn from this study are as follows:

1. The Direct Eulerian-Lagrangian coupling methodology has been applied to highly-swept, cranked wings for the first time and has proven to be an accurate way of modeling the energy exchange across the fluid-structure boundary.

2. Simulations of the NEXST-2 wing wind tunnel model revealed multiple types of aeroelastic response at different regions in the Mach versus dynamic pressure plane. A region of stable LCO with single degree-of-freedom response in the first bending mode of the wing was found at a narrow range of Mach numbers (0.94–1.04) and a wide range of dynamic pressures. Additionally, subsonic and supersonic bending-torsion flutter regions were found at higher dynamic pressures and a wide range of Mach numbers. Responses in these regions were at a higher frequency than the transonic dip responses. The simulated responses correlate closely with observed wind tunnel test data.
3. Shock motion dependent on wing deformation was found to play a major role in defining the flutter response of both the NEXST-2 wing and a generic high speed cranked wing. Depending on the flow conditions, it was found that the position of the shock could either stabilize or destabilize the response.
4. For the NEXST-2 wing, at some flow conditions, characteristics of multiple types of flutter response were observed. In some cases, one type of response dominated over other weaker responses, causing them to decay. In other cases, multiple responses existed together for many periods of oscillation.
5. Transonic dip LCO responses for the NEXST-2 wing were shown to be dependent on supersonic flow and shocks moving on the wing surface. The supersonic region must extend aft of 20% of chord for a significant portion of the oscillatory cycle for LCO to occur. Additionally, the shock that terminates the supersonic region must be free to move away from the wing trailing edge for a significant portion of the oscillatory cycle or LCO will not

occur.

6. The transonic dip for the NEXST-2 wing expanded to lower Mach numbers at higher dynamic pressures due to increased wing deformation which tended to move shocks aft on the wing surface. For Mach numbers below the transonic dip, the shock moved aft for enough of the oscillation cycle to bring the wing into the transonic dip. For Mach numbers on the high end of the transonic dip, increased deformation caused the shock to attach to the trailing edge and stabilize the wing against transonic dip responses.
7. For the NEXST-2 wing, the response in the transonic dip showed the ability to transition from small amplitude LCO to rapidly diverging bending-torsion flutter when the shock attached to the wing trailing edge. This transition only occurred for a narrow range of Mach numbers at the high end of the transonic dip region.
8. Shock-boundary layer interaction is believed to shift the transonic dip region to higher Mach numbers because of its tendency to slow the rearward movement of shocks with increasing Mach number.
9. Modal kinetic energy was shown to be a valuable metric for measuring the contribution of individual modes to aeroelastic response.
10. Simulations of a generic high speed wing with and without an inboard region with high leading edge sweep exhibited angle of attack and Mach number dependent LCO at a frequency near the fourth mode of the wing. Two structural models with a rigid inboard section were used in the simulations.

Counterintuitively, LCO amplitudes were found to be largest for the stiffer model.

11. The addition of a rigid highly swept inboard section to the generic high speed wing resulted in LCO response shifting to higher Mach numbers and angles of attack. The shift was found to be due to the inboard section altering the flow field on the outboard section.
12. For the generic high speed cranked wing, the addition of inboard flexibility was found to stabilize the response. It is believed that the change in modes caused by the additional flexibility resulted in the aerodynamic forces acting at locations with lower structural gain. The change was enough to result in a net loss of energy during an oscillation cycle instead of a net gain, and the response was stable for all conditions tested.
13. The aerodynamic meshing, structural modeling, and aeroelastic simulation codes used in previous analyses have been simplified and improved. The process of generating aeroelastic models for use in simulations is more straightforward and easier to complete.
14. The aeroelastic simulation codes have been updated to generate output files that contain deformed wing geometry along with the pressure coefficient, Mach number, pressure, fluid velocity vector, and aerodynamic power at every grid point in the computational domain. The output files for the flutter program can be used to create 3-D animations that have proven useful in the evaluation of simulation results.

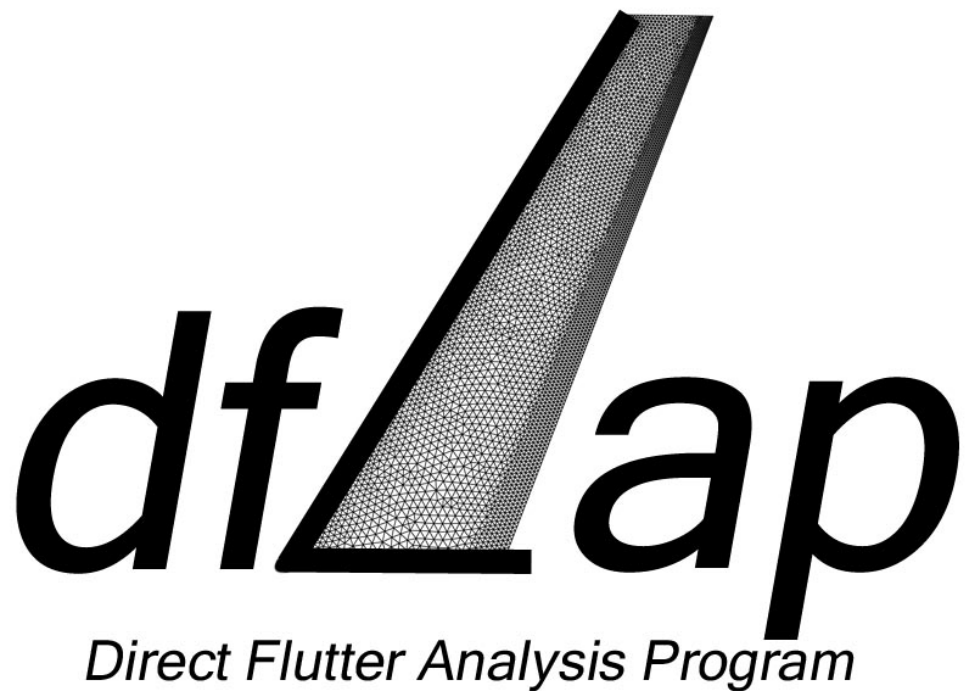
7.3 Recommendations for Future Research

In this work, the Direct Eulerian-Lagrangian coupling method was applied to a new class of problems. For the first time, flutter simulations have been run for wings with high sweep, leading and trailing edge kinks, and complicated mass and thickness distributions. With these improvements, it appears feasible that the method could be applied to more complicated wing geometries and structural models. Suggested improvements include:

1. Incorporate the ability to model more complex geometries including airfoil camber and wing mounted stores/engines.
2. Incorporate the ability to model real-world aircraft structures including spars, ribs, and stressed skin construction. Special attention should be given to accurately characterizing the fluid-structure boundary using either finite element shape functions or residual flexibility vectors with truncated modes [128].
3. Incorporate the ability to model full aircraft with control surfaces and rigid body degrees of freedom.
4. Incorporate the ability to model nonlinear behaviors of moving joints and mechanisms such as those of the B-1 wing [69].

APPENDIX A

dfLap User's Guide



User's Guide

Version 3.0

1.0 Introduction

This document serves as the User's Guide for dfLap, the Direct Flutter Analysis Program. dfLap is a collection of Python [1] functions built using the NumPy [2] and matplotlib [3] packages that work together through a graphical user interface (GUI) to make the process of generating the input files for use in computational aeroelasticity analyses as simple as possible. The program interfaces with the new *kink* versions of the 2-D meshing, steady CFD, static aeroelastic, and flutter programs. dfLap can also be used to generate visualizations of meshes and aeroelastic results. Since dfLap is built in Python, it is compatible with Windows, Mac and Linux operating systems.

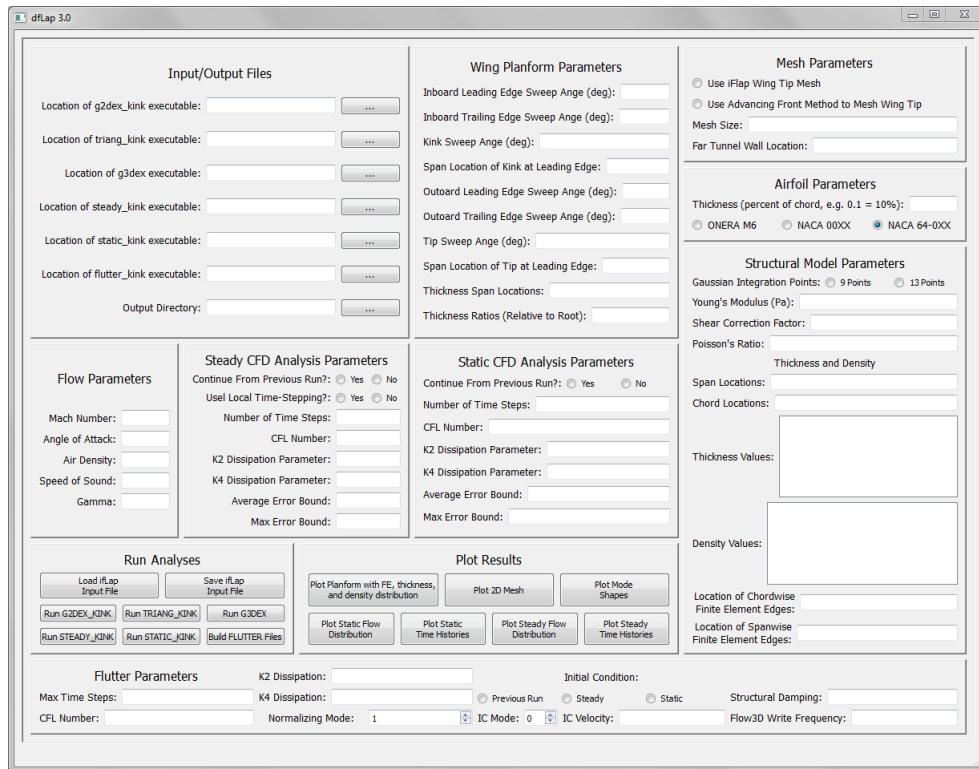


Figure 1: The dfLap interface.

Prior to dfLap the process of generating the input files for a flutter analysis was tedious and error prone. Users were forced to enter the same information in multiple places, making sure that each entry matched. It was also the user's responsibility to keep track of input file requirements and formatting. With dfLap, many of these problems are resolved. By filling in the blanks of the different input modules and clicking the appropriate buttons, input files are generated, analyses are run, and results are presented in a useful, visual format.

Chapter 2 of this user's guide presents the eleven modules in the dfLap interface. Chapter 3 discusses the Python functions used to generate and run the meshing and aeroelastic programs. Chapter 4 contains an example of how dfLap is used to calculate aeroelastic results.

2.0 The dfLap Interface

The dfLap interface is divided into eleven modules. The modules, shown in Figure 2, include: (1) Input/Output File Module, (2) Wing Planform Parameters Module, (3) Mesh Parameters Module, (4) Airfoil Parameters Module, (5) Flow Parameters Module, (6) Steady CFD Parameters Module, (7) Static Aeroelastic Parameters Module, (8) Structural Model Parameters Module, (9) Flutter Parameters Module, (10) Analysis Module, and (11) Plotting Module,

2.1 Input/Output File Module

The Input/Output File Module is used to define the path for the input and output files required by dfLap. Required files include executables for 2-D meshing

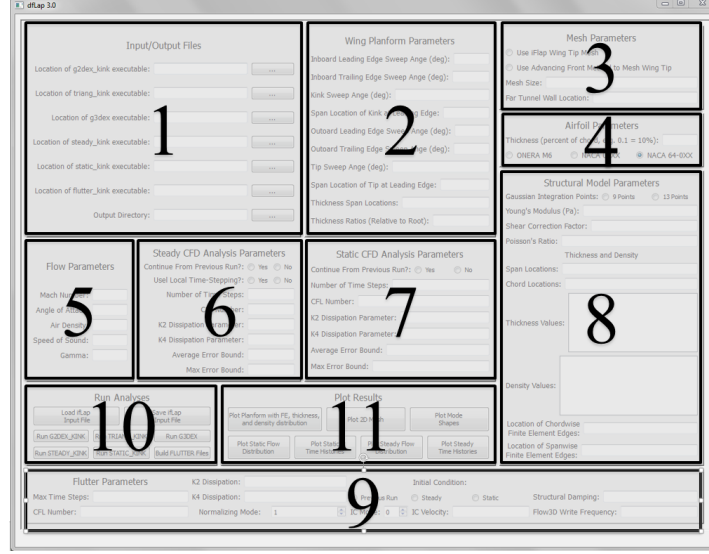


Figure 2: The dfLap modules.

(*g2dex_kink*), removal of similar nodes (*triang_kink*), 3-D meshing (*g3dex*), steady CFD program (*steady_kink*), static aeroelastic program (*static_kink*), flutter program (*flutter_kink*), and the directory for the output files. The Input/Output File Module is shown in Figure 3. The path can be typed into the text boxes manually or can be selected after clicking on the buttons to the right of the text boxes.

2.2 Wing Planform Parameters Module

The Wing Planform Parameters Module is used to input the parameters that define the shape of the wing. The module is shown in Figure 4. The parameters are described in Table 1, and shown in Figure 5. All angles have units of degrees. The length parameters Y_{L1} and Y_{L2} are based on a unit normalized root chord. The thickness parameters T_1 and T_2 define the relative thickness of the airfoil at the kink line and the tip line based on the root chord thickness.

Input/Output Files

Location of g2dex_kink executable:

Location of triang_kink executable:

Location of g3dex executable:

Location of steady_kink executable:

Location of static_kink executable:

Location of flutter_kink executable:

Output Directory:

Figure 3: The Input/Output Module.

Table 1: Wing planform parameters.

Parameter	Description
Λ_{L1}	Inboard leading edge sweep angle
Λ_{T1}	Inboard trailing edge sweep angle
Λ_{K1}	Kink sweep angle
Y_{L1}	Span location of kink at leading edge
Λ_{L2}	Outboard leading edge sweep angle
Λ_{T2}	Outboard trailing edge sweep angle
Λ_{K2}	Tip sweep angle
Y_{L2}	Span location of tip at leading edge
Y_T	Thickness definition span locations
T_{span}	Thickness ratios relative to root

Wing Planform Parameters

Inboard Leading Edge Sweep Angle (deg):

Inboard Trailing Edge Sweep Angle (deg):

Kink Sweep Angle (deg):

Span Location of Kink at Leading Edge:

Outboard Leading Edge Sweep Angle (deg):

Outboard Trailing Edge Sweep Angle (deg):

Tip Sweep Angle (deg):

Span Location of Tip at Leading Edge:

Thickness Span Locations:

Thickness Ratios (Relative to Root):

Figure 4: The Wing Planform Parameters Module.

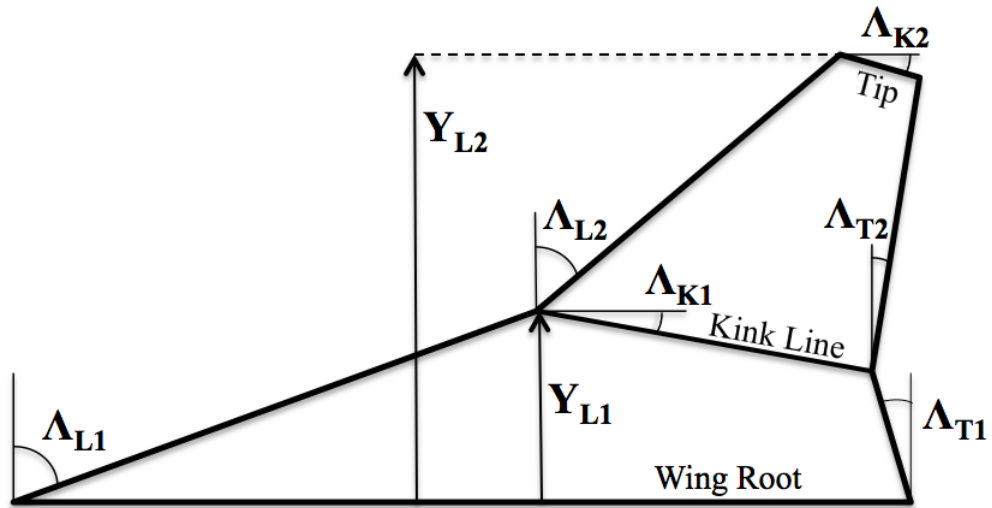


Figure 5: Wing planform parameters.

2.3 Mesh Parameters Module

The Mesh Parameters Module controls the overall mesh density, meshing of the wing tip, and the location of the tunnel far wall. The module is shown in Figure 6.

By selecting one of the radio buttons, the user can choose to generate the 2-D wing tip mesh using a predefined mesh pattern available in dfLap or by using the advancing front method (AFM). The AFM has historically proven problematic for meshing the wing tip. Because of this, it is recommended to use the method provided in dfLap. The dfLap method generates a mesh using a repeating pattern with element size based on the *Mesh Size* parameter.

The *Mesh Size* parameter defines the average size of element edges on the wing root. The mesh size on the wing surface, wing tip, and the area surrounding the wing are also controlled by this parameter. It can be set to any value between 0.016 (dense mesh) and 0.041 (coarse mesh).

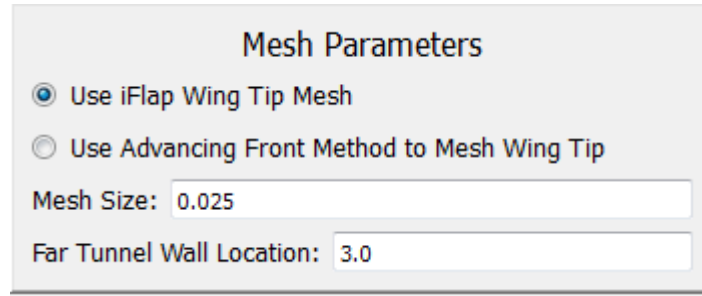


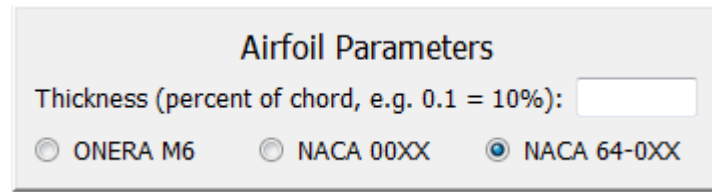
Figure 6: Mesh Parameters Module.

The tunnel far wall location can be adjusted to provide enough space beyond the wing tip in the span direction such that the tunnel wall does not interfere with the aerodynamics near the wing. This parameter should normally be approximately 1.0 larger than Y_{L2} from the Wing Planform Parameters Module.

2.4 Airfoil Parameters Module

The Airfoil Parameters Module controls the thickness and shape of the wing cross section. The module is shown in Figure 7. The thickness ratio of the wing at the

root can be entered in the box. Three airfoil shapes are available: the ONERA M6 airfoil, the NACA 00XX series, and the NACA 64-0XX series of airfoils. The ONERA airfoil has been used previously with the meshing and aeroelasticity programs, so it is included for continuity. The user selects an airfoil shape by clicking one of the radio buttons. The airfoils are shown in Figure 8 with the thickness increased to emphasize the differences.



The image shows a software interface titled "Airfoil Parameters". It features a text input field labeled "Thickness (percent of chord, e.g. 0.1 = 10%):". Below this field are three radio buttons for selecting an airfoil type: "ONERA M6", "NACA 00XX", and "NACA 64-0XX". The "NACA 64-0XX" option is currently selected, indicated by a filled blue circle.

Figure 7: Airfoil Parameters Module.

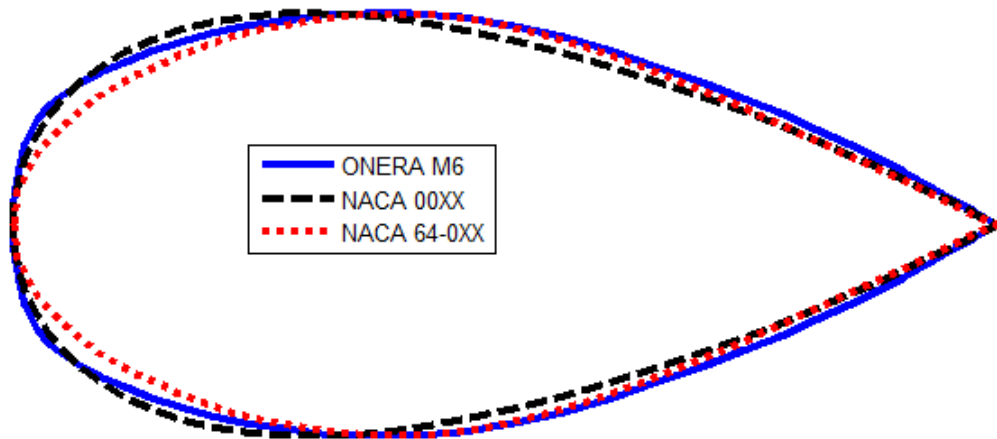
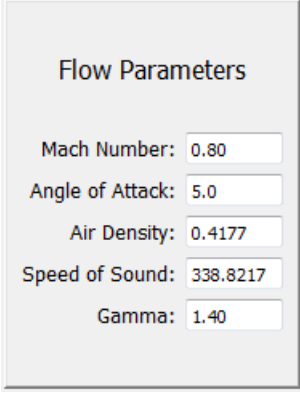


Figure 8: Airfoils available in dfLap.

2.5 Flow Parameters Module

The Flow Parameters Module controls the airflow conditions that will be used in the static CFD, steady aeroelastic, and flutter simulations. This module is shown

in Figure 9. Flow variables that can be set here include Mach number, angle of attack, air density, the speed of sound, and gamma (the ratio of specific heats of the fluid).

A screenshot of a software interface titled "Flow Parameters". It contains five rows of input fields, each with a label and a numerical value. The labels are "Mach Number:", "Angle of Attack:", "Air Density:", "Speed of Sound:", and "Gamma:". The values are 0.80, 5.0, 0.4177, 338.8217, and 1.40 respectively. The interface has a light gray background and a thin black border.

Flow Parameters	
Mach Number:	0.80
Angle of Attack:	5.0
Air Density:	0.4177
Speed of Sound:	338.8217
Gamma:	1.40

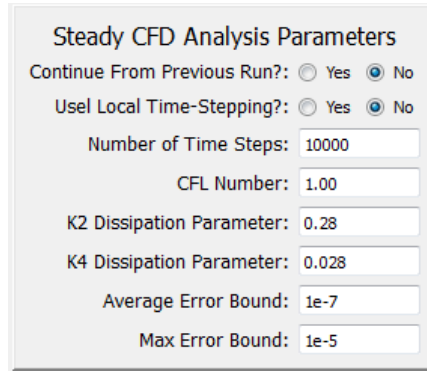
Figure 9: Flow Parameters Module.

2.6 Steady CFD Parameters Module

The Steady CFD Parameters Module controls the inputs for the steady CFD analysis. The module is shown in Figure 10. Variables that can be set here include continuation from a previous run (yes or no), local time-stepping (yes or no), number of time steps, CFL number, K2 dissipation factor, K4 dissipation factor, average error criteria for convergence, and max error criteria for convergence.

2.7 Static Aeroelastic Parameters Module

The Static Aeroelastic Parameters Module controls the inputs for the static aeroelastic analysis. The module is shown in Figure 11. Variables that can be set here include number of time steps, CFL number, K2 dissipation factor, K4 dissipation factor, average error criteria for convergence, and max error criteria for conver-



Steady CFD Analysis Parameters

Continue From Previous Run?: ☐ Yes ☒ No

Use Local Time-Stepping?: ☐ Yes ☒ No

Number of Time Steps:

CFL Number:

K2 Dissipation Parameter:

K4 Dissipation Parameter:

Average Error Bound:

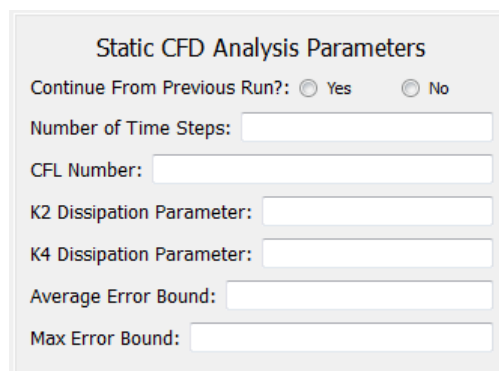
Max Error Bound:

Figure 10: Steady CFD Parameters Module.

gence. Additionally, the option to continue from a previous static run is available. This option uses the output from the previous run as the input for the current run.

2.8 Structural Model Parameters Module

The Structural Model Parameters Module controls the inputs for the structural model of the wing used in the static aeroelastic and flutter analyses. The module is shown in Figure 12. Variables that can be set here include the number of Gaussian integration points used in the consistent load vector calculation, Young's modulus,



Static CFD Analysis Parameters

Continue From Previous Run?: ☐ Yes ☐ No

Number of Time Steps:

CFL Number:

K2 Dissipation Parameter:

K4 Dissipation Parameter:

Average Error Bound:

Max Error Bound:

Figure 11: Static Aeroelastic Parameters Module.

Structural Model Parameters

Gaussian Integration Points: ☐ 9 Points
☒ 13 Points

Young's Modulus:

Shear Correction Factor:

Poisson's Ratio:

Thickness and Density

Span Locations:

Chord Locations:

Thickness Values:

.0028,.0028,.0028,.0028,.0028,.0028,.0093,.0065,.0045,.0045,.0037
.0028,.0028,.0028,.0028,.0028,.0093,.0093,.0065,.0045,.0045,.0037
.0028,.0028,.0028,.0028,.0080,.0093,.0093,.0065,.0045,.0045,.0037
.0028,.0028,.0028,.0161,0.120,.0093,.0093,.0065,.0045,.0045,.0037
.0028,.0028,0.161,0.161,0.120,.0093,.0093,.0065,.0045,.0045,.0037
.0186,0.161,0.161,0.120,.0093,.0093,.0065,.0045,.0045,.0037
.0186,0.161,0.161,0.161,0.120,.0093,.0093,.0065,.0045,.0045,.0037
.0028,.0028,0.161,0.161,.0028,.0028,.0028,.0028,.0028,.0028,.0028
.0028,.0028,0.161,0.161,.0028,.0028,.0028,.0028,.0028,.0028,.0028

Density Values:

5800,5800,5800,5800,5800,5800,5800,5800,5800,5800,5800,5800
5800,5800,5800,5800,5800,5800,5800,5800,5800,5800,5800,5800
5800,5800,5800,5800,5800,5800,5800,5800,5800,5800,5800,5800
5800,5800,5800,5800,5800,5800,5800,5800,5800,5800,5800,5800
5800,5800,100000,100000,5800,5800,5800,5800,5800,5800,5800,5800
5800,5800,100000,100000,5800,5800,5800,5800,5800,5800,5800,5800
5800,5800,100000,100000,5800,5800,5800,5800,5800,5800,5800,5800
5800,5800,100000,100000,5800,5800,5800,5800,5800,5800,20000,20000
5800,5800,100000,100000,5800,5800,5800,5800,5800,5800,20000,20000

Location of Chordwise Finite Element Edges:

Location of Spanwise Finite Element Edges:

Figure 12: Structural Model Parameters Module.

shear correction factor, Poisson's ratio, wing structural thickness and density, and chordwise and spanwise finite element distribution.

The structural thickness and density values are set by entering comma separated lists of numbers representing percent span and percent chord into the *Span Locations* and *Chord Locations* boxes respectively. In the *Thickness Values* and *Density Values* boxes, arrays of numbers are used to represent the thickness and density at the percent span and percent chord locations that are listed in the *Span Locations* and *Chord Locations* boxes. There is one row for each percent chord and one column for each percent span. The *Span Locations* box requires two or more comma separated numbers between 0 and 2. The first number must be 0 and the

last number must be 2. The wing root is at 0, the kink is at 1, and the wing tip is at 2. The *Chord Locations* box requires two or more comma separated numbers between 0 and 1. The first number must be 0 and the last number must be 1. The wing leading edge is at 0 and the trailing edge is at 1. Bilinear interpolation is used to determine the thickness and density between the entered values.

The finite element discretization of the wing is set using similar comma separated lists of numbers representing percent chord and percent span. These numbers will define the edges of the triangular finite elements. Each of these lists requires at least two numbers. For the chordwise finite elements, the first number must be 0 (representing the leading edge) and the last number must be 1 (representing the trailing edge). For the spanwise finite elements, the first element can be any value greater than or equal to 0 and less than 2. The last value must be 2. Here, the wing root is at 0, the kink is at 1, and the wing tip is at 2. When the first value in the list is larger than 0, the flexible finite element structure does not begin at the wing root. In the simulation, anything inboard of the first finite element is treated as rigid. If the first number in the spanwise list is less than 1, then there must be a finite element edge at the kink (a value of 1 must be in the list).

The difficult, confusing process of designing a structural wing model is simplified with the use of the *Plot Planform with FE, thickness, and density distribution* plotting tool. This tool is described in more detail in Section 2.11.

Flutter Parameters		Initial Condition:	
Max Time Steps: 100000	K2 Dissipation: 0.2	<input type="radio"/> Previous Run	<input type="radio"/> Steady
CFL Number: 2.5	K4 Dissipation: 0.04	<input checked="" type="radio"/> Static	Structural Damping: 0.0
Normalizing Mode: 1	IC Mode: 1	IC Velocity: 0.05	Flow3D Write Frequency: 1000

Figure 13: Flutter Module.

2.9 Flutter Parameters Module

The Flutter Parameters Module controls the inputs for the dynamic aeroelastic analysis. The module is shown in Figure 13. Variables that can be set here include maximum number of time steps, CFL number, K2 dissipation factor, K4 dissipation factor, the normalizing mode, the initial condition mode, initial velocity, structural damping, and how frequently Flow3d.vtk files (used for 3-D visualization) are generated. Additionally, the option to begin the simulation from a steady, static, or previous flutter run is available. These parameters are only used to generate the flutter input files and move the flutter executable to the run directory. Unlike the other programs, the flutter simulation cannot be started from within dfLap. Once the parameters are set and the input files are generated, the flutter analysis can be easily run from the command line.

2.10 Analysis Module

The Analysis Module allows the user to run different analyses and to save and load dfLap input files. The module is shown in Figure 14.

Entering all the data into the dfLap panel can be a tedious process. To avoid this redundant work, the user is able to load a previous dfLap state or save the current state using the *Load dfLap Input File* and *Save dfLap Input File* buttons. Upon clicking one of these buttons, a file selection window appears, and the user

can select the location of the dfLap input file. Sample input files are provided with dfLap to help users get started.

The six buttons below the *Load* and *Save* buttons are used to run different meshing or analysis programs (or to generate the flutter input files). The programs and the dfLap modules required to run them are listed in Table 2. It is assumed that the correct executable files are entered in the Input/Output Files Module. Also, with the exception of the flutter program, for each program to run, all of the programs above it in the table must have already been run. For flutter analyses, the requirements for previously run programs depend on the initial conditions chosen. For example, static results are not necessary when the analysis is started from the steady CFD solution.

2.11 Plotting Module

The plotting module gives users access to visualizations that can help with model development and analysis of results. The module is shown in Figure 15. Each of the plot buttons generates matplotlib [3] plot windows which are displayed on the screen and a *pdf* file which is saved in the *Output Directory* from the Input/Output File Module.

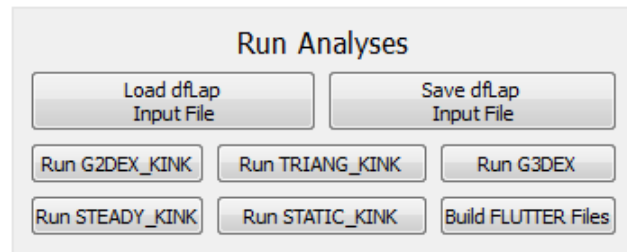


Figure 14: Analysis Module.

Table 2: dfLap programs and requirements.

Program	Description	Required dfLap Modules
<i>G2DEX_KINK</i>	2-D mesh generation	Wing Planform Parameters, Mesh Parameters, Airfoil Parameters
<i>TRIANG_KINK</i>	Removal of similar nodes	None
<i>G3DEX</i>	3-D mesh generation	None
<i>STEADY_KINK</i>	Steady CFD analysis	Flow Parameters, Steady CFD Parameters
<i>STATIC_KINK</i>	Static aeroelastic analysis	Flow Parameters, Structural Model, Static Aeroelastic Parameters
<i>FLUTTER Files</i>	Build flutter input files and move flutter executable to output directory	Flow Parameters, Structural Model, Flutter Parameters

The *Plot Planform with FE, thickness, and density distribution* button generates four plots of the planform defined in the Wing Planform Parameters Module with (1) finite element discretization, (2) structural thickness distribution, (3) structural density distribution, and (4) mass distribution as defined in the Structural Model Parameters Module. The plots are saved in *planform-with-FE.pdf*. Example output is shown in Figures 16-19.

The *Plot 2D Mesh* button generates six plots, one of each of the 2-D mesh surfaces (symmetry plane, wing surface, wing tip, wing attach wall, far wall, and tunnel ceiling). The plots are saved in a file called *mesh-plots-2d.pdf*. Prior to generating these plots, the 2-D meshing analysis must have been completed. Example plots are shown in Figures 20-25.

The *Plot Steady Flow Distribution* and *Plot Static Flow Distribution* buttons

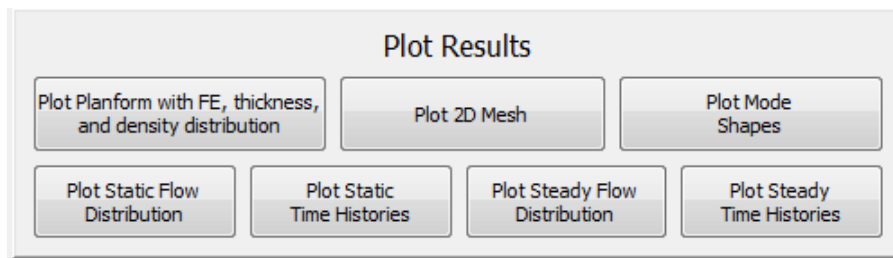


Figure 15: Plotting Module.

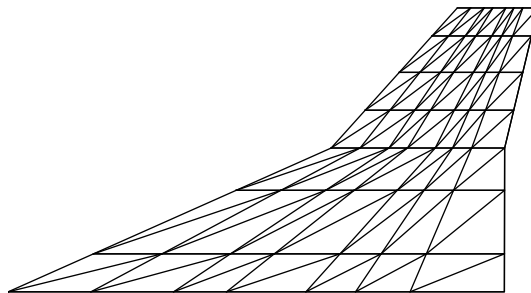


Figure 16: Wing planform with finite elements plot from dfLap.

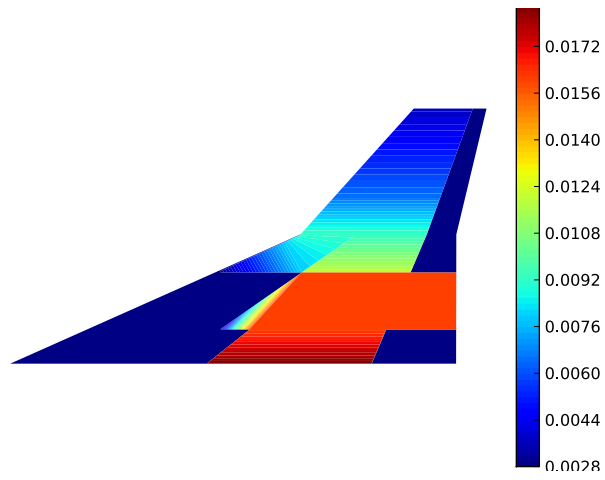


Figure 17: Wing planform with thickness distribution plot from dfLap.

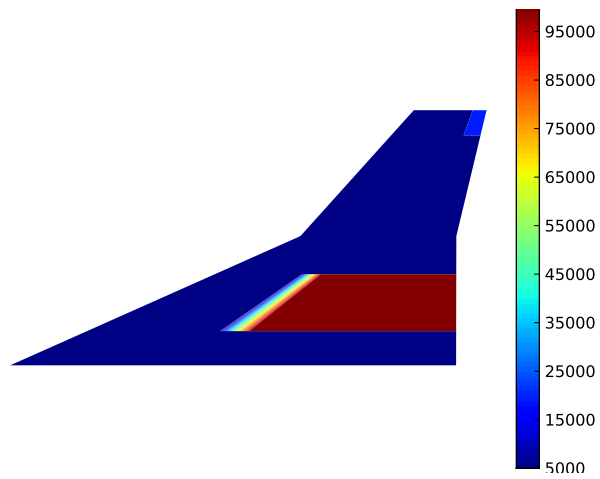


Figure 18: Wing planform with density distribution plot from dfLap.

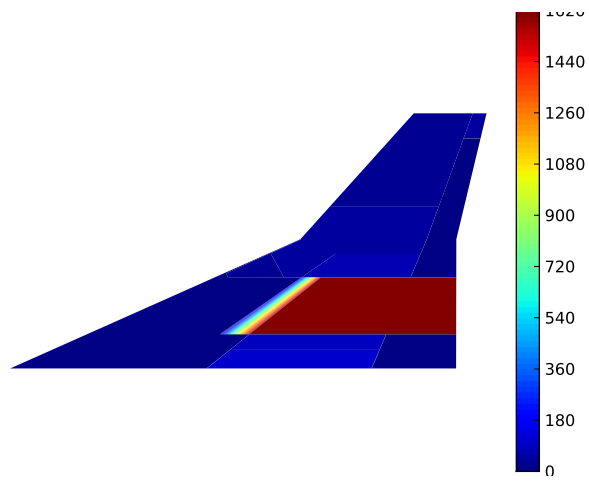


Figure 19: Wing planform with mass distribution plot from dfLap.

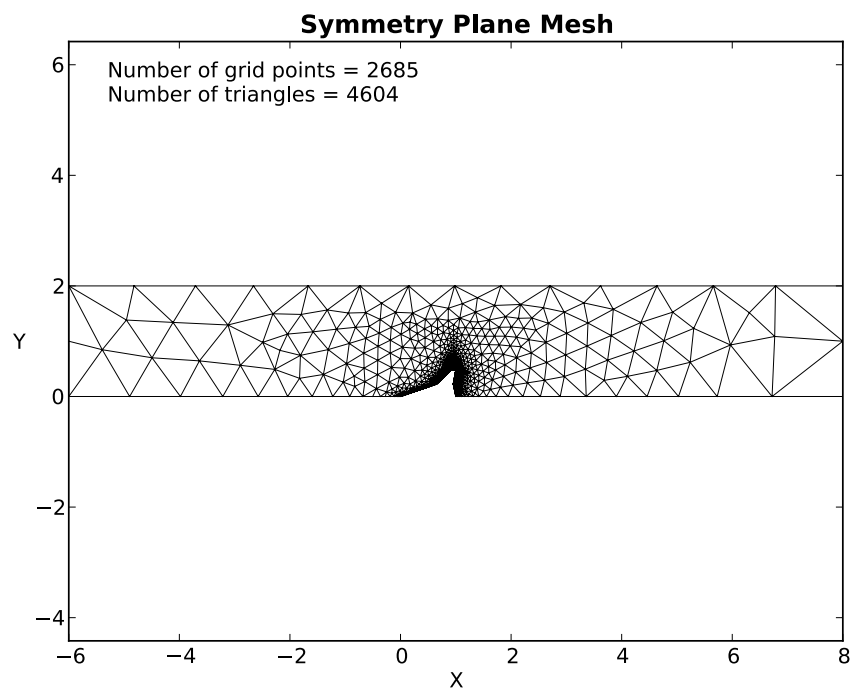


Figure 20: Symmetry plane mesh plot from dfLap.

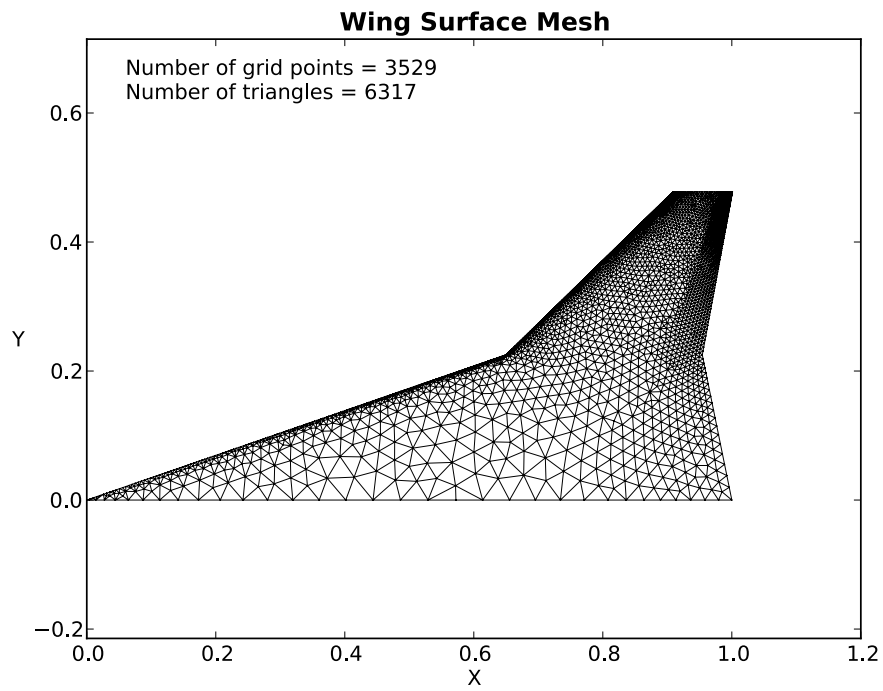


Figure 21: Wing surface mesh plot from dfLap.

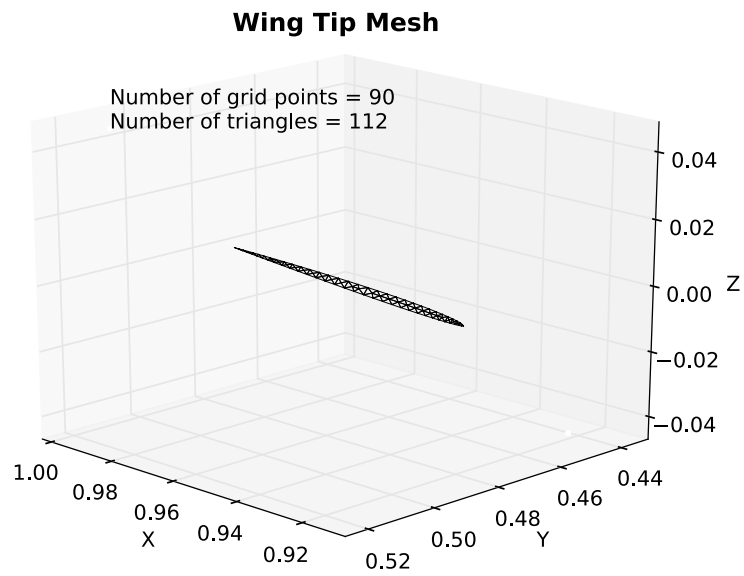


Figure 22: Wing tip mesh plot from dfLap.

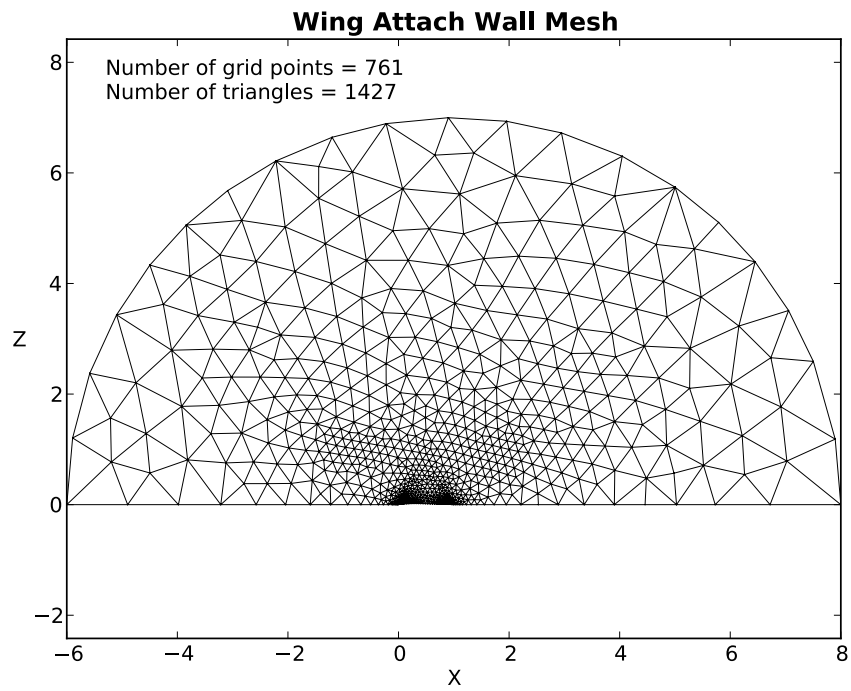


Figure 23: Attach wall mesh plot from dfLap.

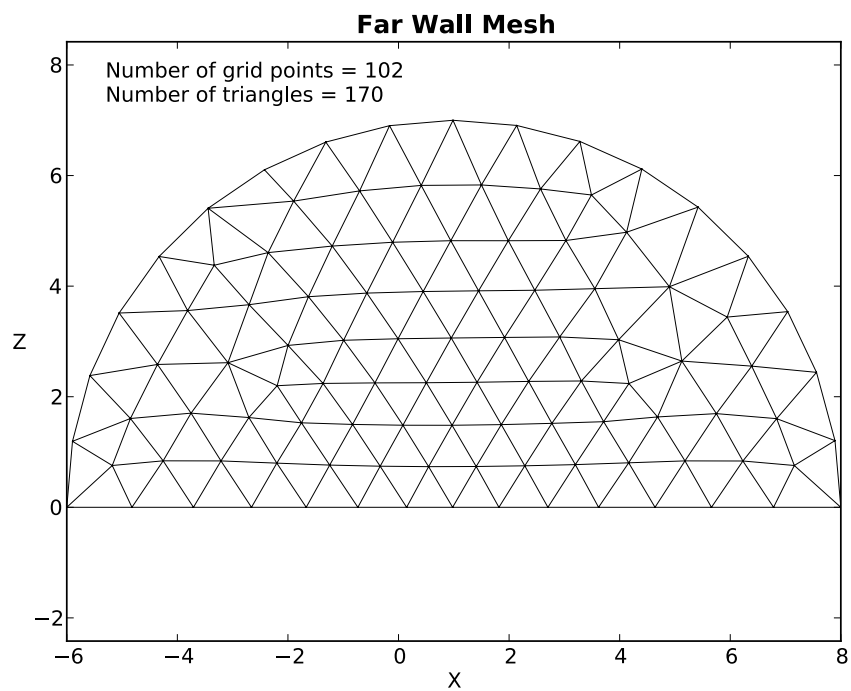


Figure 24: Far wall mesh plot from dfLap.

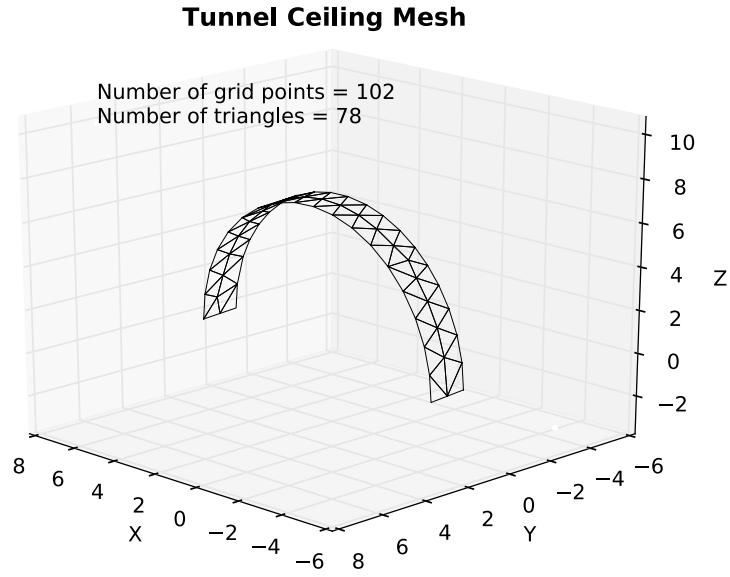


Figure 25: Tunnel ceiling mesh plot from dfLap.

generate plots of different flow parameters on the wing upper and lower surfaces. For *steady*, the plots are: Mach number, pressure coefficient (C_p), and pressure on the wing upper and lower surfaces. For *static*, seven plots are generated: deflection and Mach number, C_p , and pressure for the wing upper and lower surfaces. Figure 26 shows an example plot of the upper surface static C_p distribution. The plots are saved in files called *steady_flow_plots.pdf* and *static_flow_plots.pdf*.

The *Plot Steady Time Histories* and *Plot Static Time Histories* buttons generate plots that show the convergence of key parameters during the simulation. Parameters include average residual, maximum residual, wing tip deflection (*static* only), lift coefficient, moment coefficient, drag coefficient, and maximum Mach number. Example plots from a static analysis are shown in Figures 27 and 28. The plots are saved in files called *steady_time_plots.pdf* and *static_time_plots.pdf*.

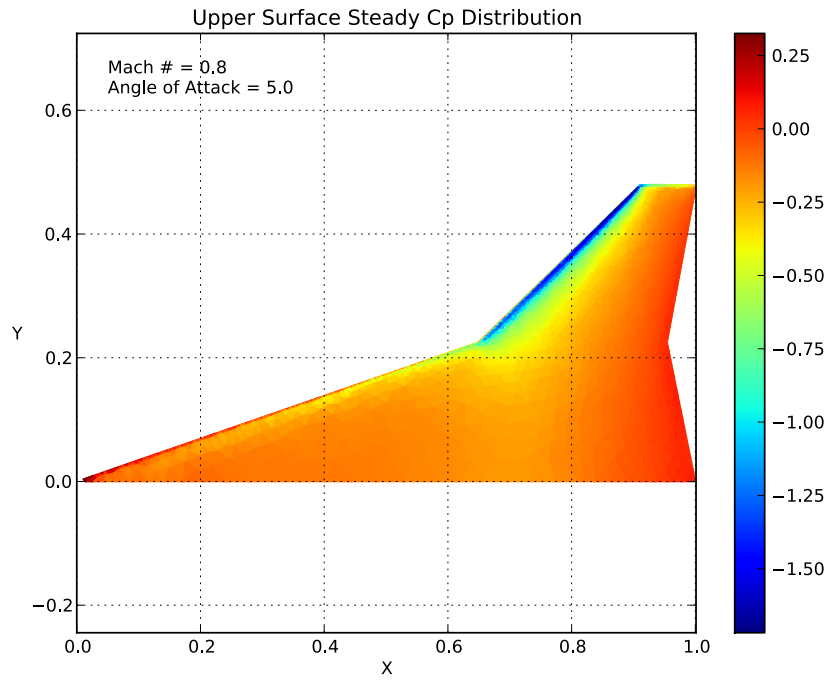


Figure 26: Example of the steady flow distribution plots from dfLap.

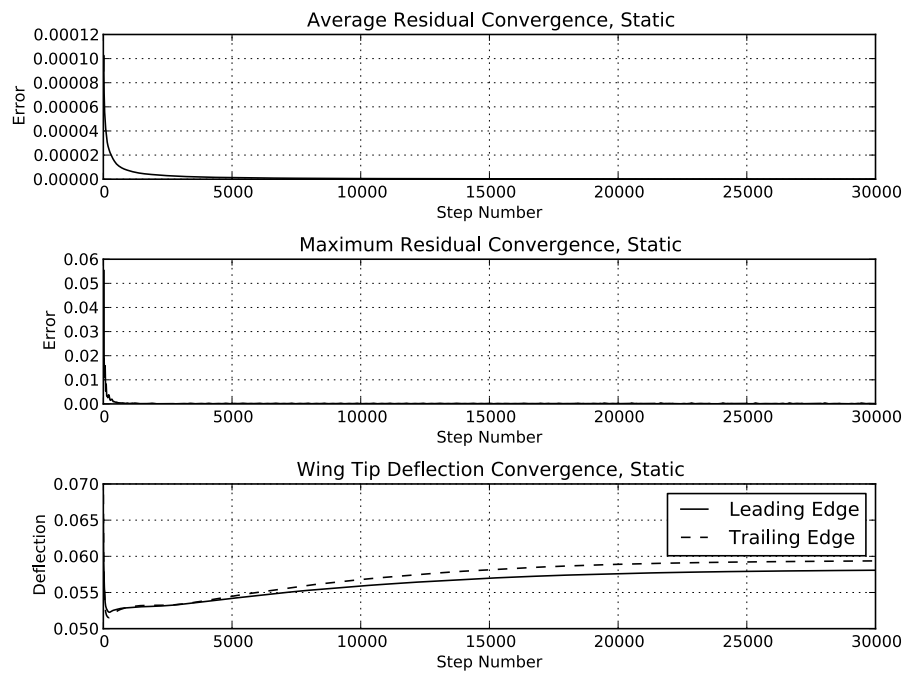


Figure 27: Example of the first page of the static time plots from dfLap.

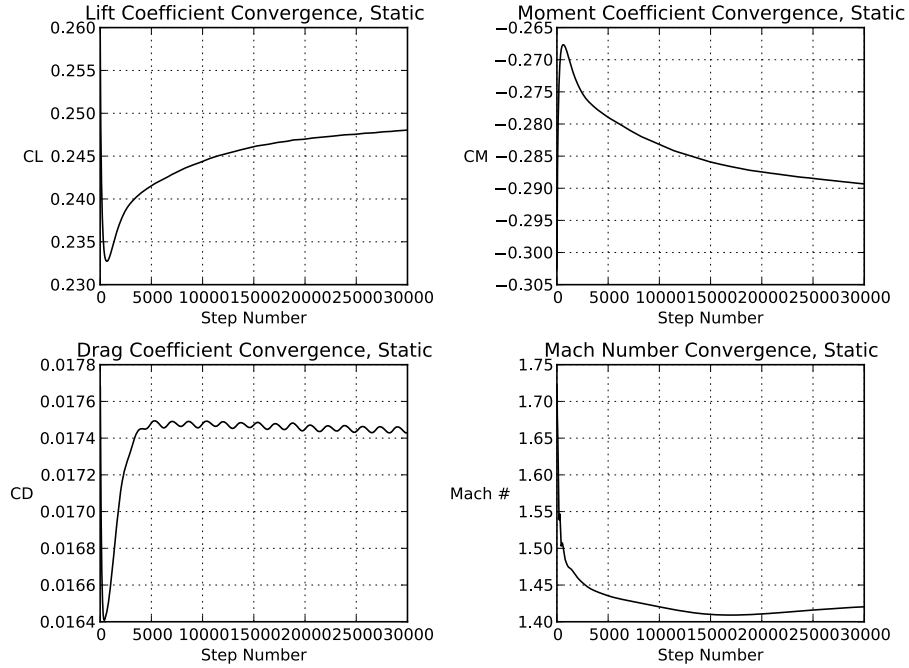


Figure 28: Example of the second page of the static time plots from dfLap.

The *Plot Mode Shapes* button generates contour plots of the first six structural modes. An example plot is shown in Figure 29. The plots are saved in a file called *structural_modes.pdf*.

3.0 dfLap's Python Functions

The dfLap source code consists of a series of Python functions. The functions are divided into four categories: input file generation, program execution, results plotting, and basic dfLap actions. Each category of functions is contained in its own Python module (A module is a single text file with a *.py* extension.). A brief explanation of the modules is provided to aid the user should the need to modify the source code arise.

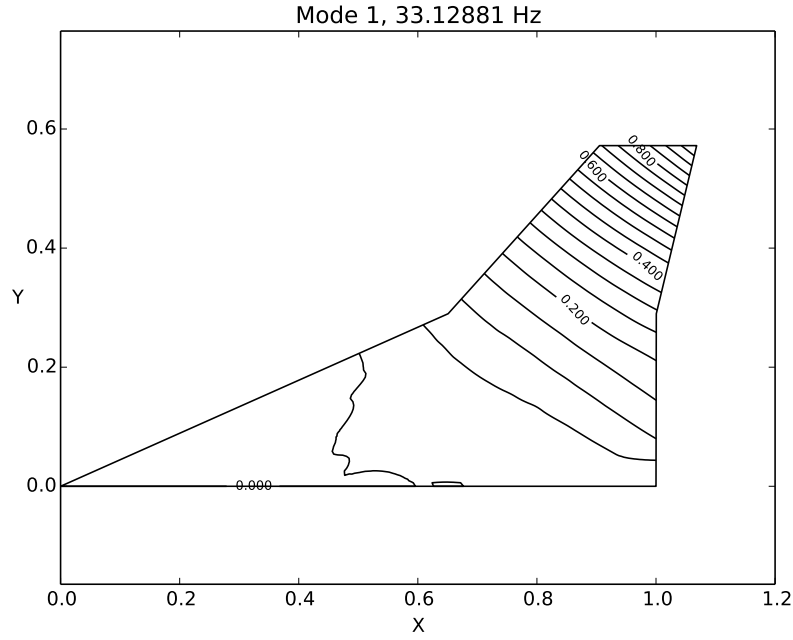


Figure 29: Example of a mode shape contour plot from dfLap.

3.1 Input File Generation

The input file generation functions create all required input files for use in the meshing and analysis programs. The input files are built when the *Run* button for each program is clicked. The input files are all text based and can be viewed in the *Output Directory* once they have been generated. Input file generation functions are all contained in the *dfLap_input_file_builder_3.0.py* module. With the exception of the parameters that can be set in the dfLap interface, the input file parameters are hard coded in the functions. If the user desires, for example, a different background grid connectivity, he will need to edit the *bkgrd* matrix in the *build_g2dex_kink_input* function.

3.2 Program Execution

The program execution functions are responsible for calling the input file generation functions, copying the program executable files to the output directory, and running the meshing and analysis programs. The functions share the *dfLap-3.0.py* module with the basic dfLap actions functions. They are easy to find within the module as all the function names begin with *Run*.

3.3 Results Plotting

The results plotting functions are used to generate the plots explained in Section 2.11. The functions are found in the *dfLap-plotting.py* module.

3.4 Basic dfLap Action

The basic dfLap actions functions control the GUI and the file dialog interfaces for the linking of executables, the loading and saving of the dfLap input file, and the notification pop-up windows. These are by far the least interesting of the functions in dfLap, and it is unlikely that the user will want or need to edit these.

4.0 Example Solution

dfLap provides a systematic method for generation of aeroelastic meshes and analysis input files. Through the combination of dfLap and the new *kink* versions of the meshing and aeroelastic programs, many problems that plagued previous versions of the process have been removed. This chapter outlines the steps used to calculate a set of static aeroelastic results using dfLap.

4.1 Setup

Setting up an aeroelastic solution in dfLap is as simple as filling in the blanks. It is recommended that the user follow the order in which the modules are described in Chapter 2.

To run any of the meshing or aeroelastic analyses, the appropriate executable programs must be available. These should be entered in the boxes in the Input/Output File Module. The location of the executables is unimportant, as they will be copied to and run in the *Output Directory*. When setting up a new analysis, it is recommended that the *Output Directory* be a new, empty directory. This directory is where the input and output files for all the programs will be saved.

If the user is unsure of the exact planform parameters desired, the *Plot Planform with FE, thickness, and density distribution* function in the Plotting Module is useful for checking and refining the planform. This plot is generated based only on planform, structural finite element, thickness, and density discretization parameters. No analyses are run to generate this plot. The geometry is calculated and plotted using only the Python functions. As with all the plotting functions, the *Output Directory* must be defined so the plots can be saved.

Once the parameters in all of the modules have been filled in, the dfLap state can be saved. By clicking the *Save dfLap Input File* button, the user can select a location and file name for the input file. All dfLap input files have a *.ifl* extension. The saved file contains plain text with all the parameters that were entered into the interface (0's and 1's are used for parameters controlled by radio buttons). The *Load dfLap Input File* button can be used to return the interface to a saved

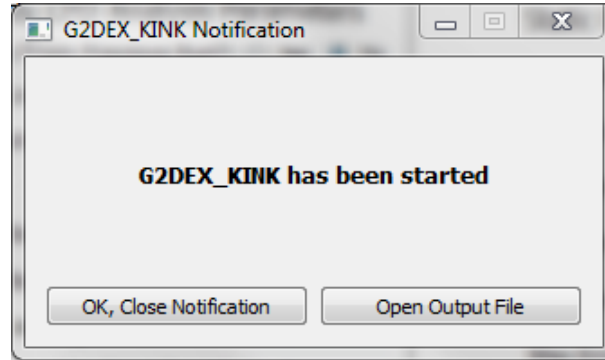


Figure 30: Example of a dfLap pop-up notification window.

state.

4.2 Generating the Fluid Mesh

Fluid mesh generation is started by clicking the *Run G2DEX_KINK* button. Running the 2-D mesh program is a significant strength of dfLap. The background mesh is automatically adjusted and the δ values defined at the background mesh grid points are scaled based on the wing planform. This nearly guarantees a good mesh every time the program is run. Prior to dfLap, this work had to be done by hand, and often took hours to complete.

When any *Run* button is clicked in dfLap, the appropriate input files are generated, the executable file is copied to the *Output Directory*, and the program is run. A pop-up window appears to notify the user that the program has started. The pop-up window has an option to open the output file (Figure 30). When the *Open Output File* button is clicked, the program standard out will be opened in the default program for displaying *.txt* files. Usually a quick look at the output file will reveal if the program has executed successfully, has failed to complete, or is in the process of running.

Once the 2-D mesh program has run successfully, it is advisable to check the resulting mesh with the *Plot 2D Mesh* function. As described in Section 2.10, this function will generate a series of plots that show the 2-D mesh.

To prepare for 3-D mesh generation, *TRIANG_KINK* must be run to identify and remove nodes that are located on top of one another. After this, the 3-D mesh can be generated by clicking *Run G3DEX*. This process can take several minutes. It is advised that the user open the output file and monitor it as the program is running. As stated previously, each time a program is run, a pop-up notification will appear giving the user direct access to the output file without having to locate it in the *Output Directory*.

4.3 Running the Aeroelastic Programs

Once the 3-D fluid mesh has been generated, the aeroelastic analysis programs can be run. First, the steady CFD solution must be run to calculate the flow around the rigid wing. After this has completed, the results are used as the initial conditions for the static aeroelastic analysis. The steady CFD or the static aeroelastic results can be used to initialize the flow for flutter calculations.

The steady CFD analysis is run by clicking the *Run STEADY_KINK* button. The first time this analysis is run, the *Continue From Previous Run?* option should be set to *No*. Once the analysis has completed, the results can be checked by plotting the steady flow distribution and the time histories. The flow distribution should be examined for abnormal discontinuities or peaks in the flow parameters that could imply a problem with the solution. The time histories can be used to examine how the solution is converging. If a more converged solution is desired,

the simulation can be restarted by setting the *Continue From Previous Run?* option to *Yes* and running the program again. The output from the first run will be copied into the initial conditions file and used to start the new run. This process can be repeated until suitable convergence occurs.

After a steady CFD solution has been calculated, the static aeroelastic simulation can be run. The process for this is similar to the steady solution. Flow distribution and time histories should be used to check the results.

The flutter program can use steady CFD or static aeroelastic results as initial conditions. When the user clicks the *Build FLUTTER Files* button, all the necessary files needed to run a flutter analysis are placed into the output directory. Because the flutter program demands a large amount of resources and runs for a long period of time, it is not started by dfLap. Instead, the user must run the program outside of dfLap. The process of running the program differs by operating system and hardware configuration.

REFERENCES

- [1] “Python Programming Language – Official Website,” <http://python.org>, 2013, [Online; accessed 26-August-2013].
- [2] “Numpy,” <http://numpy.org>, 2013, [Online; accessed 26-August-2013].
- [3] “Matplotlib: Python Plotting – Matplotlib 1.3.0 Documentation,” <http://matplotlib.org>, 2013, [Online; accessed 26-August-2013].

APPENDIX B

Modifications to the Aeroelastic Code

B.1 Introduction

Several updates were made to the existing computational aeroelastic code. Variations of this code were used in References [4, 9–11, 37–42, 44–51, 53–56, 96, 126].

The updates include:

1. The ability to model cranked wing planforms was added. This includes the ability to sweep the kink line and the wing tip. This is discussed in Section B.2.
2. A programming error that was prohibiting the generation of highly swept wings was found and fixed. This is discussed in Section B.3.
3. The wing tip meshing algorithm for the CFD portion of the code was improved. This is discussed in Section B.4.
4. The code was updated to make use of the standard LAPACK (Linear Algebra Package) library. This was implemented using a new version of the Intel Fortran compiler with the Intel Math Kernel Library (IMKL). This is discussed in Section B.5.

5. An improved wing density and thickness distribution scheme was implemented. A programming error that was affecting the thickness distribution was found and corrected. This is discussed in Section B.6.
6. The input files for the static aeroelastic and flutter programs were updated to include the speed of sound and ratio of specific heats, γ . This is discussed in Section B.7.
7. The input files for the static aeroelastic and flutter programs were updated to include structural finite element discretization. Previously, FE discretization was written into the source code, and changing it required a recompile. This is discussed in Section B.8
8. A program with a graphical user interface (GUI) was created to simplify the process of running the aeroelastic code. The program is called dfLap, the Inviscid Flutter Analysis Program. It can be used to generate input files, run simulations, and produce graphical output. dfLap is presented in Section B.9.
9. The steady CFD, static aeroelastic, and flutter programs were updated to produce new output files which are useful for analyzing simulation results. The new output files are discussed in Section B.10.

Details on these updates are included in this appendix. They are also all clearly commented in the program source code and flagged with the string “*K*” By searching for this string, the updates can be found.

B.2 Cranked Wing Geometry

To generate cranked wing planforms, changes were made to the source code for *g2dex*, *triang*, *steady*, *static*, and *flutter*. Most of the changes were made in the 2-D meshing code, *g2dex*, and the structural modeling sections of the static aeroelastic and flutter codes, but the other codes required at least minor updates.

B.2.1 *G2dex* Updates for Cranked Wing Geometry

The source code for the 2-D meshing program, *g2dex*, required several updates so that cranked wing meshes could be generated. The updated version of this code is called *g2dex_kink*.

New lines were added to the input file, *g2dex.dat*, to define the inboard and outboard leading and trailing edge sweep angles, the sweep angle of the kink line and the tip, the location of the kink, and independent aerodynamic thickness tapering of the inboard and outboard sections. Details of the parameters that define the wing planform are given in Section 2.4.2. The new input file format is read by an updated subroutine, *DATAIN2*, which replaces the old subroutine, *DATAIN*. The coordinates of the wing boundary are calculated in a new subroutine, *PLANFORM*. The updated version of *g2dex.dat* can be generated automatically with *dfLap* [Appendix A].

Previous versions of the meshing code required the wing tip to be parallel to the wing root. In the updated version, both the wing tip and the kink line (the line between the leading edge kink and trailing edge kink) can be rotated. New subroutines created to handle this change are listed below.

1. *PCHORD* and *PSPAN* take an X, Y coordinate pair as input and return the location of a point on the wing in percent chord (between 0 and 1) or percent span (between 0 and 1 for the inboard portion of the wing and between 1 and 2 for the outboard portion of the wing).
2. *CHORDLEN* takes an X, Y coordinate pair as input and returns the wing chord length at that point. Chords for the cranked geometry are lines of constant percent span.
3. *WINGTHICKNESS* takes an X, Y coordinate pair as input and returns the aerodynamic thickness of the wing at that point. The thickness of the wing varies linearly along lines of constant percent chord.
4. *LINEX* is used in the other subroutines listed here to calculate the intersection of line segments. This is useful for determining if points are on the inboard or outboard portion of the wing.

The outboard portion of the cranked wing planform is defined by two new boundary lines, numbered 21 and 22. The lines are shown in Figure 2.2. Adding them to the code required changes to the boundary line discretization subroutines, *BLINES* and *BLINED*, and the 2-D surface definition subroutine, *PATCH*, which was updated to *PATCH2*. The updated 2-D surface definitions are listed in Table 2.2.

To pass the cranked planform geometry to other programs, the output file *wing.inf* was updated. Wing geometry is stored in this file. It is used by *triang*, *steady*, *static*, and *flutter*.

B.2.2 *Triang* Updates for Cranked Wing Geometry

The source code for the program that removes overlapping nodes, *triang*, has been updated to accommodate cranked wing planforms. The updated version of the program is called *triang_kink*.

Two changes were made to *triang*. The format of the input file, *wing.inf*, generated by *g2dex_kink* was changed, so *triang* was updated to read the new format. Also, the output file, *corner.inf* was updated to include the new geometry. This file is used by *steady*, *static*, and *flutter*.

B.2.3 *Steady* Updates for Cranked Wing Geometry

The source code for *steady*, the steady CFD program, was updated to accommodate cranked wing planforms. The updated version of the program is called *steady_kink*.

The *steady* code was updated to be compatible with the new version of *wing.inf* generated by *g2dex_kink*. In addition, the *MIRROR* subroutine was updated to accommodate the cranked wing geometry. *MIRROR* takes the upper half of the computational domain created by the 3-D meshing program and creates a mirror image of it to create the full computational domain.

B.2.4 *Static* Updates for Cranked Wing Geometry

The source code for the static aeroelastic program, *static*, was updated to work with cranked wing planforms. The updated version of the program is called *static_kink*.

In the old version of *static*, the shape of the wing structural model was defined in the input file, *static.dat*. This could cause problems, as the structural and fluid models could have dissimilar geometry. To fix this, the lines defining wing geometry were removed from *static.dat*, and the source code was updated to read the geometry from the *wing.inf* file. The *wing.inf* file is generated by *g2dex_kink*, so the fluid and structural geometry is now guaranteed to be consistent. The updated version of *static.dat* can be generated automatically with dfLap [Appendix A].

The structural model discretization was previously completed in a subroutine called *ELEMTXY*. This subroutine created the nodes and element connectivity that was used to generate the structural finite element model. *ELEMTXY* was updated to accommodate the cranked wing geometry, and is now called *ELEMTXY_KINK*. The new version generates element nodes along lines of constant percent chord and constant percent span as shown in Figure 3.8(c).

The subroutine *SDMAP*, which establishes the connection between fluid nodes and structural finite elements as shown in Figure 4.2 has been updated to accommodate the cranked wing geometry. The updated subroutine is called *SDMAP_KINK*.

The subroutine used to calculate structural element deformation and local coordinate systems, *DEFCOOR*, has been updated. In the old version, the initial local coordinate systems were always coincident with the global coordinate system. This fact was used to simplify the element deformation calculations. In the updated version, in general, the initial local coordinate systems are different than the global coordinate system. Because of this, the first time *DEFCOOR* is

run in the updated program, the initial nodal coordinates and element coordinate systems are saved. They are then referenced each subsequent time *DEFCOORD* is run to calculate element deformation.

Like the steady CFD program, *static* uses the subroutine *MIRROR* to create the full computational domain. This subroutine was updated to accommodate the cranked wing geometry.

Many of the changes to *static* relied on the subroutines *PCHORD*, *PSPAN*, *LINEX*, and *CHORDLEN*. These were presented in Section B.2.1.

B.2.5 *Flutter* Updates for Cranked Wing Geometry

The source code for the *flutter* program is very similar to the *static* code. Because of this, the *flutter* code updates match the *static* code updates.

B.3 High Wing Sweep

The old version of the 2-D meshing program, *g2dex*, would often fail to run when it was used to generate wings with leading edge sweep greater than 30 degrees. In addition, as the sweep was increased, the elements on the wing would become increasingly distorted. A wing with distorted elements is shown in Figure B.1.

A programming error responsible for the distortion was discovered in the *PATCH* subroutine. The error was in the process responsible for mapping between the curved and the flat representations of the wing surface. The error was corrected, and the updated program is now able to create meshes for any leading edge sweep without distortion (Figure B.2). For an explanation of mapping

between curved and flat representations of the 2-D mesh, see Section 2.4.6.

B.4 Wing Tip Meshing

The old version of the 2-D meshing program, *g2dex*, would often fail while attempting to create a mesh on the wing tip. Two changes were made to correct this problem. First, two files formerly being used to define the airfoil shape and mesh discretization were combined into one. Second, the tip mesh can now be explicitly defined in an input file, avoiding the use of the advancing front method (AFM) on the wing tip altogether.

The old program used two files to define the mesh on the wing tip boundary lines. One file, *airfoil.dat*, defined the shape of the airfoil and the boundary lines. A second file, *omesh.out*, was used to define the location of the mesh grid points on the boundary lines. Making these two files work together involved mapping the grid points, defined by *omesh.out*, onto the shape, defined by *airfoil.dat*. These

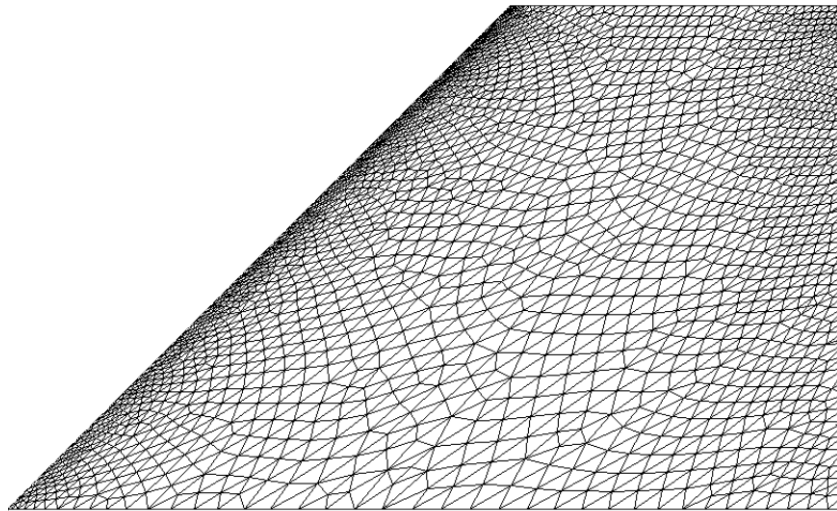


Figure B.1: Wing surface with distorted elements.

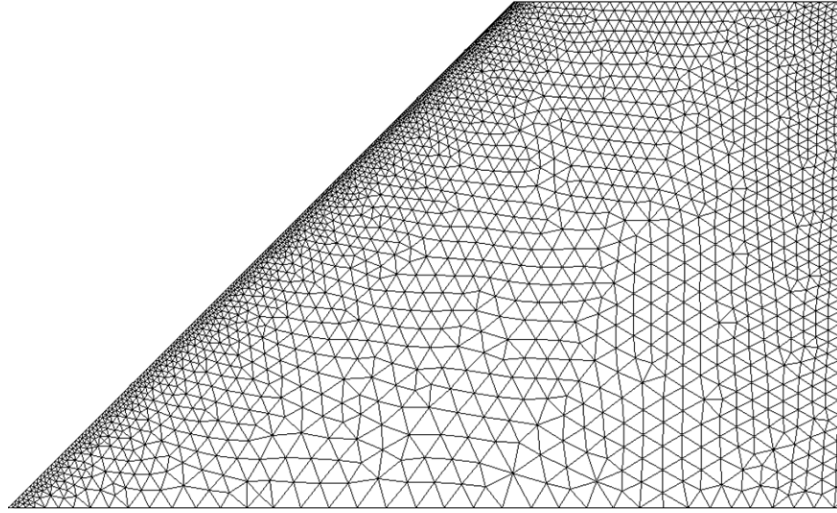


Figure B.2: Wing surface after code update.

files were also used to define the mesh on the wing surface. In the new version of the code, *g2dex_kink*, one file is used to define both the wing shape and boundary line discretization. The combined file kept the name of the old *omesh.out* file. The *airfoil.dat* file is no longer used.

The old program used the AFM to discretize the wing tip. This process has created problems throughout the life of the meshing code. Different solutions have been attempted, but often, the user was forced to go into the source code and alter the logic until a usable mesh was generated. To avoid issues like this, the code has been updated to read in a user provided tip mesh which can be mapped to any airfoil shape. A new subroutine, *TIPMESH*, reads a pre-defined wing tip mesh geometry from a connectivity file *nds_tip.txt*, and a coordinates file, *xyz_tip.txt*. The files are in the 2-D flat wing tip coordinate system (see Section 2.4.6). The coordinates are scaled based on the geometry defined in *omesh.out* and mapped onto the curved 2-D tip surface. The *omesh.out*, *nds_tip.txt*, and *xyz_tip.txt* files must all “match”. That is, they all must use the same number of points to define

the airfoil shape.

One pre-defined mesh is provided with dfLap, but new meshes can be developed and used. Using the AFM to discretize the wing tip is still an option, and if the process is completed successfully, the tip mesh is saved for use in future meshes. Details of the tip meshing process can be found in the dfLap User's Guide in Appendix A.

B.5 The Intel Fortran Compiler and LAPACK

The static aeroelastic and flutter programs require some basic linear algebra processes including solving systems of equations, calculating matrix inverses, and eigenvalue and eigenvector calculation. In the old code, these processes were accomplished using the International Mathematics and Statistics Library (IMSL) numerical software libraries which were included with an old version of the DIGITAL Visual Fortran compiler. Unfortunately, this compiler is no longer supported by the manufacturer and getting it to work with new operating systems and computer hardware has become problematic. The leading Fortran compiler in the current marketplace is provided by Intel and is part of the Intel Parallel Studio [129]. It includes the Intel Math Kernel Library (IMKL) [130], which is an optimized version of several open source libraries including the Linear Algebra Package (LAPACK) libraries. The IMKL libraries are highly vectorized and threaded and are designed to take advantage of modern Intel processors with multiple cores.

The new libraries were included in the static aeroelastic and flutter programs by writing a series of subroutines which are called the same way as old

IMSL subroutines (same names and input/output). The new subroutines call the IMKL/LAPACK subroutines and are included in a file called *imsl_replacement.f*. Using these subroutines with the new Intel Fortran compiler has improved static aeroelastic run times by a factor of two on an eight core Intel CPU.

B.6 Thickness and Density Distribution

With the updated geometry described in Section B.2, an improved mass and stiffness distribution scheme was also desired. The old version of the code allowed linear variation of structural density and thickness from root to tip. The new version of the code uses text based input files to define the thickness (*span_thickness_in.txt*) and density (*span_density_in.txt*) at 2000 locations along the wing span. The variation does not have to be linear or even continuous. The first 1000 lines apply to the inboard portion of the wing, while the second 1000 lines apply to the outboard portion. Input files can be generated using dfLap.

To handle the update, the subroutine that builds the structural finite element mass and stiffness matrices, *MKSTR*, was updated to a new version, *MKSTR_KINK*. Also, a new subroutine, *GET_THICK_RHO*, was written to read the input files, interpolate between the defined values, and pass the thickness and density to *MKSTR_KINK*.

In the process of updating the code, a programming error was discovered in *MKSTR* that was affecting the thickness distribution. A loop was being incremented incorrectly, which was causing incorrect thickness values to be applied to nearly the entire wing. Once the error was corrected, linear frequencies were re-

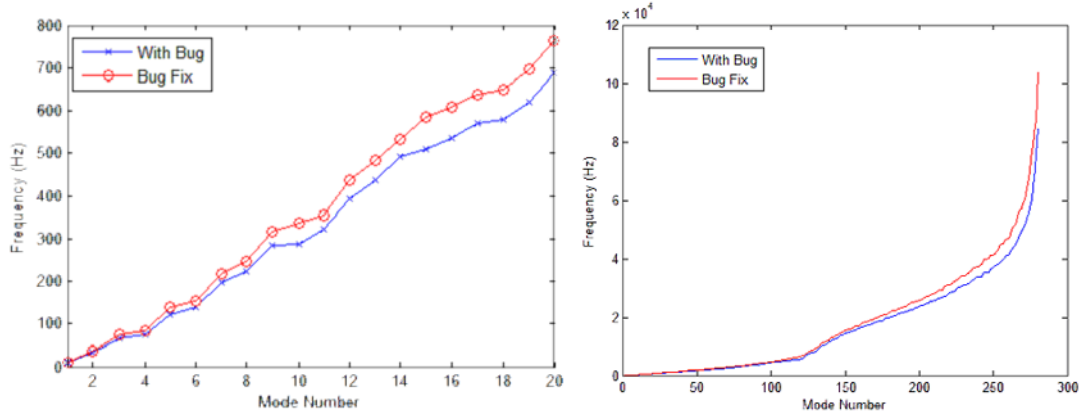


Figure B.3: Comparison of wing structural frequencies.

calculated and compared (Figure B.3). The correction resulted in a slightly stiffer wing. Under aerodynamic loading, the tip displacement decreased by about 4.5%.

B.7 Flow Properties

The speed of sound and the ratio of specific heats for the fluid is now defined in the input files for the flutter and static aeroelastic programs. In the old versions of *static* and *flutter*, the ratio of specific heats was hard coded into the program, and the speed of sound was calculated from the free stream velocity, which was read from the input file. By moving the ratio of specific heats to the input file, it can be changed without recompiling the source code. The free stream velocity line of the input file was replaced with a line for the speed of sound. This was done based on user preference.

B.8 Structural Discretization

In the old version of the static aeroelastic and flutter programs, structural finite element discretization was set by a series of parameters that were shared between multiple subroutines. Changing the discretization required the parameter values to be changed consistently in multiple places throughout the code and for the code to be recompiled. The parameters could not be changed to regular variables because their values were used to declare the size of the structural finite element matrices

In the new versions, structural discretization is set in the program input files, and can be changed quickly and easily. To implement the change, the source code was updated to take advantage of some features of Fortran 90. All the parameters from the old code still exist, but their values were set to a number larger than would be required by the largest structural model expected. This process caused a problem when the code was compiled with Fortran 77 when the structural finite element matrices were passed as input or output to subroutines. In Fortran 77, the subroutines expected a different sized matrix than was passed, and the program crashed. As a work around, Fortran 90 array sections were used. Array sections allow only a portion of an array to be passed to a subroutine. The updated version of the programs must be compiled using Fortran 90.

B.9 dfLap

To aid in the creation of input files for all the programs, a simplified front end program with a GUI was created. The program is called dfLap, the Inviscid

Flutter Analysis Program. Along with creating input files, it makes the process of generating the fluid mesh as simple as possible. It can also be used to generate visualizations of meshes and aeroelastic results. The dfLap User's Guide can be found in Appendix A.

B.10 Output Files

Previous versions of the steady CFD, static aeroelastic, and flutter programs produced output files to create flow visualizations on the wing surface. To improve the understanding of the flow field around the wing, a new subroutine, *FLOW3D_OUTPUT* was written to generate output files (*flow3d_steady.vtk* and *flow3d_static.vtk*) that contain flow parameters for the entire computational domain. The files are in the standard *vtk* file format [131] and are readable by a large number of CFD visualization programs. The output files contain the deformed wing geometry along with the pressure coefficient, Mach number, pressure, fluid velocity vector, and aerodynamic power at every grid point in the computational domain. The output files for the flutter program are generated at a user defined time step and can be used to create 3-D animations. The supplementary videos included with this dissertation were created using these 3-D animations. Images generated from these files can be found throughout Chapter 6.

REFERENCES

- [1] Bhatia, K. and Wertheimer, J., “Aeroelastic Challenges for a High Speed Civil Transport,” *34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, La Jolla, CA*, 1993, pp. 3663–3682.
- [2] Ashley, H., “Role of Shocks in the ‘Sub-Transonic’ Flutter Phenomenon,” *Journal of Aircraft*, Vol. 17, No. 3, 1980, pp. 187–197.
- [3] Tijdeman, H. and Seebass, R., “Transonic Flow Past Oscillating Airfoils,” *Annual Review of Fluid Mechanics*, Vol. 12, No. 1, 1980, pp. 181–222.
- [4] Bendiksen, O. O., “Role of Shock Dynamics in Transonic Flutter,” *AIAA Dynamics Specialists Conference, Dallas, TX*, 1992, pp. 401–414.
- [5] Lambourne, N., “Some Instabilities Arising from the Interactions Between Shock Waves and Boundary Layers,” Advisory Report 182, AGARD, April 1958.
- [6] Silva, W., Keller, D., Florance, J., Cole, S., and Scott, R., “Experimental Steady and Unsteady Aerodynamic and Flutter Results for HSCT Semispan Models,” *41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Atlanta, GA*, April 2000.
- [7] Guruswamy, G. P. and Goorjian, P. M., “Unsteady Transonic Flow Simulation on a Full-Span-Wing-Body Configuration,” *AIAA Journal*, Vol. 26, No. 12, December 1988, pp. 1450–1456.
- [8] Guruswamy, G. P., “Vortical Flow Computations on a Flexible Blended Wing-Body Configuration,” *AIAA Journal*, Vol. 30, No. 10, 1992, pp. 2497–2503.
- [9] Bendiksen, O. O., “A New Approach to Computational Aeroelasticity,” *32nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Baltimore, MD*, 1991.
- [10] Bendiksen, O. O., “Modern Developments in Computational Aeroelasticity,” *Proceedings of the Institution of Mechanical Engineers. Part G: Journal of Aerospace Engineering*, Vol. 218, No. 3, 2004, pp. 157–177.
- [11] Bendiksen, O. O. and Seber, G., “A Direct Fluid-Structure Coupling Scheme for Nonlinear Finite Element Aeroelastic Codes,” *International Forum on Aeroelasticity and Structural Dynamics, Stockholm, Sweden*, 2007.
- [12] Murman, E. M. and Cole, J. D., “Calculation of Plane Steady Transonic Flows,” *AIAA Journal*, Vol. 9, No. 1, 1971, pp. 114–121.

- [13] Ballhaus, W. F. and Bailey, F. R., "Numerical Calculation of Transonic Flow About Swept Wings," *5th AIAA Fluid and Plasma Dynamics Conference, Boston, MA*, June 1972.
- [14] Ballhaus, W. F. and Goorjian, P. M., "Implicit Finite-Difference Computations of Unsteady Transonic Flows About Airfoils," *AIAA Journal*, Vol. 15, No. 12, 1977, pp. 1728–1735.
- [15] Borland, C. J., Rizzetta, D. P., and Yoshihar, H., "Numerical Solution of Three-Dimensional Unsteady Transonic Flow Over Swept Wings," *AIAA Journal*, Vol. 20, No. 3, 1982, pp. 340–347.
- [16] Ballhaus, W. and Goorjian, P., "Computation of Unsteady Transonic Flows by the Indicial Method," *AIAA Journal*, Vol. 16, No. 2, 1978, pp. 117–124.
- [17] Rizzetta, D. P., "Time-Dependent Response of a Two-Dimensional Airfoil in Transonic Flow," *AIAA Journal*, Vol. 17, No. 1, 1979, pp. 26–32.
- [18] Borland, C. and Rizzetta, D., "Nonlinear Transonic Flutter Analysis," *AIAA Journal*, Vol. 20, 1982, pp. 1606–1615.
- [19] Guruswamy, G. P. and Goorjian, P. M., "An Efficient Coordinate Transformation Technique for Unsteady, Transonic Aerodynamic Analysis of Low Aspect-Ratio Wings," *25th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference, Palm Springs, CA*, May 1984.
- [20] Guruswamy, G. P., Goorjian, P. M., Ide, H., and Miller, G. D., "Transonic Aeroelastic Analysis of the B-1 Wing," *Journal of Aircraft*, Vol. 23, No. 7, July 1986, pp. 547–553.
- [21] Edwards, J. W., Bennett, R. M., Seidel, D. A., and Whitlow Jr, W., "Time-Marching Transonic Flutter Solutions Including Angle-of-Attack Effects," *Journal of Aircraft*, Vol. 20, No. 11, 1983, pp. 899–906.
- [22] Cunningham, H. J., Batina, J. T., and Bennett, R. M., "Modern Wing Flutter Analysis by Computational Fluid Dynamics Methods," *Journal of Aircraft*, Vol. 25, No. 10, 1988, pp. 962–968.
- [23] Batina, J. T., Seidel, D. A., Bland, S. R., and Bennett, R. M., "Unsteady Transonic Flow Calculations for Realistic Aircraft Configurations," *Journal of Aircraft*, Vol. 26, No. 1, January 1989, pp. 21–27.
- [24] Venkatakrishnan, V. and Jameson, A., "Computation of Unsteady Transonic Flows by the Solution of Euler Equations," *AIAA Journal*, Vol. 26, No. 8, 1988, pp. 974–981.

- [25] Bendiksen, O. O. and Kousen, K., "Transonic Flutter Analysis Using the Euler Equations," *AIAA Dynamics Specialists Conference, Monterey, CA*, April 1987.
- [26] Kousen, K. and Bendiksen, O. O., "Nonlinear Aspects of the Transonic Aeroelastic Stability Problem," *29th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference, Williamsburg, VA*, 1988.
- [27] Wu, J. C., Sankar, L. N., and Kaza, K. R. V., "A Technique for the Prediction of Airfoil Flutter Characteristics in Separated Flows," *Journal of Aircraft*, Vol. 26, No. 2, 1989, pp. 168–177.
- [28] Wu, J. C., *A Study of Unsteady Turbulent Flow Past Airfoils*, Ph.d. dissertation, Georgia Institute of Technology, Atlanta, GA, August 1988.
- [29] Guruswamy, G. P., "Unsteady Aerodynamic and Aeroelastic Calculations for Wings Using Euler Equations," *AIAA Journal*, Vol. 28, No. 3, 1990, pp. 461–469.
- [30] Guruswamy, G., "Numerical Simulation of Vortical Flows on Flexible Wings," *27th Aerospace Sciences Meetings, Reno, NV*, January 1989.
- [31] Guruswamy, G., "Vortical Flow Computations on Swept Flexible Wings Using Navier-Stokes Equations," *30th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Mobile, AL*, April 1989.
- [32] Guruswamy, G. P. and Byun, C., "Fluid-Structural Interactions Using Navier-Stokes Flow Equations Coupled with Shell Finite Element Structures," *24th AIAA Fluid Dynamics Conference, Orlando, FL*, 1993.
- [33] Robinson, B. A., Yang, H. T. Y., and Batina, J. T., "Aeroelastic Analysis of Wings Using the Euler Equations with a Deforming Mesh," *Journal of Aircraft*, Vol. 28, No. 11, 1991, pp. 781–788.
- [34] Lee-Rausch, E. M. and Batina, J. T., "Wing Flutter Boundary Prediction Using Unsteady Euler Aerodynamic Method," *Journal of Aircraft*, Vol. 32, No. 2, March 1995, pp. 416–422.
- [35] Lee-Rausch, E. and Batina, J. T., "Wing Flutter Computations Using an Aerodynamic Model Based on the Navier-Stokes Equations," *Journal of Aircraft*, Vol. 33, No. 6, 1996, pp. 1139–1147.
- [36] Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1381–1388.

- [37] Hwang, G. Y. and Bendiksen, O. O., "Parallel Finite Element Solutions of Nonlinear Aeroelastic Problems in 3D Transonic Flow," *38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Kissimmee, FL*, April 1997.
- [38] Davis, G. A. and Bendiksen, O. O., "Transonic Panel Flutter," *34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, La Jolla, CA*, 1993.
- [39] Davis, G. A., *Transonic Aeroelasticity Solutions Using Finite Element in an Arbitrary Lagrangian-Eulerian Formulation*, Ph.d. dissertation, Mechanical and Aerospace Engineering Department, University of California, Los Angeles, 1994.
- [40] Hsiao, C. and Bendiksen, O. O., "Finite Element Euler Calculations of Unsteady Transonic Cascade Flows," *Proceedings of the Dynamics Specialist Conference*, 1992.
- [41] Hsiao, C. and Bendiksen, O. O., "Finite Element Calculations of Transonic Flutter in Cascades," *29th AIAA/SAE/ASME/ASEE Joint Propulsion Conference and Exhibit, Monterey, CA*, June 1993.
- [42] Hsiao, C., *Mixed Euler-Lagrangian Finite Element Solutions of Unsteady Flow and Flutter in Transonic Cascades*, Ph.d. dissertation, Mechanical and Aerospace Engineering Department, University of California, Los Angeles, 1993.
- [43] Schmitt, V. and Charpin, F., "Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers," *AGARD Advisory Report 138*, May 1979.
- [44] Bendiksen, O. O. and Hwang, G. Y., "Transonic Flutter Suppression Using Dynamic Twist Control," *37th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Salt Lake City, UT*, 1996.
- [45] Hwang, G. Y. and Bendiksen, O. O., "A Flutter Control Concept for Highly Flexible Transonic Wings," *38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Kissimmee, FL*, April 1997.
- [46] Hwang, G. Y., *Parallel Finite Element Solutions of Nonlinear Aeroelastic and Aeroservoelastic Problems in Three-Dimensional Transonic Flows*, Ph.d. dissertation, Mechanical and Aerospace Engineering Department, University of California, Los Angeles, March 1997.
- [47] Bendiksen, O. O., Hwang, G. Y., and Piersol, J., "Nonlinear Aeroelastic and Aeroservoelastic Calculations for Transonic Wings," *Meeting Paper Archive*, American Institute of Aeronautics and Astronautics, 1997.

- [48] Bendiksen, O. O., “Transonic Limit Cycle Flutter/LCO,” *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Palm Springs, CA*, April 2004.
- [49] Seber, G. and Bendiksen, O. O., “Nonlinear Flutter Calculations Using Finite Elements in a Direct Eulerian-Lagrangian Formulation,” *AIAA Journal*, Vol. 46, No. 6, 2008, pp. 1331–1341.
- [50] Bendiksen, O. O. and Seber, G., “Fluid-Structure Interactions with Both Structural and Fluid Nonlinearities,” *Journal of Sound and Vibration*, Vol. 315, No. 3, 2008, pp. 664–684.
- [51] Seber, G., *Aeroelastic Wing Response Analysis Using Finite Elements in a Large Deformation Direct Eulerian-Lagrangian Formulation*, Ph.D. thesis, Mechanical and Aerospace Engineering Department, University of California, Los Angeles, 2004.
- [52] Dietz, G., Schewe, G., Kießling, F., and Sinapius, M., “Limit-Cycle-Oscillation Experiments at a Transport Aircraft Wing Model,” *International Forum on Aeroelasticity and Structural Dynamics, Amsterdam*, 2003.
- [53] Bendiksen, O. O., “Transonic Limit Cycle Flutter of High-Aspect-Ratio Swept Wings,” *Journal of Aircraft*, Vol. 45, No. 5, 2008, pp. 1522–1533.
- [54] Bendiksen, O. O., “High-Altitude Limit Cycle Flutter of Transonic Wings,” *Journal of Aircraft*, Vol. 46, No. 1, 2009, pp. 123–136.
- [55] Bendiksen, O. O., “Influence of Shocks on Transonic Flutter of Flexible Wings,” *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Palm Springs, CA*, May 2009.
- [56] Bendiksen, O. O., “Transonic Stabilization Laws for Unsteady Aerodynamics and Flutter,” *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, HI*, April 2012.
- [57] Baker, M., Mendoza, R., and Hartwich, P., “Transonic Aeroelastic Analysis of a High Speed Transport Wind Tunnel Model,” *40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, St. Louis, MO*, 1999.
- [58] Schuster, D. M., Spain, C. V., Turnock, D. L., Rausch, R. D., Hamouda, M., Vogler, W. A., and Stockwell, A. E., “Development, Analysis and Testing of the High Speed Research Flexible Semispan Model,” Tech. Rep. CR-1999-209556, NASA, September 1999.
- [59] Schuster, D. M. and Rausch, R. D., “Transonic Dynamics Tunnel Force and Pressure Data Acquired on the HSR Rigid Semispan Model,” Tech. Rep. CR-1999-209555, NASA, September 1999.

- [60] Gordnier, R. E. and Melville, R. B., “Numerical Simulation of Large Amplitude Aeroelastic Wing Response,” *29th AIAA Fluid Dynamics Conference, Albuquerque, NM*, 1998.
- [61] Gordnier, R. E. and Melville, R. B., “Physical Mechanisms for Limit-Cycle Oscillations of a Cropped Delta Wing,” *30th AIAA Fluid Dynamics Conference, Norfolk, VA*, 1999.
- [62] Gordnier, R. E. and Melville, R. B., “Numerical Simulation of Limit-Cycle Oscillations of a Cropped Delta Wing Using the Full Navier-Stokes Equations,” *International Journal of Computational Fluid Dynamics*, Vol. 14, 2001, pp. 211–224.
- [63] Gordnier, R. E., “Computation of Limit-Cycle Oscillations of a Delta Wing,” *Journal of Aircraft*, Vol. 40, No. 6, 2003, pp. 1206–1207.
- [64] Attar, P. J. and Gordnier, R. E., “Aeroelastic Prediction of the Limit Cycle Oscillations of a Cropped Delta Wing,” *Journal of Fluids and Structures*, Vol. 22, No. 1, 2006, pp. 45–58.
- [65] Peng, C. and Han, J., “Numerical Investigation of the Effects of Structural Geometric and Material Nonlinearities on Limit-Cycle Oscillation of a Cropped Delta Wing,” *Journal of Fluids and Structures*, Vol. 27, No. 4, 2011, pp. 611–622.
- [66] Schairer, E. and Hand, L., “Measurements of Unsteady Aeroelastic Model Deformation by Stereo Photogrammetry,” *Journal of Aircraft*, Vol. 36, No. 6, 1999, pp. 1033–1040.
- [67] Hartwich, P., Dobbs, S., Arslan, A., and Kim, S., “Navier-Stokes Computations of Limit-Cycle Oscillations for a B-1-Like Configuration,” *Journal of Aircraft*, Vol. 38, No. 2, 2001, pp. 239–247.
- [68] Stevenson, J. R., “Shock-Induced Self-Excited Airfoil Bending Oscillations,” *Aerospace Flutter and Dynamics Council, Las Vegas, NV*, October 1978.
- [69] Dobbs, S. K., Miller, G. D., and Stevenson, J. R., “Self Induced Oscillation Wind Tunnel Test of a Variable Sweep Wing,” *26th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference, Orlando, FL*, April 1985.
- [70] Ericsson, L., “Comment on ‘Navier-Stokes Computations of Limit-Cycle Oscillations for a B-1-Like Configuration’,” *Journal of Aircraft*, Vol. 39, No. 3, 2002, pp. 519.
- [71] Ericsson, L. E., “Vortex-Induced Bending Oscillation of a Swept Wing,” *Journal of Aircraft*, Vol. 24, No. 3, 1987, pp. 195–202.

- [72] Machida, S., Kikuchi, T., Arizono, H., Saitoh, K., and Nakamichi, J., "Transonic Flutter Characteristics of an Arrow Wing Mounted with Engine," *43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada*, 2005.
- [73] Arizono, H., Machida, S., Kikuchi, T., Nakamichi, J., Inoue, T., and Akiba, K., "Flutter Wing Tunnel Testing with Scaled Model for SST," Tech. rep., JAXA, 2005.
- [74] Mavriplis, D. J., "Unstructured Grid Techniques," *Annual Review of Fluid Mechanics*, Vol. 29, No. 1, 1997, pp. 473–514.
- [75] Löhner, R., "Progress in Grid Generation via the Advancing Front Technique," *Engineering with Computers*, Vol. 12, No. 3-4, 1996, pp. 186–210.
- [76] Biedron, R. and Thomas, J., "Recent Enhancements to the FUN3D Flow Solver for Moving-Mesh Applications," *47th AIAA Aerospace Sciences Meeting, Orlando, FL*, Jan. 2009.
- [77] Frederick, C. O., Wong, Y. C., and Edge, F. W., "Two-Dimensional Automatic Mesh Generation for Structural Analysis," *International Journal for Numerical Methods in Engineering*, Vol. 2, No. 1, 1970, pp. 133–144.
- [78] Nguyen, V.-P., "Automatic Mesh Generation with Tetrahedron Elements," *International Journal for Numerical Methods in Engineering*, Vol. 18, No. 2, 1982, pp. 273–289.
- [79] Green, P. J. and Sibson, R., "Computing Dirichlet Tessellations in the Plane," *The Computer Journal*, Vol. 21, No. 2, 1978, pp. 168–173.
- [80] Bowyer, A., "Computing Dirichlet Tessellations," *The Computer Journal*, Vol. 24, No. 2, 1981, pp. 162–166.
- [81] Watson, D. F., "Computing the N-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes," *The Computer Journal*, Vol. 24, No. 2, 1981, pp. 167–172.
- [82] Sloan, S. and Houlby, G., "An Implementation of Watson's Algorithm for Computing 2-Dimensional Delaunay Triangulations," *Advances in Engineering Software*, Vol. 6, No. 4, 1984, pp. 192–197.
- [83] Lo, S. H., "A New Mesh Generation Scheme for Arbitrary Planar Domains," *International Journal for Numerical Methods in Engineering*, Vol. 21, No. 8, 1985, pp. 1403–1426.
- [84] Jameson, A., Baker, T., and Weatherill, N., "Calculation of Inviscid Transonic Flow Over a Complete Aircraft," *24th Aerospace Sciences Meeting, Reno, Nevada*, 1986.

- [85] Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., “Adaptive Remeshing for Compressible Flow Computations,” *Journal of Computational Physics*, Vol. 72, No. 2, 1987, pp. 449–466.
- [86] Löhner, R. and Parikh, P., “Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method,” *International Journal for Numerical Methods in Fluids*, Vol. 8, No. 10, 1988, pp. 1135–1149.
- [87] Löhner, R., “Some Useful Data Structures for the Generation of Unstructured Grids,” *Communications in Applied Numerical Methods*, Vol. 4, No. 1, 1988, pp. 123–135.
- [88] Gumbert, C., Lohner, R., Parikh, P., and Pirzadeh, S., “A Package for Unstructured Grid Generation and Finite Element Flow Solvers,” *7th AIAA Applied Aerodynamics Conference*, 1989, pp. 99–108.
- [89] Peraire, J., Peiro, J., and Morgan, K., “Multigrid Solution of the 3-D Compressible Euler Equations on Unstructured Tetrahedral Grids,” *International Journal for Numerical Methods in Engineering*, Vol. 36, No. 6, 1993, pp. 1029–1044.
- [90] Löhner, R., Camberos, J., and Merriam, M., “Parallel Unstructured Grid Generation,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 95, No. 3, 1992, pp. 343–357.
- [91] Kallinderis, Y., Khawaja, A., and McMorris, H., “Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flows Around Complex Geometries,” *AIAA Journal*, Vol. 34, No. 2, 1996, pp. 291–298.
- [92] Perry, B., Silva, W., Florance, J., Wieseman, C., Pototzky, A., Sanetrik, M., Scott, R., Keller, D., Cole, S., and Coulson, D., “Plans and Status of Wind-Tunnel Testing Employing an Aeroservoelastic Semispan Model,” *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, HI*, April 2007.
- [93] Löhner, R., “A Parallel Advancing Front Grid Generation Scheme,” *International Journal for Numerical Methods in Engineering*, Vol. 51, No. 6, 2001, pp. 663–678.
- [94] Ito, Y., Shih, A. M., Erukala, A. K., Soni, B. K., Chernikov, A., Chrisochoides, N. P., and Nakahashi, K., “Parallel Unstructured Mesh Generation by an Advancing Front Method,” *Mathematics and Computers in Simulation*, Vol. 75, No. 56, 2007, pp. 200–209.
- [95] Löhner, R., “A 2nd Generation Parallel Advancing Front Grid Generator,” *Proceedings of the 21st International Meshing Roundtable*, 2013, pp. 457–474.

- [96] Davis, G. A. and Bendiksen, O. O., “Unsteady Transonic Two-Dimensional Euler Solutions Using Finite Elements,” *AIAA Journal*, Vol. 31, No. 6, 1993, pp. 1051–1058.
- [97] Hirt, C., Amsden, A., and Cook, J., “An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds,” *Journal of Computational Physics*, Vol. 14, No. 3, 1974, pp. 227–253.
- [98] Donea, J., Giuliani, S., and Halleux, J., “An Arbitrary Lagrangian-Eulerian Finite Element Method for Transient Dynamic Fluid-Structure Interactions,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 33, No. 13, 1982, pp. 689–723.
- [99] Takashi, N. and Hughes, T. J., “An Arbitrary Lagrangian-Eulerian Finite Element Method for Interaction of Fluid and a Rigid Body,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 95, No. 1, 1992, pp. 115–138.
- [100] Pasiliao, C. L. and Dubben, J. A., “Analysis of CFD Pressure Coefficients on the F-16 Wing Associated with Limit Cycle Oscillations,” *46th Aerospace Sciences Meeting, Reno, NV*, January 2008.
- [101] Longatte, E., Verreman, V., and Souli, M., “Time Marching for Simulation of Fluid-Structure Interaction Problems,” *Journal of Fluids and Structures*, Vol. 25, No. 1, 2009, pp. 95–111.
- [102] Beran, P. S., Lucia, D. J., and Pettit, C. L., “Reduced-Order Modelling of Limit-Cycle Oscillation for Aeroelastic Systems,” *Journal of Fluids and Structures*, Vol. 19, No. 5, 2004, pp. 575–590.
- [103] Byun, C., Guruswamy, G., and Kutler, P., “Wing-Body Aeroelasticity Using Finite-Difference Fluid/Finite-Element Structural Equations on Parallel Computers,” *35th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Hilton Head, SC*, 1994.
- [104] Dowell, E., Thomas, J., and Hall, K., “Transonic Limit Cycle Oscillation Analysis Using Reduced Order Aerodynamic Models,” *Journal of Fluids and Structures*, Vol. 19, No. 1, 2004, pp. 17–27.
- [105] Jameson, A., Schmidt, W., and Turkel, E., “Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge Kutta Time Stepping Schemes,” *14th AIAA Fluid and Plasma Dynamics Conference, Palo Alto, CA*, June 1981.
- [106] Jameson, A. and Mavriplis, D., “Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh,” *AIAA Journal*, Vol. 24, 1986, pp. 611–618.

- [107] Mavriplis, D., “Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes,” *AIAA Journal*, Vol. 26, 1988, pp. 824–831.
- [108] Mavriplis, D. J., “Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes,” *AIAA Journal*, Vol. 28, No. 2, 1990, pp. 213–221.
- [109] Tang, L., Bartels, R. E., Chen, P. C., and Liu, D. D., “Numerical Investigation of Transonic Limit Cycle Oscillations of a Two-Dimensional Supercritical Wing,” *Journal of Fluids and Structures*, Vol. 17, No. 1, 2003, pp. 29–41.
- [110] Thomas, J., Dowell, E., and Hall, K., “Nonlinear Inviscid Aerodynamic Effects on Transonic Divergence, Flutter, and Limit-Cycle Oscillations,” *AIAA Journal*, Vol. 40, No. 4, April 2002, pp. 638–646.
- [111] Zhang, X. and Lai, K. L., “Simulation of Transonic Aeroservoelasticity Using Cartesian-Grid Based Flow Solver,” *42nd AIAA Fluid Dynamics Conference and Exhibit, New Orleans, LA*, June 2012.
- [112] Jin, H. and Tanner, R. I., “Generation of Unstructured Tetrahedral Meshes by Advancing Front Technique,” *International Journal for Numerical Methods in Engineering*, Vol. 36, No. 11, 1993, pp. 1805–1823.
- [113] Batina, J. T., “Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis,” *AIAA Journal*, Vol. 29, No. 3, 1991, pp. 327–333.
- [114] Shankaran, S. and Jameson, A., “Aerodynamic and Aeroelastic Applications of a Parallel, Multigrid, Unstructured Flow Solver,” *The 41st AIAA Aerospace Science Meeting and Exhibit, Reno, NV*, 2003.
- [115] “Python Programming Language – Official Website,” <http://python.org>, 2013, [Online; accessed 26-August-2013].
- [116] Doggett Jr, R. V., Morgan, H. G., and Rainey, A. G., “An Experimental Investigation of Aerodynamic Effects of Airfoil Thickness on Transonic Flutter Characteristics,” *NASA Technical Memorandum X-79*, 1959.
- [117] Bathe, K. J., Ramm, E., and Wilson, E. L., “Finite Element Formulations for Large Deformation Dynamic Analysis,” *International Journal for Numerical Methods in Engineering*, Vol. 9, No. 2, 1975, pp. 353–386.
- [118] Attar, P. J., Dowell, E. H., and White, J. R., “Modeling the LCO of a Delta Wing Using a High Fidelity Structural Model,” *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Palm Springs, CA*, 2004.

- [119] Attar, P. J., Dowell, E. H., and White, J. R., “Modeling Delta Wing Limit-Cycle Oscillations Using a High-Fidelity Structural Model,” *Journal of Aircraft*, Vol. 42, No. 5, 2005, pp. 1209–1217.
- [120] Relvas, A. and Suleman, A., “Fluid-Structure Interaction Modelling of Non-linear Aeroelastic Structures Using the Finite Element Corotational Theory,” *Journal of Fluids and Structures*, Vol. 22, No. 1, 2006, pp. 59–75.
- [121] Attar, P. J., Dowell, E. H., and Tang, D., “Delta Wing with Store Limit-Cycle-Oscillation Modeling Using a High-Fidelity Structural Model,” *Journal of Aircraft*, Vol. 45, No. 3, 2008, pp. 1054–1061.
- [122] Batoz, J. L. and Lardeur, P., “A Discrete Shear Triangular Nine D.O.F. Element for the Analysis of Thick to Very Thin Plates,” *International Journal for Numerical Methods in Engineering*, Vol. 28, No. 3, 1989, pp. 533–560.
- [123] Patel, S., *A Discrete Shear Triangular Finite Element with Applications to the Vibration Analysis of Thin and Thick Plates*, Master’s thesis, University of California, Los Angeles, 1995.
- [124] Turner, M. J., Clough, R. W., Martin, H. C., and Topp, L. J., “Stiffness and Deflection Analysis of Complex Structures,” *Journal of Aeronautical Science*, Vol. 23, No. 9, 1956, pp. 805–823.
- [125] Rajasekaran, S. and Murray, D. W., “Incremental Finite Element Matrices,” *Journal of the Structural Division*, Vol. 99, No. 12, 1973, pp. 2423–2438.
- [126] Bendiksen, O. O., “Transonic Single-Degree-of-Freedom Flutter and Natural Mode Instabilities,” *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Boston, MA*, April 2013.
- [127] Mavriplis, D., “Three-Dimensional Unstructured Multigrid for the Euler Equations,” *AIAA Journal*, Vol. 30, No. 7, 1992, pp. 1753–1761.
- [128] MSC Software Corporation, Santa Ana, CA, *MSC Nastran 2013 Dynamic Analysis User’s Guide*.
- [129] “Intel Parallel Studio XE 2013,” <http://software.intel.com/en-us/intel-parallel-studio-xe/>, 2013, [Online; accessed 18-August-2013].
- [130] “Intel Math Kernel Library 11.0,” <http://software.intel.com/en-us/intel-mkl>, 2013, [Online; accessed 18-August-2013].
- [131] “VTK - The Visualization Toolkit,” <http://vtk.org>, 2013, [Online; accessed 18-August-2013].