Transonic Flutter Prediction and Aeroelastic Tailoring for Next-Generation Transport Aircraft

by

Max Maria Jacques Opgenoord

B.Sc. Aerospace Engineering Delft University of Technology (2014)
S.M. Aeronautics and Astronautics Massachusetts Institute of Technology (2016)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

at the

Massachusetts Institute of Technology

September 2018

© Massachusetts Institute of Technology 2018. All rights reserved.

Author
Max M. J. Opgenoord
Department of Aeronautics and Astronautics
August 23, 2018

Certified by
Karen E. Willcox
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by
Mark Drela
Terry J. Kohler Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by
David L. Darmofal
Professor of Aeronautics and Astronautics
Thesis Committee Member

Accepted by
Hamsa Balakrishnan
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee
Transonic Flutter Prediction and\nAeroelastic Tailoring for\nNext-Generation Transport Aircraft

by

Max M. J. Opgenoord

Submitted to the Department of\nAeronautics and Astronautics on August 23, 2018,\nin Partial Fulfillment of the Requirements for the Degree of\nDoctor of Philosophy in Aerospace Computational Engineering

ABSTRACT

Novel commercial transport aircraft concepts feature large wing spans to increase their fuel efficiency; these wings are more flexible, leading to more potential aeroelastic problems. Furthermore, these aircraft fly in the transonic flow regime, where flutter prediction is difficult. The goals for this thesis are to devise a method to reduce the computational burden of including transonic flutter constraints in conceptual design tools, and to offer a potential solution for mitigating flutter problems through the use of additive manufacturing techniques, specifically focusing on a design methodology for lattice structures.

To reduce the computational expense of considering transonic flutter in conceptual aircraft design, a physics-based low-order method for transonic flutter prediction is developed, which is based on small unsteady disturbances about a known steady flow solution. The states of the model are the circulation and doublet perturbations, and their evolution equation coefficients are calibrated using off-line unsteady two-dimensional flow simulations. The model is formulated for swept high-aspect ratio wings through strip theory and 3D corrections. The resulting low-order unsteady flow model is coupled to a typical-section structural model (for airfoils) or a beam model (for wings) to accurately predict flutter of airfoils and wings. The method is fast enough to permit incorporation of transonic flutter constraints in conceptual aircraft design calculations, as it only involves solving for the eigenvalues of small state-space systems. This model is used to describe the influence of transonic flutter on next-generation aircraft configurations, where it was found that transonic flutter constraints can limit the efficiency gains seen by better material technology.
As a potential approach for mitigating flutter, additively manufactured lattice structures are aeroelastically tailored to increase the flutter margin of wings. Adaptive meshing techniques are used to design the topology of the lattice to align with the load direction while adhering to manufacturing constraints, and the lattice is optimized to minimize the structural weight and to improve the flutter margin. The internal structure of a wing is aeroelastically tailored using this design strategy to increase the flutter margin, which only adds minimal weight to the structure due to the large design freedom the lattice structure offers.

*Thesis Supervisor:*
Karen E. Willcox  
Professor of Aeronautics and Astronautics

*Thesis Supervisor:*
Mark Drela  
Terry J. Kohler Professor of Aeronautics and Astronautics
This thesis is dedicated to my brother

Bram Opgenoord
PhD? Not for me. At least that is what yours truly thought back in 2014. And yet here we are. In fact, I have never felt at home anywhere quite as much as I have at MIT these past four years, and am glad I stuck around after my Masters. This may not have been the case if it were not for the many wonderful people that have supported me and my research during this time.

I really hit the jackpot with Professor Karen Willcox as my advisor. She welcomed me into her research group as a 21-year-old with more ambitions than skills, and gave me the support and trust to make the most out of my time here. She always asks the right questions and always provides helpful research suggestions. She also gave me the freedom to pursue side projects in research or as part of design teams, which was a huge part of my life here. Moreover, Karen deeply cares for all the students and postdocs in her group, focusing on our well-being and our professional growth. Thank you for everything—I will miss working with you.

It has also been a great honor to work with Professor Mark Drela, whose research I have admired ever since starting my undergraduate degree. His physical intuition and clarity in presenting difficult ideas is as astonishing as it is intimidating. Thank you for all the lessons in aerodynamics and aeroelasticity, and for the interesting discussions on vehicle designs.

To my third committee member, Professor David Darmofal, thank you for your helpful suggestions and for taking an interest in my work. I am excited for the work that is coming out of your research group, especially as SANS nears completion. On that note, I also owe Dr. Marshall Galbraith for the help with SANS; it is unfortunate the timing did not work out for me to use it in this research.

I am indebted to Dr. Frode Engelsen and Professor John Hart for taking time out of their busy schedules to serve as my thesis readers. Dr. Engelsen has also been kind enough to discuss the industry perspective on aeroelastic design on numerous occasions, which has been quite helpful in shaping this work.

Besides my thesis committee and readers, several people have made valuable contributions to my research. Valentin Churavy and Professor Alan Edelman are thanked for the julia help; Dr. Thomas Economon and Professor Juan Alonso for answering my many questions about SU2;
Bob Haimes for all the geometry help, and for being Bob; Professor Wesley Harris for the helpful discussions on potential flow theory and aeroelasticity; Philip Caplan,† for the valuable insights in computational geometry and Riemannian metric fields; and Dr. Boris Krämer for the enlightening discussions on reduced-order modeling. Finally, Jean Sofronas, Beth Marois, Robin Courchesne-Sato, Meghan Pepin, Melanie Burliss, Joyce Light, Bill Litant, and Anthony Zolnik are thanked for their help in tracking down professors, fighting the MIT bureaucracy, helping us communicate our research to the rest of the world, and making sure we can work without ceiling tiles dropping on our heads.

The MIT MAKERWORKSHOP has also been instrumental in completing this research, as I was able to manufacture several test parts for my thesis research there. Thank you to all the people running that shop—it has been wonderful to be a part of such a talented maker community as a mentor.

Finally, the support by the NASA LEARN program—grant number NNX14AC73A—and the SUTD-MIT International Design Center (IDC) is gratefully acknowledged.

While support in the actual research is quite important in completing a doctoral degree, the support from friends and family is equally so.

A major contributor to my positive graduate school experience has been the ACDEL. I fell with my nose in the butter‡ by having such great cubemates in Hugh, Cory, and Nisha.† Thank you for all the banter, late night philosophical discussions, and for being a research sounding board—but mostly the banter. Part of my last semester was spent in Singapore with Alex and Michael—thank you both for the good times over there. A big thank you also to the other ACDEL members (current and former) for the laughter, the paintball fights, and sharing in the misery of the (almost) windowless office and the Building 31 construction: Philippe, Phil, Ferran, Victor, Patrick, Ben, Boris, Anirban, Brian, Andy, Rémi, and Shun.

Being part of the MIT HYPERLOOP TEAM was one of the most fulfilling experiences of my time here. It was a privilege to run the team together with John, Chris, and Philippe. And Philippe, thank you for dragging me into that mess. To the rest of the Hyperloop team, thank you for the good times and some of the best memories‡ of my time at MIT, especially the Aero/Structures and Whitecap folks.

Outside the lab, a big thank you to my friends around campus, in particular Alex, Sarthak, Akil, and Parker, for dragging me out of the lab every once in a while for races, movie nights, weekend getaways, or just a nice lunch. I am also grateful to my friends from Delft for keeping in touch, in particular Constantin, Sascha, and Dawud.

I count myself lucky to still have close friends from elementary and

†These people get an extra thank you for voluntarily proofreading this thesis, and/or helping me prepare for my thesis defense.

‡You are correct, this is not an English expression.

‡And also some of the worst memories, the brakes still give me nightmares.
high school as well—whenever I am home, it is like I never left. They always keep in touch, they go out of their way to see me when I am back, and some have even visited here. For all of that and so much more, thank you Jamie, Miriam, Tom, Rob, Wesley, and the rest of the Trevianum gang.

Fulfilling my dreams meant moving away from people I deeply care about—I hope they will some day forgive me for that. I cannot thank my cousin Brigitte enough for always looking out for me as the big sister I never had. Knowing that you and Jack were looking after Mom and Dad gave me the peace of mind to focus on my studies. The weekly calls with you and my wonderful goddaughter Jill—as well as Jack, Youri, and Jorn—were always a highlight of my week, and always cheered me up whenever I needed it.

One could spend a lifetime searching and never find more loving and caring parents than my mother and father. They have always supported me, encouraged me, and been my greatest champions in whatever enterprise I pursued, even when that resulted in the occasional broken wind shield. Furthermore, Mom's creativity and Dad's engineering wits shaped me as an engineer and have made a large impact on this thesis. I cannot thank you both enough for everything you have done for me.

Finally, my brother Bram showed me what true strength and perseverance is. Bram is my hero, and to him I dedicate this thesis.

Max Opgenoord
Cambridge, Massachusetts
August 2018
CONTENTS

Abstract 3
Nomenclature 13
Acronyms 17
List of Figures 19
List of Tables 23

1 Introduction 25
  1.1 Transonic Flutter 26
  1.2 Design for Additive Manufacturing 29
  1.3 Aeroelastic Tailoring 33
  1.4 Thesis Goals & Outline 35

2 Physics-Based Low-Order Model for Transonic Flutter Prediction 37
  2.1 Physics-Based State-Space Model Formulation 37
  2.2 Model Fitting from CFD Data 43
  2.3 Validation of Airfoil Flutter Model 48
  2.4 Influence of Airfoil Parameters on Flutter 54

3 Transonic Flutter Prediction for Aircraft Wings 57
  3.1 Wing Flutter Model 57
  3.2 Validation of Wing Flutter Model 63
  3.3 Conceptual Aircraft Design Tool 66
  3.4 Influence of Wing Geometry on Flutter 68
  3.5 Influence of Flutter on Aircraft Designs 75

4 Design of Lattice Structures for Additive Manufacturing 79
  4.1 Design of Lattice Topology 79
  4.2 Optimal Cross-Sectional Area of Lattice Struts 87
  4.3 Example Designs 90

5 Aeroelastic Tailoring using Lattice Structures 97
  5.1 Design for Flutter Mitigation 97
  5.2 Verification of Low-Order Structural Model 103
  5.3 Tailoring of Wing in Transonic Flow 106
6 Conclusions and Outlook 111
  6.1 Contributions 111
  6.2 Opportunities for Further Research 114
  6.3 Potential Additional Applications 118

Bibliography 121

Appendices

A Flutter Boundary Computation 145

B Finite Element Solver for Linear Elasticity 147
  B.1 Discretization 147
  B.2 Verification 150

C Geometry Generation for Lattice Structures 153
  C.1 Solid Representation of Lattice Structures 153
  C.2 CAD Generation 159
  C.3 Slicing of Lattice Structures for Printing 160

D Adjoint for Low-Order Structural Model of Lattice Structures 163
  D.1 Adjoints for Elementary Matrix Operations 163
  D.2 Approach for Structural Model 167

E Code Listings 169

Index 171
# NOMENCLATURE

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A, E$</td>
<td>State-space matrices</td>
</tr>
<tr>
<td>$\mathcal{R}$</td>
<td>Aspect ratio</td>
</tr>
<tr>
<td>$A_a, B_a$</td>
<td>State-space matrices of aerodynamic system</td>
</tr>
<tr>
<td>$A_\Gamma, A_\Gamma', A_\kappa$</td>
<td>Coefficients of $\Gamma$ evolution equation</td>
</tr>
<tr>
<td>$B$</td>
<td>Strut deformation matrix of lattice</td>
</tr>
<tr>
<td>$B_\Gamma, B_\kappa, B_\kappa'$</td>
<td>Coefficients of $\kappa_x$ evolution equation</td>
</tr>
<tr>
<td>$C$</td>
<td>Connectivity matrix of lattice nodes and struts</td>
</tr>
<tr>
<td>$\mathcal{C}$</td>
<td>Fourth-order stiffness tensor</td>
</tr>
<tr>
<td>$C_{th}(\bar{\omega})$</td>
<td>Theodorsen's function</td>
</tr>
<tr>
<td>$D_{wave}$</td>
<td>Wave drag</td>
</tr>
<tr>
<td>$E$</td>
<td>Young's modulus</td>
</tr>
<tr>
<td>$EI$</td>
<td>Bending stiffness</td>
</tr>
<tr>
<td>$F$</td>
<td>Nodal forces on lattice</td>
</tr>
<tr>
<td>$GJ$</td>
<td>Torsional stiffness</td>
</tr>
<tr>
<td>$H$</td>
<td>Angular momentum</td>
</tr>
<tr>
<td>$I_\bar{\gamma}$</td>
<td>Mass moment of inertia per span around elastic axis</td>
</tr>
<tr>
<td>$I_\theta$</td>
<td>Moment of inertia around elastic axis</td>
</tr>
<tr>
<td>$\mathcal{I}_\bar{\gamma}$</td>
<td>Low-order matrix for mass moment of inertia per span</td>
</tr>
<tr>
<td>$K$</td>
<td>Stiffness matrix of lattice</td>
</tr>
<tr>
<td>$L$</td>
<td>Lift per span</td>
</tr>
<tr>
<td>$L_c$</td>
<td>Circulatory lift</td>
</tr>
<tr>
<td>$L_{nc}$</td>
<td>Non-circulatory lift</td>
</tr>
<tr>
<td>$M$</td>
<td>Mass matrix</td>
</tr>
<tr>
<td>$\mathcal{M}$</td>
<td>Low-order matrix for mass per span</td>
</tr>
<tr>
<td>$M_{crit}$</td>
<td>Critical Mach number</td>
</tr>
<tr>
<td>$M_{\infty}$</td>
<td>Freestream Mach number</td>
</tr>
<tr>
<td>$M_{CR}$</td>
<td>Cruise Mach number</td>
</tr>
<tr>
<td>$M_{c/2}$</td>
<td>Aerodynamic moment per span around mid-chord</td>
</tr>
<tr>
<td>$M_{ea}$</td>
<td>Aerodynamic moment per span around elastic axis</td>
</tr>
</tbody>
</table>
\( M_{nc} \) Non-circulatory moment around the elastic axis  
\( \mathcal{M}(x) \) Metric  
\( \mathcal{M}(y) \) Internal bending moment  
\( Re \) Reynolds number  
\( S \) Internal shear force  
\( S_y \) Mass unbalance per span  
\( S_\theta \) Mass unbalance  
\( S_y \) Low-order matrix for mass unbalance per span  
\( St \) Strouhal number  
\( T_{(\cdot)} \rightarrow (\cdot) \) Map between low-order and full-order lattice model  
\( T \) Internal torsion moment  
\( V \) Velocity field  
\( \mathcal{V} \) Volume  
\( V_0 \) Velocity field of steady solution  
\( V_\mu \) Flutter speed index  
\( V_\infty \) Freestream velocity  
\( V_L \) Wing-perpendicular velocity  
\( V_{\text{cruise}} \) Cruise speed  
\( V_f \) Flutter speed  
\( W_{\text{MTO}} \) Maximum take-off weight  
\( X, X', X'' \) Matrices containing sequences of snapshots of state vector  
\( a \) Acceleration vector  
\( a, a_1, b \) Discrete state-space matrices  
\( a_{ik} \) Cross-sectional area of strut connecting node \( i \) and \( k \)  
\( b \) Wing span  
\( c \) Airfoil chord length  
\( c_\ell \) Lift coefficient  
\( c_{\ell_0} \) Baseline sectional lift coefficient  
\( d \) Distance between quarter-chord and elastic axis  
\( f \) Body force per unit volume  
\( f_{\text{crit}} \) Euler critical buckling load  
\( h \) Vertical displacement, positive downward  
\( h_1, h_2, h_3 \) Length of major axes of metric  
\( h_a \) Altitude  
\( k_e \) Column effective length factor
$k_h$  Spring constant for vertical motion

$k_{\theta}$  Torsional spring constant

$\bar{l}$  Effective wing length

$l_{ik}$  Length of strut connecting node $i$ and $k$

$\ell_M$  Edge length under Riemannian metric

$m$  Mass per span

$m_{\text{wing}}$  Wing mass

$n$  Normal vector

$n_1, n_2, n_3$  Normal vector of major axes of metric

$n_{\text{printer}}$  $z$-axis of printer

$p$  Polynomial order

$p_y$  Parameterized span loading

$q$  Unit quaternions

$r$  Position vector

$r_{\theta}$  Radius of gyration per span about the elastic axis

$t$  Time

$u$  Displacement vector

$u_a$  State-space control vector of aerodynamic system

$w$  Vertical (downward) velocity

$x$  State vector

$x, y, z$  Physical coordinates

$\bar{x}, \bar{y}, \bar{z}$  Coordinates aligned with swept wing

$x_a$  State vector of aerodynamic system

$x_{cg}$  Position of center of gravity relative to wing leading edge

$x_{ca}$  Position of elastic axis relative to wing leading edge

$\Gamma$  Circulation strength

$\Lambda$  Sweep angle

$\Upsilon'$  Matrix containing sequence of snapshots of control input vector

$\alpha$  Angle of attack

$\beta$  Angle between printer $z$-axis and part $z$-axis

$\beta_{\text{max}}$  Maximum overhang angle

$\gamma$  Local dihedral angle

$\varepsilon$  Strain tensor

$\eta_{\text{eng}}$  Nondimensional spanwise location of engine
\( \theta \)  
Pitch angle

\( \kappa_x \)  
\( x \)-doublet strength

\( \lambda \)  
Taper ratio

\( \lambda_1 \)  
Lamé's first parameter

\( \mu \)  
Apparent mass ratio

\( \mu_s \)  
Shear modulus

\( \nu \)  
Poisson's ratio

\( \xi_{\text{eng},x}, \xi_{\text{eng},z} \)  
Nondimensional chordwise location of engine

\( \rho_\infty \)  
Freestream density

\( \sigma \)  
Volume-source strength

\( \sigma_C \)  
Maximum compressive stress

\( \sigma_i \)  
Stress in \( i \)th strut of lattice

\( \sigma_1, \sigma_2, \sigma_3 \)  
Principal stresses

\( \mathbf{\sigma} \)  
Cauchy stress tensor

\( \sigma_T \)  
Maximum tensile stress

\( \tau \)  
Thickness ratio of airfoil

\( \phi \)  
Velocity potential

\( \varphi \)  
Basis function

\( \chi_{\text{fil}} \)  
Maximum real eigenvalue of aeroelastic system

\( \omega \)  
Angular velocity

\( \mathbf{\omega} \)  
Vorticity

\( \bar{\omega} \)  
Reduced frequency

\( \omega_\theta, \omega_h \)  
Uncoupled in-vacuo natural frequencies in pitch and heave
ACRONYMS

ABS Acrylonitrile Butadiene Styrene. 90

AD Automatic Differentiation. 163

AM Additive Manufacturing. 84, 159

BSCW Benchmark Supercritical Wing. 64, 65


CFD Computational Fluid Dynamics. 26, 29, 37, 43, 45, 46, 49, 50, 65, 66, 81, 97, 111–113, 145

DED Direct Energy Deposition. 31

DG Discontinuous Galerkin. 147

DMD Dynamic Mode Decomposition. 23, 44, 45, 49–52, 145

EBM Electron Beam Melting. 31

EDG Embedded Discontinuous Galerkin. 148

FEM Finite Element Method. 22, 82, 98, 119, 147, 148, 172

FFF Fused Filament Fabrication. 31, 32, 91, 92

GE General Electric. 93, 117

HDG Hybridizable Discontinuous Galerkin. 82, 147–150, 169

MDO Multi-Disciplinary Design Optimization. 27

NASA National Aeronautics and Space Administration. 8, 20, 25, 64–66

NLP Nonlinear Programming. 103

PAPA Pitch and Plunge Apparatus. 65

PDE Partial Differential Equation. 46

RANS Reynolds Averaged Navier-Stokes. 42, 66
**SIMP** Solid Isotropic Material with Penalization. 30, 34, 81

**SLA** Stereolithography. 31, 32

**SLM** Selective Laser Melting. 31, 32

**SLS** Selective Laser Sintering. 31, 32

**SMC** Standard Mean Chord. 68

**SPD** Symmetric Positive Definite. 81, 82

**STL** Standard Tessellation Language. 119, 159

**SVD** Singular Value Decomposition. 45

**TASOPT** Transport Aircraft System OPTimization. 66–68, 72, 73, 75

**TSD** Transonic Small-Disturbance. 28

**UAV** Unmanned Aerial Vehicle. 117, 120

**URANS** Unsteady Reynolds Averaged Navier-Stokes. 114

**UV** Ultraviolet. 31
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Information travel for airfoil in transonic flow.</td>
<td>27</td>
</tr>
<tr>
<td>1.2</td>
<td>Typical transonic flutter boundary.</td>
<td>28</td>
</tr>
<tr>
<td>1.3</td>
<td>Overview of additive manufacturing techniques.</td>
<td>32</td>
</tr>
<tr>
<td>1.4</td>
<td>Influence of manufacturing constraints on optimality of part.</td>
<td>33</td>
</tr>
<tr>
<td>2.1</td>
<td>Summary of transonic airfoil flutter model construction.</td>
<td>37</td>
</tr>
<tr>
<td>2.2</td>
<td>Vorticity and field sources representing velocity field are approximated by overall circulation $\Gamma$ and doublet $\kappa_x$.</td>
<td>39</td>
</tr>
<tr>
<td>2.3</td>
<td>Variation of $\kappa_{x,0}$ with $c_{\epsilon_0}$ and $M_{\infty}$ for the RAE2822 airfoil.</td>
<td>40</td>
</tr>
<tr>
<td>2.4</td>
<td>Typical section model.</td>
<td>42</td>
</tr>
<tr>
<td>2.5</td>
<td>Details of meshes to illustrate how the circulation and $x$-doublet are computed through line integrals.</td>
<td>46</td>
</tr>
<tr>
<td>2.6</td>
<td>Volume-source fields around RAE2822 airfoil in subsonic and transonic flow.</td>
<td>47</td>
</tr>
<tr>
<td>2.7</td>
<td>Example of ramped forced pitching oscillation.</td>
<td>48</td>
</tr>
<tr>
<td>2.8</td>
<td>Phase diagram of circulation strength $\Gamma$ for forced heaving at different reduced frequencies for subsonic and transonic flow.</td>
<td>49</td>
</tr>
<tr>
<td>2.9</td>
<td>Phase diagram of $x$-doublet strength $\kappa_x$ for forced heaving at different reduced frequencies for subsonic and transonic flow.</td>
<td>49</td>
</tr>
<tr>
<td>2.10</td>
<td>Comparison of phase diagrams between SU2 results and low-order model for forced pitching oscillations at $\bar{\omega} = 0.35$ for $M_{\infty} = 0.6$ for the RAE2822 airfoil.</td>
<td>50</td>
</tr>
<tr>
<td>2.11</td>
<td>Comparison of time series between SU2 results and low-order model for forced pitching oscillations</td>
<td>50</td>
</tr>
<tr>
<td>2.12</td>
<td>Comparison of phase diagrams between SU2 results and low-order model for forced heaving oscillations at $\bar{\omega} = 0.25$ for $M_{\infty} = 0.6$ for the RAE2822 airfoil.</td>
<td>51</td>
</tr>
<tr>
<td>2.13</td>
<td>Bode plot of transfer function between circulation strength and pitch angle for different Mach numbers for the RAE2822 airfoil.</td>
<td>52</td>
</tr>
<tr>
<td>2.14</td>
<td>Flutter speed versus $x_{cg}/c$ positions for different freestream Mach numbers.</td>
<td>53</td>
</tr>
</tbody>
</table>
2.15 Comparison of transonic flutter boundary for Isogai Case A between our method and methods from literature. 54
2.16 Influence of baseline lift coefficient $c_{\ell_0}$ on the transonic flutter boundary for the 11\% C-airfoil. 54
2.17 Overview of C-airfoil family, used in the design of the D8.x aircraft. 55
2.18 Influence of airfoil thickness on the transonic flutter boundary for the C-airfoil family at $c_{\ell_0} = 0.6$. 55
2.19 Comparison between C-airfoil and E-airfoil (12\%). 55
2.20 Comparison of flutter boundary between the E-airfoil and C-airfoil, both 12\% thick at $c_{\ell_0} = 0.6$. 56

3.1 Assumptions in strip theory. 58
3.2 Swept wing considered in model. 58
3.3 Discretized beam model used in flutter model. 60
3.4 Wing box cross section. 61
3.5 Inclusion of engine in beam model. 63
3.6 Mach number field around NASA SC(2)-0414 airfoil at $M_\infty = 0.74$ and $\alpha = 0^\circ$. 64
3.7 Sectional lift coefficient perturbation at 60\% semi-span versus pitch angle for a wing with a NASA SC(2)-0414 airfoil and $AR = 8$. 64
3.8 Benchmark Supercritical Wing in wind tunnel. 65
3.9 Comparison between flutter predictions from experiments, literature and our method for the Benchmark Supercritical Wing. 65
3.10 Flutter coefficient database for 11\% thick C-airfoil. 67
3.11 Geometry for clamped straight swept tapered wing. 68
3.12 Influence of taper ratio on the flutter boundary. 69
3.13 Influence of aspect ratio and freestream Mach number on the flutter boundary. 70
3.14 Influence of wing sweep angle on the flutter boundary. 71
3.15 Influence of engine weight and freestream Mach number on the flutter boundary. 72
3.16 Comparison of D8.0 and D8.2 configurations. 73
3.17 $(M_\infty, h_a)$ envelope for the D8.2 aircraft. 73
3.18 Influence of aspect ratio $AR$ on maximum aeroelastic eigenvalue $\chi_{fl}$ for two different aircraft. 74
3.19 Influence of wing sweep angle $\Lambda$ on maximum aeroelastic eigenvalue $\chi_{fl}$ for two different aircraft. 75
3.20 Advanced-technology D8.0 designed with and without flutter constraints. 75
3.21 Fuel burn increase as a result of transonic flutter constraints for different design cruise Mach numbers. 76
3.22 Variations of optimum fuel burn and maximum take-off weight with specific allowable stress. 77

4.1 Approach for designing a manufacturable load-dependent lattice. 80
4.2 Lattice structure and continuous fields of its properties. 80
4.3 Geometric interpretation of a metric. 81
4.4 Example of a metric field, illustrated using ellipses. 82
4.5 The stress tensor is not necessarily positive definite, but an updated metric for adaptation purposes can be made positive definite by taking the absolute value of the principle stresses. 83
4.6 Aligning the metric with the stress direction. 83
4.7 Overhang angle constraints and bridge length manufacturing constraints. 84
4.8 Geometry of optimization problem for including manufacturing constraints in the local metric. 85
4.9 2D example of including manufacturing constraints by optimizing the local metric. 86
4.10 The lattice is aligned with the printer direction for this column under compression. 87
4.11 The manufacturability of the lattice topology can be improved through edge swaps and by adding internal support. 87
4.12 Truss structure optimization. 88
4.13 The matrix $C$ defines the connectivity between the struts and nodes of the lattice. 89
4.14 Load case for bracket design example. 90
4.15 Von Mises stress through domain of bracket. 91
4.16 Comparison of optimized anisotropic lattice and isotropic lattice. 91
4.17 Printed lattice structure. 92
4.18 Load case for the aircraft bracket. 93
4.19 Von Mises stress through domain of $\theta \in$ bracket. 94
4.20 Optimized lattice structure for aircraft bracket. 95

5.1 The full-order dynamic lattice model is reduced to a low-order lattice model to be used in the lattice optimization. 98
5.2 Mass matrix generation. 99
5.3 Mapping the deflections from the low-order beam model to the full lattice model. 100
5.4 Lattice with 136 nodes and 535 struts used for validation. 104
5.5 Comparison between second bending mode of full-order lattice model and low-order beam model. 104
5.6 Comparison between first torsion mode of full-order lattice model and low-order beam model. 105
5.7 Asymmetric lattice with 126 nodes and 489 struts used for validation of lattice structures. 105
5.8 Von Mises stress throughout design space for wing in transonic flow. 106
5.9 Original lattice that is optimized without considering aeroelastic instabilities. 106
5.10 Flutter boundary for both the original lattice and the aeroelastically tailored lattice. 107
5.11 Final aeroelastically tailored lattice. 107
5.12 Eigenmodes for optimized wing lattice. 108
A.1 Illustration of using the bisection method to find the flutter speed at $M_\infty = 0.65$. 146
B.1 Comparison of four different FEM discretizations. 148
B.2 $L_2$ solution error convergence for linear elasticity problem in 2D. 151
B.3 $L_2$ solution error convergence for linear elasticity problem in 3D. 152
C.1 Lattice representation with 6 nodes and 13 struts. 154
C.2 Example of solid representation where the struts are tessellated and the nodes are generated using convex hulls. 155
C.3 Cutting planes and intersecting vertices on a node. 156
C.4 Active vertices for intersections of strut 1. 156
C.5 Projection of points on strut onto nodes. 157
C.6 Tapering of strut geometry. 157
C.7 Flat base lattice structures. 158
C.8 Geometry of struts connecting the nodes. 158
C.9 Tessellation of struts. 159
C.10 Examples of lattices generated using the geometry generation algorithm. 159
C.11 Threaded hole in parametric CAD representation of lattice structure. 160
C.12 Slicing the lattice frame representation directly. 161
LIST OF TABLES

2.1 Accuracy of DMD fit for forced pitching oscillations for RAE2822 airfoil. 51
2.2 Accuracy of DMD fit for forced pitching oscillations at frequencies that were not part of the DMD calibration for RAE2822 airfoil. 52
2.3 Airfoil parameters used for flutter calculations in Figure 2.14. 52
3.1 Nondimensional parameters for straight, swept, tapered wing. 69
4.1 Manufacturability and performance for anisotropic and isotropic lattices. 92
4.2 Manufacturability for manufacturing and designing a bracket with different manufacturing orientations. 93
5.1 Comparison between frequencies obtained from the full-order lattice model and the low-order beam model. 104
5.2 Comparison between frequencies obtained from the full-order lattice model and the low-order beam model for a case where the center of gravity and elastic axis do not coincide with the mid-chord. 105
B.1 Rates of convergence of $L_2$ solution error for different discretization orders in 2D. 150
B.2 Rates of convergence of $L_2$ solution error for different discretization orders in 3D. 151
Some fear flutter because they do not understand it.  
And some fear it because they do.  
— Theodore von Kármán

With the impact of the airline industry on the environment increasingly under scrutiny and airlines operating on ever-tighter profit margins, the desire for more efficient transport aircraft is stronger than ever. This has led the aircraft industry to investigate ways of increasing the fuel efficiency of its aircraft designs. To meet NASA’s ambitious N+3 goals of reducing aircraft fuel burn by 60% by 2035, a move towards novel aircraft configurations is necessary.

Improvements in fuel efficiency since the start of the jet era have predominantly come from improvements in engine technology.1 Further advances in fuel efficiency can be achieved by engine improvements as well as improvements in the airframe design,2 specifically in aircraft structures, materials, aerodynamics, and especially the integration between those disciplines. For instance, the use of more advanced materials in the Boeing 787 has led to lighter—but more flexible—wings, which together with improved aerodynamics and more efficient engines has led to a large improvement in fuel efficiency.

For the aerodynamic design of next-generation aircraft concepts, there is a push for higher aspect ratio wings to increase the aerodynamic efficiency by reducing the induced drag³ examples being the D8.x aircraft4 or the Truss-Braced Wing concept.5 These high-aspect ratio wings are less stiff and therefore aeroelastic effects become an even more important driver in the wing design.6 Flutter constraints should therefore be included in the design of these aircraft as early as possible, preferably in the conceptual design stage. When we turn to novel aircraft configurations in particular, considering flutter in early design stages is crucial, as almost no historical data is available for such configurations. This will prevent costly design changes in later design phases, or worse during flight testing.

Problematically, these commercial jetliners operate in the transonic flow regime, for which accurate flutter prediction is difficult and which therefore typically requires high-fidelity flow solutions. Computing such flow solutions is too expensive for the conceptual design phase, where potentially thousands of designs need to be evaluated. Thus, there exists

1. Lee et al., Historical and Future Trends in Aircraft Performance, Cost, and Emissions. 2001
2. Bezos-O’Connor et al., Fuel Efficiencies Through Airframe Improvements. 2011
4. Drela, Development of the D8 Transport Configuration. 2011
a need for a flutter model that is cheap enough to allow for potentially thousands of flutter evaluations of wing designs in the conceptual design phase, but which is also accurate enough to capture subtle design trade-offs for commercial transport aircraft operating in the transonic flow regime. Moreover, even when a flutter problem has been identified, it is not always clear how to alter the design to mitigate flutter instabilities.

The focus of this thesis, therefore, is to (1) develop an accurate low-order transonic flutter prediction method to aid in the conceptual design phase for next-generation transport aircraft, and (2) to offer a possible flutter mitigation strategy through novel manufacturing techniques, specifically focusing on a design methodology for additive manufacturing. We therefore develop an accurate, fast, low-order model for small-amplitude transonic unsteady airloads, intended for flutter prediction in early-stage design and optimization of transport aircraft. Furthermore, we develop a design methodology for additively manufactured lattice structures, which can be aeroelastically tailored to mitigate flutter.

In the following, we provide a review of the problems associated with transonic flutter and discuss the state-of-the art methods for transonic flutter prediction (Section 1.1), review methods for design for additive manufacturing (Section 1.2), and discuss aeroelastic tailoring strategies (Section 1.3). The goals for this thesis as well as its structure are outlined in Section 1.4.

1.1 TRANSONIC FLUTTER

Aeroelasticity is the study of aerodynamic, elastic, and inertial forces on a body in a fluid flow. Flutter is an aeroelastic phenomenon where these forces start exciting each other, leading to an instability in the structure. In an aircraft wing, this results in large cyclic bending and twisting motions of the wing, likely leading to structural failure of the wing. The onset of flutter, therefore, has to be avoided at all times and rigorous flight testing procedures are in place to ensure that flutter does not occur within an aircraft's operational envelope. Ideally, engineers design around aeroelastic issues as early as possible in the design process by analyzing the aeroelastic behavior of aircraft wings well before such flight tests.

Incorporating flutter prediction into early-stage design, however, is a considerable challenge for the transonic flow regime relevant to most civil transport aircraft designs, since existing accurate models for transonic flutter prediction typically require extensive Computational Fluid Dynamics (CFD) analyses. Such simulations are too expensive to conduct in early design stages where potentially thousands of wing designs might be considered. A wealth of literature is available on static aeroelastic optimization of aircraft wings for both conventional tube-wing configu-
rations,\textsuperscript{7–9} and next-generation configurations.\textsuperscript{10} Dynamic aeroelasticity, however, is often not included in these studies. Flutter constraints can be included in state-of-the-art multidisciplinary design optimization settings, but even if the static aeroelastic methods are high fidelity, the methods for unsteady (compressible) flow usually either linearize the unsteady response or use a Prandtl-Glauert correction,\textsuperscript{11,12} which is limited to subsonic flow. Moreover, all aforementioned methods use high-fidelity aerodynamic and structural analysis methods, and are therefore too expensive to use in conceptual aircraft design. Conceptual aircraft design tools, such as SUAVE,\textsuperscript{13} typically do not include flutter constraints. Mallik et al.\textsuperscript{14} included transonic flutter constraints in a Multi-Disciplinary Design Optimization (MDO) setting for investigating Truss-Braced Wing designs, but the flutter model consisted of Theodorsen theory with compressibility corrections, limiting its accuracy for transonic flows.

While flutter for low subsonic flows can, in general, be accurately predicted with linearized small-disturbance theories, such methods fail for transonic flow where the traditional small-disturbance formulation is inherently nonlinear.\textsuperscript{15} For example, linear theory predicts that thinner wings would be less susceptible to flutter, but wind tunnel tests have shown the opposite to be true.\textsuperscript{15} Another interesting phenomenon that occurs in transonic flows is the transonic dip in the flutter boundary.\textsuperscript{16–19} In subsonic flow, information travels in all directions, but in supersonic flow, information can only travel in the direction of the flow. Thus, for an airfoil in transonic flow, pressure perturbations from e.g. the trailing edge take a much longer time to reach the leading edge of the airfoil compared to subsonic flow. This is illustrated in Figure 1.1, where in transonic flow, the information travel from B to A has to take a much longer route than the information travel from A to B. This results in phase lag in the aerodynamic response, which in turn—together with other processes at work—leads to a very different flutter response: a dip in the transonic flutter boundary.

![Figure 1.1: Information travel for airfoil in transonic flow. The path from B to A is much longer than from A to B.](image-url)
Linear theories cannot predict the location or shape of the transonic dip, and tend to produce optimistic estimates for the flutter boundary,\(^2\) a typical transonic flutter boundary is shown in Figure 1.2 together with a flutter boundary estimate from linear theory. Linear theory misses the transonic dip, computing an overly optimistic estimate for transonic flow conditions and an overly conservative estimate closer to Mach 1.

The seminal work by Theodorsen\(^2\) is widely used for incompressible flutter predictions for two-dimensional airfoils. Several researchers extended Theodorsen’s theory to subsonic flows,\(^22–24\) with none of these methods extending to the transonic regime. Moreover, Theodorsen theory is only strictly valid for simple harmonic motion, which applies only at the flutter boundary. For flutter prediction of aircraft wings, Barmby et al.\(^26\) extended Theodorsen’s theory to swept wings in incompressible flow. Yates extended this theory to subsonic, supersonic and hypersonic flow—the method loses accuracy for transonic flow—with a modified strip theory and validated the theoretical results with experimental results.\(^26\)

Possio\(^2\) applied the acceleration potential to a two-dimensional subsonic compressible nonstationary problem and arrived at an integral equation (Possio’s equation) for subsonic unsteady compressible flow. Closed-form solutions to an approximation of Possio’s equation were only found in 1999.\(^28,29\) However, the accuracy of Possio’s method deteriorates for transonic flows and for high-frequency oscillations.\(^30\) Lomax et al.\(^31\) used piston theory to analytically find the lift response to unsteady perturbations in subsonic flow, but only for the non-circular part of the lift. Stahara and Spreiter,\(^32\) Isogai,\(^33\) and Dowell\(^34\) developed models for linearized transonic flow, applicable to thin airfoils in shock-free flows. A linearization of the Transonic Small-Disturbance (TSD) equation for unsteady transonic flow was first proposed by Nixon,\(^35\) which involves changing the airfoil geometry for transonic flows such that the shock location does not change during any perturbations.

These analytical/theoretical methods all suffer from a loss of accuracy in transonic flows with shocks on the airfoil surface, which are typical on jet transports. Other approaches have attempted to address

---

**Figure 1.2:** Transonic flutter boundary with typical transonic dip, which linear theory cannot predict.

---


25. Barmby et al., *Study of Effects of Sweep on the Flutter of Cantilever Wings*. 1951


29. Lin et al., *Aerodynamic Lift and Moment Calculations Using a Closed-Form Solution of the Possio Equation*. 2000


32. Stahara et al., *Development of a Non-linear Unsteady Transonic Flow Theory*. 1973

this limitation by deriving models using a combination of theory and experimental data, as in the work of Leishman and Nguyen who formulated a state-space representation of unsteady airfoil behavior with some parameters estimated from experimental data, or by deriving reduced-order models from high-fidelity CFD models. A reduced-order modeling approach starts with the full state (flow field) information collected from a CFD solver and identifies a low-dimensional subspace in which the most important system dynamics evolve. Methods such as the harmonic balance, proper orthogonal decomposition, and Volterra series have all been used to derive aeroelastic reduced-order models. These methods have been applied in practice successfully, amongst others, for unsteady transonic flow around full wings, a full F-16 aircraft configuration, and many more.

In contrast to the reduced-order modeling approach, which starts with a high-order model and reduces the state by pure numerics, our work constructs a physics-based low-order model a priori, in terms of the flow-field’s volume-source and vorticity moments. Our state variables are the unsteady perturbations of these moments from some known steady transonic solution, and their time evolution is governed by postulated equations whose coefficients are treated as parameters. These parameters are calibrated using high-fidelity CFD results. Our underlying assumption is that the baseline steady flow is statically nonlinear, but sufficiently small disturbances around the baseline solution can be assumed to produce a linear response in the overall volume-source and vorticity moments. The resulting model produces unsteady perturbation lift and moment outputs, which can be combined with a conventional typical-section structural elastic model to obtain an overall aeroelastic state-space model suitable for flutter prediction.

We develop the present aerodynamic model as an extension of the one used in aswing, which employs Theodorsen theory for calibration. In contrast, the present model is calibrated using data from high-fidelity Euler CFD simulations of the transonic unsteady flow around airfoils. This model is then extended to aircraft wings using strip theory and 3D corrections.

1.2 DESIGN FOR ADDITIVE MANUFACTURING

Additive manufacturing builds up products in a layer-wise fashion, eliminating most of the geometric limitations present in subtractive manufacturing methods. However, designing for additive manufacturing is still not a straightforward problem. This is especially true for the design of lightweight structures, such as lattice structures. Topology optimization methods for such structures are either too expensive or start with an arbitrary lattice topology which limits the performance of the resulting
part. This thesis describes a methodology to design the topology of the lattice such that it is aligned with the stress direction and such that its density is based on the stress magnitude throughout the part, all without the need for expensive topology optimization steps.

Additive manufacturing is attractive to high-performance industries as the cost of a product does not scale exponentially with geometric complexity.\textsuperscript{49} It is therefore used across industries, for example in the aerospace, automotive, and biomedical industries.\textsuperscript{50–52} The interest in additive manufacturing has sky–rocketed over the past decade due to the lower cost resulting from the expiration of key patents and therefore is no longer confined to just rapid prototyping applications, but is currently used in final products\textsuperscript{53} and for tooling.\textsuperscript{54} In final products, additive manufacturing can be used to increase system performance,\textsuperscript{55} for example by better optimized designs or by improving heat dissipation through free–form cooling channels.\textsuperscript{56} For the aerospace industry in particular, additive manufacturing offers a way to drastically reduce the buy–to–fly ratio of its parts and reduce its environmental impact.\textsuperscript{57}

Additive manufacturing allows for more design freedom than subtractive manufacturing techniques, but this also results in design methods for subtractive manufacturing being suboptimal for additively manufactured parts. Designers therefore turn to inspiration from nature (biomimicry),\textsuperscript{58} or resort to purely numerical design optimization approaches using topology optimization methods.\textsuperscript{59}

An example of a bio-inspired structure that is of interest to structural engineers is the lattice structure. Lattice structures occur in almost any living tissue in nature, displaying optimized properties such as stiffness–to–weight ratio, strength–to–weight ratio,\textsuperscript{60,61} thermal conductivity, acoustic absorption, and gas permeability.\textsuperscript{62} Furthermore, lattice structures are typically more resistant to buckling, and are attractive from a fail–safe design perspective as they typically allow for multiple load paths through a part. In the conventional structural engineering context, they are usually called truss structures, and are used extensively in civil engineering for e.g., buildings, bridges, or transmission towers,\textsuperscript{63–65} as well as in aerospace engineering—mostly in the early days of flight.\textsuperscript{66} However, those structures are still quite different from lattice structures in nature, for example the trabecular bone structure in animal bones.

In order to design structural parts for additive manufacturing, topology optimization methods are often used. Such topology optimization methods are based on structural finite element methods where either each part of the domain is assigned a density determined by an optimizer in so-called Solid Isotropic Material with Penalization (SIMP) methods,\textsuperscript{59,67} or where the optimizer can control the parameters of a level–set to yield a sharp boundary for the final part.\textsuperscript{58,69} However, such
methods require an immense number of design variables in order to resemble an organic structure. For example, recently Aage et al. performed a topology optimization study on the internal wing structure of a commercial jetliner which took 1–5 days on 8,000 cores to complete.70

Alternatively, one can also directly design lattice structures. In this approach, ground structure optimization methods71 are often used. These methods numerically approximate optimal Michel structures72-73 using a finite number of lattice struts, yielding a truss or lattice structure more resembling an animal bone structure. However, these methods are inherently limited by the original mesh on which the ground structure is generated. Graf et al.74 changed the mesh for the ground structure by tailoring unit cells to the stress state in different parts of the structure, but only used a limited number of unit cells.

Although additive manufacturing offers larger design freedom compared to conventional manufacturing techniques, some manufacturing constraints remain. To illustrate this, we provide a brief overview of manufacturing techniques (Figure 1.3). We consider three main classes of manufacturing techniques; for a more complete overview of additive manufacturing processes, see e.g., Ref. 75. First, in wire-feed processes (Figure 1.3b), the material gets deposited by the printer head which also melts the material. The most well-known additive manufacturing process—Fused Filament Fabrication (FFF)—is a wire-feed process. This manufacturing process is used widely for desktop printers, but also to print very large structures as it scales well with dimension, requiring only increasingly large movements from the print head. Direct Energy Deposition (DED) uses a similar process to FFF, where essentially a 3D part is welded together. It is attractive for producing large metal parts or repairing large existing parts, and has been used to manufacture entire bridges or complete rocket fuel tanks. Second, powder-based manufacturing processes use a localized heating source to melt powder which then solidifies into a part (Figure 1.3c). Selective Laser Melting (SLM) is widely used to produce metal parts, whereas Selective Laser Sintering (SLS) is used for plastic parts. Electron Beam Melting (EBM) is similar to SLM, but uses an electron beam to melt the powder, instead of a laser. Third, Stereolithography (SLA) works by focusing an Ultraviolet (UV) laser on a photopolymer, which solidifies when exposed to UV (Figure 1.3d). It is typically used to manufactured small, very detailed parts, such as (molds for) jewelry.

All these manufacturing methods build parts in a layer-wise fashion and new layers therefore need support from the layer below them, limiting the overhang angles between subsequent layers. Even for powder-based techniques, such overhang limits usually remain to prevent warping. Selective Laser Sintering, however, does not have such overhang constraints as it is powder-based and the temperature gradients are

---

70. Aage et al., Giga-Voxel Computational Morphogenesis for Structural Design. 2017
71. Zegard et al., GRAND: Ground Structure Based Topology Optimization for Arbitrary 2D Domains using MATLAB. 2014
72. Michell, The Limits of Economy of Material in Frame Structure. 1904
73. Dorn, Automatic Design of Optimal Structures. 1964
75. Bikas et al., Additive Manufacturing Methods and Modelling Approaches: A Critical Review. 2015
Critical overhang angles are typically \(45° \pm 10°\), but this number is strongly dependent on the manufacturing process and the material. As illustrated in Figure 1.3, the geometric resolution is limited—and dependent on manufacturing process and material—which limits the minimum wall thickness.

![Part of Blended Wing Body that is to be manufactured](image_a)

![Wire-feed additive manufacturing processes, such as Fused Filament Fabrication (FFF)](image_b)

![Powder-based additive manufacturing processes, such as Selective Laser Sintering (SLS), and Selective Laser Melting (SLM)](image_c)

![Stereolithography (SLA)](image_d)

Figure 1.3: Overview of additive manufacturing techniques considered in this thesis.
the manufacturing constraints. Including manufacturing constraints in the optimization is important, because dedicated design for additive manufacturing—including its manufacturing constraints—is of major importance for the introduction of additive manufacturing as an economically viable production technique.

In this thesis, we devise a methodology for designing lattice structures to be manufactured using additive manufacturing, focusing on optimal and manufacturable structures. The methodology is developed to be cheap to allow for inclusion in early design stages, which is enabled by the use of low-order methods and adaptive meshing techniques.

1.3 AEROELASTIC TAILORING

Novel manufacturing techniques open up additional design space for aerospace vehicles. High aspect ratio wings in novel aircraft concepts, for example, offer the benefits of higher aerodynamic efficiency, but present the challenge of being more susceptible to aeroelastic problems such as flutter. Novel manufacturing techniques present an opportunity to address this challenge via improved material properties (e.g., increased stiffness) and also by enabling unconventional internal wing layouts. This thesis examines the possibility of designing the internal structure of aircraft wings as a lattice structure, while mitigating flutter. The fabrication of these lattice structures is enabled by advances in additive manufacturing technology.

To date, the aeroelastic optimization of aircraft wings has mostly focused on conventional internal wing structures—i.e., an orthogonal array of ribs and spars. The structural efficiency of the wing can be further improved by using novel manufacturing techniques, which allow for moving away from the conventional orthogonal rib-spar lay-

---

82. Qian, Undercut and Overhang Angle Control in Topology Optimization: A Density Gradient Based Integral Approach. 2017
83. Shidid et al., Just-In-Time Design and Additive Manufacture of Patient-Specific Medical Implants. 2016
84. Vaneker, The Role of Design for Additive Manufacturing in the Successful Economical Introduction of AM. 2017

---

Figure 1.4: Manufacturing constraints need to be included in the optimization to ensure the final manufactured part has optimal performance.

---

86. Blair et al., Joined-Wing Aeroelastic Design with Geometric Nonlinearity. 2005
Several ways to parametrize the internal structure of the wing have been demonstrated. For example, Balabanov and Haftka,87 and Locatelli et al.88 retained the rib-spar layout but allowed for curvilinear components, which is therefore an inherently two-dimensional parametrization. Alternatively, three-dimensional parametrizations are used where the optimizer is allowed to place material anywhere in the wing. These problems are typically solved using simp methods or level-set methods, as demonstrated by, amongst others, Maute and Reich,89 Dunning et al.,90 and Kambampati et al.91 for aircraft wings and wing boxes. A combination of a conventional wing structure layout with topologically optimized parts is investigated by Stanford and Dunning92 where an orthogonal rib-spar layout is used but where topology optimization is applied to the ribs and spars. Recently, Townsend et al.93 optimized plate-like wing structures using level-set topology optimization to minimize the weight of the wing and to mitigate flutter, and demonstrated large weight savings by using such an optimized structure. As mentioned before, Aage et al.70 used topology optimization to design the internal structure of an aircraft wing using giga-voxel resolution, but that study did not consider aerostructural coupling or dynamic aeroelasticity.

Other studies have shown the potential benefits of novel manufacturing techniques on wing material properties. For example, Haddapour et al.94 and Stodieck et al.95 demonstrated increased wing stiffness for tow-steering with composite materials.96,97 Stanford et al.98 investigated the differences between several novel tailoring schemes on wings subject to transonic flutter constraints and found that a spatially varying thickness distribution—through additive manufacturing—yielded more benefit than tow-steering alone.

Aeroelastic optimization has also been conducted for helicopter and propeller blades. Ganguli et al.99 used high-fidelity flow and structural simulations to minimize rotor vibrations, while constraining aeroelastic stability. Glaz et al.100 developed a design methodology to minimize vibrations in helicopter forward flight using surrogate models for the structural and aerodynamic loads. Pagano et al.101 coupled aeroelastic simulations to aeroacoustic evaluations for the design of propeller blades that minimize noise. In this work, however, we focus on aircraft wings and simply note that the design methodology could also be applied to helicopter and propeller blades in the future, as they could also take advantage of lightweight aeroelastically tailored lattice structures.

Lattice structures are of particular interest in aircraft design because of their high stiffness-to-weight ratio.62 Walker et al.102 optimized the internal structure of a wing using lattice structures utilizing commercial software. Lattice structures have also been used in morphing wing applications.103,104 In this thesis we focus on additively manufactured lattice structures as a possible solution to mitigating flutter of high aspect
ratio wings, using the aforementioned design methodology for lattice structures.

1.4 THESIS GOALS & OUTLINE

This thesis develops low-order methods for transonic flutter prediction to be used in the conceptual design stage, and develops a design methodology for lattice structures to mitigate aeroelastic instabilities with minimal mass increase through additive manufacturing.

The first goal therefore is to improve the accuracy and lower the cost of flutter analysis in the conceptual design stage to prevent costly design changes in later design phases or testing. This can be achieved by developing a physics-based low-order model for transonic flutter of airfoils (Chapter 2), and then applying this model to swept high-aspect-ratio wings through 3D corrections (Chapter 3). As an application, we investigate the influence of transonic flutter on the conceptual design of next-generation transport aircraft by implementing this low-order flutter model in a physics-based conceptual design tool (also in Chapter 3).

Before any lattice structure can be aeroelastically tailored, the lattice itself has to be designed. The second goal is therefore to devise and implement a design methodology which includes manufacturing constraints for additive manufacturing techniques to increase its applicability (Chapter 4). The third goal is to use such lattice structures for the internal structure of an aircraft wing, while tailoring their aeroelastic properties to mitigate any aeroelastic instabilities (Chapter 5).

Finally, the approaches to achieving these goals are summarized in Chapter 6, where we also discuss avenues for future contributions as well as further applications of this research.
This chapter develops a new physics-based airfoil flutter assessment capability that is suitable for incorporation in early-stage conceptual design studies, using the approach in Figure 2.1. The model is based on small unsteady disturbances from a steady solution, with the flow-field's lowest moments of vorticity and volume-source density perturbations as its states (Section 2.1). Several unknown coefficients in this model are calibrated using high-fidelity CFD simulations (Section 2.2). This low-order model maintains the same level of accuracy as substantially more expensive Euler or Navier-Stokes solutions (Section 2.3), and is used to describe the aeroelastic behavior of several airfoils (Section 2.4).

2.1 PHYSICS-BASED STATE-SPACE MODEL FORMULATION

The flutter model consists of an aerodynamic model (Section 2.1.1) and a structural model (Section 2.1.2). The aeroelastic state-space system can be formed by combining these two models. Analysis of the resulting coupled state-space system indicates whether flutter occurs.
2.1.1 AERODYNAMIC LOW-ORDER MODEL

The physics-based low-order aerodynamic model begins with the Helmholtz decomposition, which expresses any instantaneous velocity field \( \mathbf{V} \) as integrals over the velocity’s divergence and curl fields, commonly known as the volume-source density \( \sigma \) and the vorticity \( \omega \):

\[
\begin{align*}
\sigma & \equiv \nabla \cdot \mathbf{V} \\
\omega & \equiv \nabla \times \mathbf{V}
\end{align*}
\]

\( \sigma \) and \( \omega \) are defined and computed appropriately. One example is given by Oskam,\(^{106}\) who obtained 2D transonic airfoil solutions using this approach. Specifically, the volume-source field was represented by panels covering space around the airfoil, whose \( \sigma \) strengths were computed via the full-potential equation, together with the usual normal-doublet panels on the surface computed via flow tangency.

We define the unsteady volume-source and vorticity fields to have the form

\[
\begin{align*}
\sigma(r, t) &= \sigma_0(r) + \Delta \sigma(r, t) \\
\omega(r, t) &= \omega_0(r) + \Delta \omega(r, t),
\end{align*}
\]

where \( \sigma_0 \) and \( \omega_0 \) correspond to some known steady transonic flow, and \( \Delta \sigma \) and \( \Delta \omega \) are small perturbations defining the unsteady flow we are seeking. Since the Helmholtz decomposition (2.3) is linear in \( \sigma \) and \( \omega \), the corresponding velocity field will have the same baseline plus perturbation form:

\[
\mathbf{V}(r, t) = \mathbf{V}_0(r) + \Delta \mathbf{V}(r, t).
\]

We stress that “small disturbance” here refers to small disturbances from the steady solution, and not small perturbations from the freestream (which is the more common usage).


\(^{106}\) Oskam, Transonic Panel Method of the Full Potential Equation Applied to Multicomponent Airfoils, 1985
To construct the low-order model for the overall airfoil lift and pitching moment, we first assume that the effects of the $\sigma$ and $\omega$-fields (Figure 2.2) can be captured by only their leading spatial moments. The zeroth moment of the vorticity, or equivalently its overall lumped strength, is the airfoil circulation

$$\Gamma(t) \equiv \iint \omega(t) \cdot \hat{y} \, d\mathcal{A},$$

where $\hat{y}$ is the unit vector into the plane. For steady flow, the circulation captures the lift exactly via the Kutta-Joukowsky theorem. As sketched in Figure 2.2, the volume-source field $\sigma$ is mostly positive over the upper front of the airfoil where the flow accelerates, and mostly negative over the shock and the upper rear where the flow decelerates. Its zeroth moment is comparable to the wave drag, which typically is very small and negligible compared to the lift force:

$$\frac{D_{\text{wave}}}{\rho_\infty V_\infty} \approx \iint \sigma \, d\mathcal{A} \approx 0.$$

Consequently, we must consider the first $x$-moment of the volume-source field, which is its overall lumped $x$-doublet strength,

$$\kappa_x(t) \equiv \iint -x \sigma(t) \, d\mathcal{A}.$$
which is here postulated to capture the compressibility effects on the circulation and the resulting lift and moment.

Corresponding to decompositions (2.4) and (2.5), we will here actually seek the small perturbations

\[ \Delta \Gamma(t) \equiv \Gamma(t) - \Gamma_0 \]
\[ \Delta \kappa_x(t) \equiv \kappa_x(t) - \kappa_{x,0} \]

from the steady-solution values \( \Gamma_0, \kappa_{x,0} \). The steady circulation \( \Gamma_0 \) is directly proportional to the steady lift coefficient \( c_{\ell_0} \) or the corresponding angle of attack \( \alpha_0 \), while \( \kappa_{x,0} \) is also a strong function of the freestream Mach number \( M_\infty \) and to some extent of the airfoil shape. Typical \( \kappa_{x,0} \) variation for a transonic airfoil is shown in Figure 2.3. Note that the dependence on \( c_{\ell_0} \) is almost linear.

For the imposition of flow tangency at the control point, we define the instantaneous airfoil-surface normal velocity at the control point as

\[ U_n(t) \equiv U_{c.p.} \cdot \hat{n}_{c.p.} \]

while the instantaneous fluid normal velocity at the control point is defined as

\[ V_n(t) \equiv V_{c.p.} \cdot \hat{n}_{c.p.} \]

Next, we approximate the normal fluid velocity perturbation at the three-quarter control point as

\[ \Delta V_n(t) = A_\Gamma \Delta \Gamma + A_\Gamma \Delta \dot{\Gamma} + A_\kappa \Delta \kappa_x, \]

which can be considered to be a lumped form of the velocity decomposition (2.3), with the \( \Delta \dot{\Gamma} \) term capturing the contribution of the shed vorticity. Following the single-element Weissinger vortex-lattice method\textsuperscript{107,108} we set

\[ A_\Gamma = -\frac{1}{\pi} \frac{1}{\ell}, \]

![Figure 2.3: Variation of doublet strength \( \kappa_{x,0} \) with \( c_{\ell_0} \) and \( M_\infty \) for the RAE2822 transonic airfoil, as computed using MSES.\textsuperscript{109–111}](image)

\textsuperscript{107} Weissinger, Über die Auftiefrsverteilung von Pfeilflügeln. 1942
\textsuperscript{108} Weissinger, The Lift Distribution of Swept-Back Wings. 1947
and following the ASWING shed-vorticity model\textsuperscript{47,48} we set

\[ \Gamma_0 = -\frac{b_1}{V_\infty}. \]

The calibrated lag constant value \( b_1 = 2/\pi \) gives results that closely approximate those of Theodorsen.\textsuperscript{21} However, the values of \( \Gamma_0 \) and \( \Gamma_1 \) can change as a function of, amongst others, the airfoil geometry and Mach number. If we place the \( x \)-doublet at \((x_\kappa, z_\kappa) = (c/2, c/4)\), then the doublet’s influence on the three quarter-chord control point gives

\[ A_\kappa = \frac{1}{4\pi c^2}. \]

In general, however, the effective doublet position is not known, and is expected to depend on the baseline steady solution. Hence, the \( A_\kappa \) coefficient will be a calibrated function of the baseline steady flow solution. Regardless, we obtain the evolution equation for the circulation by requiring flow tangency at the control point,

\[ \Delta V_n = U_n \]

\[ \Rightarrow A_1 \Delta \Gamma = U_n - A_1 \Delta \Gamma - A_\kappa \Delta \kappa_x. \]

The evolution equation for the \( x \)-doublet is postulated to be second-order. This follows from the full potential equation,\textsuperscript{112}

\[ \nabla^2 \phi - \frac{1}{a^2} \left[ \frac{\partial^2 \phi}{\partial t^2} + \frac{\partial}{\partial t} \left( \frac{\|V\|^2}{2} \right) + V \cdot \nabla \left( \frac{\|V\|^2}{2} \right) \right] = 0. \] (2.6)

Note that \( \nabla^2 \phi = \sigma \), so that taking the Laplacian of Eq. (2.6) would give a wave equation for \( \sigma \), involving second time derivatives. Hence, \( \kappa_x \)—the first moment of \( \sigma \)—must at a minimum have a second time derivative in its evolution equation. Therefore, we postulate the evolution equation for \( \kappa_x \) to be

\[ \Delta \kappa_x = B_1 \Delta \Gamma + B_\kappa \Delta \kappa_x + B_\dot{\kappa} \Delta \dot{\kappa}_x, \] (2.7a)

where

\[ B_1 = B_1 \left( M_\infty, c_\ell_0 \right), \quad B_\kappa = B_\kappa \left( M_\infty, c_\ell_0 \right), \quad B_\dot{\kappa} = B_\dot{\kappa} \left( M_\infty, c_\ell_0 \right) \] (2.7b)

are calibrated functions for the airfoil (or airfoil family) being analyzed, and where

\[ c_\ell_0 = \frac{2\Gamma_0}{V_\infty c} \]

is the lift coefficient of the baseline solution. The lift and the moment around the elastic axis, per unit span, are

\[ \Delta L_c(t) = \rho_\infty V_\infty \Delta \Gamma + \rho_\infty c \Delta \dot{\Gamma} \]
\[ \Delta L(t) = \Delta L_c(t) + \Delta L_{nc}(t) \]
\[ \Delta M(t) = \Delta M_{c/4}(t) + \Delta L_c(t) d + \Delta M_{nc}(t), \]
where $\Delta M_{c/d}(t) \equiv \frac{1}{2} \rho_{\infty} V_{\infty}^2 c^2 \Delta c_m$. Here, $\Delta c_m$ is a calibrated function of $\Delta \Gamma/c V_{\infty}$, $\Delta \Gamma/c V_{\infty}^2$, $\Delta \kappa/c V_{\infty}^2$, and $\Delta \kappa/c V_{\infty}$, using a least-squares fit to a number of unsteady Euler or RANS calculations. Following Theodorsen, the non-circulatory lift $\Delta L_{nc}(t)$ and moment around the elastic axis $\Delta M_{nc}(t)$ are added-mass terms defined as

$$\Delta L_{nc}(t) = \frac{1}{4} \rho_{\infty} \pi c^2 \left[ \Delta \ddot{h} - \left( x_{ea} - \frac{c}{2} \right) \Delta \dot{\theta} \right] + \frac{1}{4} \rho_{\infty} \pi c^2 V_{\infty} \Delta \dot{\theta}$$

$$\Delta M_{nc}(t) = \frac{1}{4} \rho_{\infty} \pi c^2 \left( x_{ea} - \frac{c}{2} \right) \left[ \Delta \dddot{h} - \left( x_{ea} - \frac{c}{2} \right) \Delta \ddot{\theta} \right] - \frac{\rho_{\infty} \pi c^4}{128} \Delta \dot{\theta}$$

### 2.1.2 Typical Section Structural Model

We focus here on the typical section (Figure 2.4), as used frequently in the aeroelasticity community. Using Hamilton’s equation and Lagrangian mechanics, the equations of motion of an airfoil in heave and pitch motion are found to be\textsuperscript{113,114}

$$m \ddot{h} + k_h \Delta h + S_\theta \Delta \dot{\theta} = -\Delta L \quad (2.8a)$$

$$S_\theta \dddot{h} + k_\theta \Delta \dot{\theta} + I_\theta \Delta \ddot{\theta} = \Delta M \quad (2.8b)$$

where here we define $\Delta h$ to be the heave perturbation, defined positive downward, and $\Delta \theta$ to be the pitch angle perturbation, defined positive clockwise. $m$ is the total mass/span of the section, $S_\theta \equiv (x_{cg} - x_{ea}) m$ is the mass unbalance/span, where $x_{cg}$ and $x_{ea}$ are the distances of the center of gravity and elastic center, respectively, from the leading edge. $I_\theta$ is the moment of inertia/span around the elastic axis. Finally, $k_h$ is the heave displacement spring constant/span, and $k_\theta$ is the torsional spring constant/span. The pitching moment equation (2.8b) is constructed about the elastic axis (or shear center) location, defined to be a distance $d$ from the quarter chord point.

---

\textsuperscript{21} Theodorsen, General Theory of Aero-

\textsuperscript{113} Bisplinghoff et al., Aeroelasticity. 1996

\textsuperscript{114} Dowell, A Modern Course in Aeroelastics-

\textsuperscript{1935} Instability and the Mechanism of Flutter. 1955

---

\textsuperscript{21} Theodorsen, General Theory of Aero-

\textsuperscript{113} Bisplinghoff et al., Aeroelasticity. 1996

\textsuperscript{114} Dowell, A Modern Course in Aeroelastics-

\textsuperscript{1935} Instability and the Mechanism of Flutter. 1955
2.1.3 OVERALL STATE-SPACE MODEL AND FLUTTER PREDICTION

The aeroelastic model couples the aerodynamic and structural models, using the following overall state vector:

\[
x(t) = [\Delta h(t), \Delta \theta(t), \Delta w(t), \Delta \omega(t), \Delta \Gamma(t), \Delta \kappa_x(t), \Delta \dot{\kappa}_x(t)]^T,
\]

where \( \Delta \omega(t) \equiv \Delta \dot{\theta}(t) \) is the pitch rate perturbation and \( \Delta w(t) \equiv \Delta \dot{h}(t) \) is the vertical (downward) velocity perturbation. The vertical velocity and pitch rate force the aerodynamic system through the normal velocity at the control point,

\[ U_n = \Delta w + V_\infty \Delta \theta + \frac{c}{4} \Delta \omega. \]

The aeroelastic system equations now become

\[ \Delta \dot{h} = \Delta w \]
\[ \Delta \dot{\theta} = \Delta \omega \]
\[ m \Delta \dot{w} + \frac{\rho_\infty c}{A_f} \left( \Delta w + \frac{c}{4} \Delta \omega - A_f \Delta \Gamma - A_k \Delta \kappa_x \right) = -k_h \Delta h - \Delta L \]
\[ = -k_h \Delta h - \rho_\infty V_\infty \Delta \Gamma - \rho_\infty c \Delta \dot{\Gamma} \]
\[ = -k_h \Delta h - \rho_\infty V_\infty \Delta \Gamma \]
\[ = -\frac{\rho_\infty c}{A_f} \left( \Delta w + \frac{c}{4} \Delta \omega - A_f \Delta \Gamma - A_k \Delta \kappa_x \right) \]
\[ - \Delta L_{nc}(t) \]
\[ S_{\theta} \Delta \dot{\theta} + I_{\theta} \Delta \dot{\omega} = -k_{\theta} \Delta \theta + \Delta L_d + \Delta M \]
\[ = -k_{\theta} \Delta \theta + \rho_\infty V_\infty \Delta \Gamma d \]
\[ + \frac{\rho_\infty c}{A_f} \left( \Delta w + \frac{c}{4} \Delta \omega - A_f \Delta \Gamma - A_k \Delta \kappa_x \right) d \]
\[ + \Delta M_{cf/4} + \Delta M_{nc}(t) \]
\[ \Delta \dot{\Gamma} = \frac{1}{A_f} \left[ \Delta w + \frac{c}{4} \Delta \omega - A_f \Delta \Gamma - A_k \Delta \kappa_x \right] \]
\[ \Delta \ddot{\kappa}_x = B_\Gamma \Delta \Gamma + B_k \Delta \kappa_x + B_\kappa \Delta \dot{\kappa}_x. \]

Eq. (2.10) has the form of a standard descriptor state-space model

\[ E \dot{x} = Ax. \]

Flutter is indicated if any eigenvalue of the matrix \((E^{-1}A)\) has a positive real part. The unknown coefficients of the \(E, A\) matrices are calibrated using CFD simulations of forced pitching and heaving airfoil motions, as described in the next section.

2.2 MODEL FITTING FROM CFD DATA

This section describes our approach to fit the unknown coefficients of the low-order flutter model using data generated with unsteady CFD
simulations. The structural dynamics of the typical section airfoil are the same for subsonic and transonic flows, and it is therefore only the aerodynamic model that needs calibration for transonic flows. We decompose the state-space system in Eq. (2.11), to extract the aerodynamic state-space system with state $x_a$, input $u_a$, and state-space matrices $A_a$ and $B_a$, defined as follows:

$$
\begin{bmatrix}
\Delta \Gamma \\
\Delta \kappa_x \\
\Delta \kappa_x
\end{bmatrix} =
\begin{bmatrix}
-\frac{A_f}{\tilde{A}} & -\frac{A_k}{\tilde{A}} & 0 \\
0 & 0 & 1 \\
B_f & B_k & B_k
\end{bmatrix}
\begin{bmatrix}
\Delta \Gamma \\
\Delta \kappa_x \\
\Delta \kappa_x
\end{bmatrix} +
\begin{bmatrix}
\frac{V_w}{\tilde{A}} & \frac{c/d}{\tilde{A}} & \frac{1}{\tilde{A}}
\end{bmatrix}
\begin{bmatrix}
\Delta \theta \\
\Delta \omega
\end{bmatrix}. \tag{2.12}
$$

2.2.1 STATE-SPACE SYSTEM IDENTIFICATION

The unknown coefficients in the low-order flutter model developed in Section 2.1 are $A_f$, $A_k$, $B_f$, $B_k$, $\tilde{A}$. To determine these coefficients from data, we employ the state-space system identification method of Dynamic Mode Decomposition (DMD). Specifically, we use dynamic mode decomposition with control (DMDc), which requires us to collect snapshots of the states and control inputs at each time step. Therefore, we first run unsteady compressible flow simulations for forced pitching/heaving oscillations to obtain $\Delta \Gamma(t_i)$, $\Delta \kappa_x(t_i)$, $\Delta \theta(t_i)$, $\Delta \omega(t_i)$, and $\Delta \omega(t_i)$ with $t_i = (i - 1)\Delta t$, $i = 1, \ldots, (N + 2)$.

These snapshots are collected in the matrices $X \in \mathbb{R}^{1 \times N}$, $X' \in \mathbb{R}^{2 \times N}$, $X'' \in \mathbb{R}^{3 \times N}$, and $Y' \in \mathbb{R}^{3 \times N}$ as follows:

$$
X = \begin{bmatrix} 
\Delta \kappa_x(t_1) & \Delta \kappa_x(t_2) & \ldots & \Delta \kappa_x(t_N) 
\end{bmatrix},
$$

$$
X' = \begin{bmatrix} 
\Delta \Gamma(t_2) & \Delta \Gamma(t_3) & \ldots & \Delta \Gamma(t_{N+1}) 
\end{bmatrix},
$$

$$
X'' = \begin{bmatrix} 
\Delta \kappa_x(t_3) & \Delta \kappa_x(t_4) & \ldots & \Delta \kappa_x(t_{N+2}) 
\end{bmatrix},
$$

$$
Y' = \begin{bmatrix} 
\Delta \theta(t_2) & \Delta \theta(t_3) & \ldots & \Delta \theta(t_{N+1}) \\
\Delta \omega(t_2) & \Delta \omega(t_3) & \ldots & \Delta \omega(t_{N+1}) \\
\Delta \omega(t_2) & \Delta \omega(t_3) & \ldots & \Delta \omega(t_{N+1}) 
\end{bmatrix}.
$$

Note that we need the matrix $X''$ because we postulated the evolution equation for $\kappa_x$ to be second-order in Eq. (2.7a).

DMD fits a discrete state-space system, relating $X''$ to $X'$, $X$ and $Y'$ as

$$
X'' \approx a X' + a_l X + b Y', \tag{2.13}
$$

where $a \in \mathbb{R}^{2 \times 2}$, $a_l \in \mathbb{R}^{2 \times 1}$, $b \in \mathbb{R}^{2 \times 3}$ are matrices containing the unknown coefficients to be fit. Eq. (2.13) can be rewritten in the form

$$
X'' \approx G \Omega, \tag{2.14}
$$

115. Schmid, Dynamic Mode Decomposition of Numerical and Experimental Data, 2010
116. Proctor et al., Dynamic Mode Decomposition with Control, 2016

44 2.2 MODEL FITTING FROM CFD DATA
with \( G = [a \ a_l \ b] \in \mathbb{R}^{2 \times 6} \) and \( \Omega = [X' \ X \ Y'] \in \mathbb{R}^{6 \times N} \).

In Eq. (2.14), \( X'' \) and \( \Omega \) are obtained from the CFD simulations, and \( G \) is then found by solving the least-squares optimization problem

\[
G = \arg \min_G \|X'' - G\Omega\|_F.
\]

To solve this minimization problem, one computes a singular value decomposition (SVD) of \( \Omega = U\Sigma V^T \), where \( U \in \mathbb{R}^{6 \times 6}, V \in \mathbb{R}^{N \times 6} \), and the diagonal matrix \( \Sigma \in \mathbb{R}^{6 \times 6} \). The least-squares fit \( G \) is then found as

\[
G = X''V\Sigma^{-1}U^T.
\]

In contrast to the reduced-order modeling approach where DMD is typically used, we do not truncate the singular values in \( \Sigma \) when computing \( G \). In reduced-order modeling, the row dimension of the snapshot matrices is the total number of states in the CFD model (typically \( 10^4 \) or higher), and DMD is used to identify a model of much lower order. In contrast, in this system identification setting, we construct the data matrices \( X, X', X'', \) and \( Y' \) with our six physical low-order quantities, such that the state vector of the DMD representation already has the desired dimension.

The matrices \( a, a_l, \) and \( b \) are found by decomposing the linear operator \( U \) as

\[
[a, a_l, b] \approx [X''V\Sigma^{-1}U_1, X''V\Sigma^{-1}U_2, X''V\Sigma^{-1}U_3],
\]

where \( U_1 \in \mathbb{R}^{6 \times 2}, U_2 \in \mathbb{R}^{6 \times 1}, \) and \( U_3 \in \mathbb{R}^{6 \times 3} \) are the first two columns of \( U \), the third column of \( U \), and the last three columns of \( U \), respectively.

Finally, we relate the discrete state-space matrices \( a, a_l, b \) to the continuous state-space matrices \( A_a \) and \( B_a \) in Eq. (2.12) using Tustin’s method.\(^{117}\)

The process described in this section allows for fitting one state-space model for one particular freestream Mach number, airfoil and baseline lift coefficient. We would like to characterize the flutter boundary for a range of Mach numbers, airfoil thicknesses, and baseline lift coefficients. Therefore, the method described in this section is used to find the aerodynamic state-space matrices for a set of freestream Mach numbers, baseline lift coefficients, and different thicknesses of the same airfoil family. Splines are then fit for the entries in the aerodynamic state-space matrices to find a low-order model at an interpolatory Mach number, baseline lift coefficient, and/or airfoil thickness.

### 2.2.2 CFD DATA GENERATION

We fit the aerodynamic part of the state-space system using data from CFD solutions. Throughout this work, we use the SU2 tool suite.\(^{118}\) SU2
is a multi-purpose PDE solver, and specifically developed with aerospace applications in mind, such as unsteady compressible flow over an airfoil. For CFD applications, SU2 employs a finite volume method with standard edge-based structure on a dual grid. For unsteady simulations it uses a dual time-stepping strategy for high-order accuracy in time.\(^\text{119}\)

Two changes to SU2 were necessary for this work. First, the parameters of our problem \((\kappa_x, \Gamma)\) need to be computed by post-processing SU2 solutions. The circulation \(\Gamma(t)\) is defined as

\[
\Gamma(t) = \iint \omega(t) \cdot \hat{y} \, dA = \oint V(t) \cdot dl.
\]

If this line integral is computed sufficiently close to the airfoil, it is more accurate than integrating the vorticity, which is susceptible to round-off errors and noise in the gradient computation. Note that we need to choose the contour for the line integral to exclude the wake by keeping the contour in front of the trailing edge of the airfoil (Figure 2.5b), such that we compute the circulation \textit{without} including shed vorticity.\(^*\)

\(^*\)Shed vorticity should not be included, as we are interested in the instantaneous circulation change of the airfoil. The change in circulation of the airfoil sheds into the wake—the circulation change computed from a contour including both airfoil and wake would therefore be zero.

### Figure 2.5: Details of meshes to illustrate how the circulation and \(x\)-doublet are computed through line integrals.

(a) Detail of triangular mesh with cut. Black lines indicate primal grid with nodes (circles), thick black lines indicate the line along which the line integrals are computed.

(b) Mesh of NACA64A010 airfoil with line around airfoil that is used for the line integral in the computation of the circulation \(\Gamma\).

(c) Mesh of NACA64A010 airfoil with line around airfoil that is used for the line integral in the computation of the \(x\)-doublet \(\kappa_x\).

Computing the \(x\)-doublet through integration of the divergence of the velocity field is even more problematic, because the extrema in the volume-source field are highly localized. As an example, the volume-source field around the RAE2822 airfoil in subsonic and transonic flow

\(^\text{119}\) Jameson, Time Dependent Calculations using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings. 1991

\(^\text{109}\) Drela et al., Viscous-Inviscid Analysis of Transonic and Low Reynolds Number Airfoils. 1987

\(^\text{110}\) Drela, Design and Optimization Method for Multi-Element Airfoils. 1993
is shown in Figure 2.6 for different lift coefficients, as computed with MSEs.109–111

Using the contour shown in Figure 2.5c, the $x$-doublet is computed by evaluating

$$\kappa_x(t) = - \int x \nabla \cdot \mathbf{V}(t) \, d\mathcal{A} = - \oint x \mathbf{V}(t) \cdot \mathbf{n} \, ds + \int \mathbf{U}(t) \, d\mathcal{A}$$

$$= - \oint x \left[ \mathbf{V}(t) - \mathbf{V}_\infty \right] \cdot \mathbf{n} \, ds + \int \left[ \mathbf{U}(t) - \mathbf{U}_\infty \right] \, d\mathcal{A},$$

where $\mathbf{U}$ is the velocity in the freestream $x$-direction. The final expression for $\kappa_x(t)$ subtracts off $\mathbf{V}_\infty$ to minimize cancellation errors. When using

Figure 2.6: Volume-source fields around RAE2822 airfoil in subsonic and transonic flow as computed by MSEs.109–111
a structured grid (e.g., an O-grid or a C-grid for airfoils), computing these quantities using line integrals is straightforward. One can just track the grid lines at a certain distance around the airfoil. However, throughout this work we mostly use unstructured grids, complicating the computation of the line integrals. In this case, we find a set of edges in the mesh at a certain distance from the airfoil that form a closed loop around that airfoil. First, the set of edges that intersect a circle centered around the airfoil is identified, by finding the edges for which one edge point is inside the circle and one is outside on the circle (see Figure 2.5a). An arbitrary precision method by Shewchuk\(^{120}\) is employed for finding this set of edges. Subsequently, any hanging nodes are removed from this set. The result of such an approach is shown in Figure 2.5c. For the doublet computation, we must also integrate the \(U\) velocity within this circle, which is done by summing over all cells of the primal grid (instead of the dual grid, which SU2 uses).

The second change to SU2 was to alter the dynamic mesh rotations in order to allow for different prescribed motions—which include multiple frequencies with a ramp-up to maximum amplitude—and to allow for restarting the solution from a steady result without any discontinuities in the grid velocities. An example of such a prescribed motion is shown in Figure 2.7.

![Figure 2.7: Example of ramped forced pitching oscillation.](image)

2.3 VALIDATION OF AIRFOIL FLUTTER MODEL

Using the methodology outlined in Sections 2.1 and 2.2, low-order models are constructed for several different airfoils. First, we demonstrate the accuracy of the calibration for the aerodynamic model. Second, we validate the accuracy of the low-order model for the standard Isogai Case A.

2.3.1 EXAMPLE CALIBRATION RESULTS FOR RAE2822 AIRFOIL

The low-order model is calibrated using Euler simulations of pitching and heaving airfoils at different Mach numbers and reduced frequencies. The results in this section use simulations of an RAE2822 transonic airfoil in forced pitching motion along the path shown in Figure 2.7 with an

\(^{1}\)Choosing the radius of this circle is a trade-off between not including all generated source strength—if the radius is too small—or adding numerical noise—if the radius is too large; the appropriate radius is determined from a convergence study.

---

\(^{120}\) Shewchuk, Adaptive Precision Floating-Point Arithmetic and Fast Robust Geometric Predicates. 1997
amplitude of $\theta = 0.3^\circ$, and also in a forced heaving motion using a similar trajectory for $h$ with an amplitude of $h/V_\infty = 0.005$.

First, we show the results of just the SU2 forced-heaving simulations for different Mach numbers in Figures 2.8 and 2.9. Both the phase and the magnitude variation of the circulation and the $x$-doublet are seen to significantly depend on the forcing reduced frequency $\tilde{\omega}$, and its effect varies with the baseline Mach number $M_\infty$. For transonic flow, the dependence of the $x$-doublet phase on frequency is especially strong. A physical explanation of this behavior is that the delays in propagation of the shock and acoustic waves which comprise the $\sigma$-field variation will be strongly affected by the size of the supersonic zone, and hence by $M_\infty$.

These CFD results for different Mach numbers and different reduced frequencies are then used to calibrate a low-order model. In this work, we focus on fitting one low-order model per Mach number for reduced frequencies in the range of $\tilde{\omega} = 0.05$ to $\tilde{\omega} = 0.40$. In the DMD formulation of Section 2.2, we require that all CFD simulations use the same fixed time step $\Delta t$. Therefore, the low frequency simulations use more time

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figures/figure2.png}
\caption{Phase diagram of circulation strength $\Gamma$ for forced heaving at different reduced frequencies for subsonic and transonic flow for the RAE2822 airfoil with $\alpha_0 = 1.5^\circ$ ($M_{\text{crit}} = 0.648$).}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figures/figure3.png}
\caption{Phase diagram of $x$-doublet strength $\kappa_x$ for forced heaving at different reduced frequencies for subsonic and transonic flow for the RAE2822 airfoil with $\alpha_0 = 1.5^\circ$ ($M_{\text{crit}} = 0.648$).}
\end{figure}
steps than the high frequency simulations. Throughout this work, we set the time step such that for the highest reduced frequency 40 time steps per period are used.

For the calibration we combine the forced pitching and heaving simulations to yield one fit per Mach number and airfoil. Applying the DMD method of Section 2.2 to these CFD results yields the fits shown in Figures 2.10 and 2.11 for pitching motion, and Figure 2.12 for heaving motion. The low-order model captures the circulation response quite accurately. Capture of the $x$-doublet response is quite reasonable as well, except for the final part after we stop forcing the airfoil. This is likely due to the numerical noise in the SU2 computation, which then also throws off the fit.
We also show the errors in the fit for different Mach numbers in Tables 2.1a and 2.1b. The error in each state is computed as

$$\epsilon_i = \frac{\|x_{i,DMD} - x_{i,SU2}\|_2}{\|x_{i,SU2}\|_2}.$$ 

The results in Tables 2.1a and 2.1b confirm that the fit for circulation is significantly better than the fit for the x-doublet.

<table>
<thead>
<tr>
<th>$\bar{\omega}$</th>
<th>Error $\Gamma$, %</th>
<th>Error $\kappa_x$, %</th>
<th>$\bar{\omega}$</th>
<th>Error $\Gamma$, %</th>
<th>Error $\kappa_x$, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>2.8</td>
<td>7.7</td>
<td>0.05</td>
<td>4.9</td>
<td>4.1</td>
</tr>
<tr>
<td>0.10</td>
<td>2.5</td>
<td>5.7</td>
<td>0.10</td>
<td>3.8</td>
<td>5.9</td>
</tr>
<tr>
<td>0.15</td>
<td>2.9</td>
<td>11.5</td>
<td>0.15</td>
<td>2.5</td>
<td>14.4</td>
</tr>
<tr>
<td>0.20</td>
<td>2.7</td>
<td>16.6</td>
<td>0.20</td>
<td>2.8</td>
<td>4.8</td>
</tr>
<tr>
<td>0.25</td>
<td>2.3</td>
<td>16.2</td>
<td>0.25</td>
<td>5.8</td>
<td>12.4</td>
</tr>
<tr>
<td>0.30</td>
<td>1.5</td>
<td>12.1</td>
<td>0.30</td>
<td>3.6</td>
<td>16.1</td>
</tr>
<tr>
<td>0.35</td>
<td>1.3</td>
<td>13.0</td>
<td>0.35</td>
<td>1.5</td>
<td>12.9</td>
</tr>
<tr>
<td>0.40</td>
<td>3.9</td>
<td>19.5</td>
<td>0.40</td>
<td>5.8</td>
<td>16.3</td>
</tr>
</tbody>
</table>

(a) $M_\infty = 0.30$ at $\alpha_0 = 1.5^\circ$, $M_{\text{crit}} = 0.648$ (b) $M_\infty = 0.70$ at $\alpha_0 = 1.5^\circ$, $M_{\text{crit}} = 0.648$

The goal of a calibrated low-order model is to be predictive at different frequencies. To demonstrate this, we calibrate one low-order model for $M_\infty = 0.30$ and one low-order model for $M_\infty = 0.70$, and test these low-order models at frequencies that were not part of the frequencies used for calibration. Table 2.2 show the accuracy of the DMD fits for these Mach numbers. We see that the accuracy for these frequencies is quite similar to the accuracy for frequencies that were part of the calibration (Tables 2.1a and 2.1b).

We fit low-order models for different freestream Mach numbers to investigate the effect of compressibility on the aerodynamic response of the airfoil. The effect of Mach number is shown by the magnitude and phase of the transfer function between circulation and the pitching angle $\theta$ in Figure 2.13, where we also show the result for Theodorsen’s theory. We see that the magnitude of the transfer function is strongly dependent...
on Mach number, as expected, but that the Mach number influence becomes smaller as the reduced frequency increases. The magnitude of the transfer function is also quite different between subsonic and transonic Mach numbers. Furthermore, the phase lag increases with subsonic Mach number and the Mach number dependence becomes stronger as the reduced frequency increases. However, for high transonic Mach numbers—e.g., $M_\infty = 0.85$—the phase lag suddenly becomes smaller. The trends in these Bode plots are similar to Theodorsen theory, but the values are noticeable different, even for low Mach numbers. This is likely due to considering an airfoil with thickness rather than a flat plate.

This low-order model can be used to investigate the effect of Mach number on the flutter boundary of the RAE2822 airfoil. Low-order models of the form Eq. (2.12) are fit using DMD at different freestream Mach numbers. For a given Mach number, the aerodynamic low-order model is coupled to the structural model—with airfoil parameters as listed in Table 2.3—to obtain the full aeroelastic state-space model in Eq. (2.12).

<table>
<thead>
<tr>
<th>$\bar{\omega}$</th>
<th>Error $\Gamma$, %</th>
<th>Error $\kappa_{xy}$, %</th>
<th>$\bar{\omega}$</th>
<th>Error $\Gamma$, %</th>
<th>Error $\kappa_{xy}$, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.125</td>
<td>2.7</td>
<td>8.5</td>
<td>0.125</td>
<td>3.6</td>
<td>17.2</td>
</tr>
<tr>
<td>0.175</td>
<td>2.9</td>
<td>14.4</td>
<td>0.175</td>
<td>2.2</td>
<td>9.7</td>
</tr>
<tr>
<td>0.225</td>
<td>2.5</td>
<td>17.1</td>
<td>0.225</td>
<td>4.9</td>
<td>6.6</td>
</tr>
<tr>
<td>0.275</td>
<td>1.9</td>
<td>14.5</td>
<td>0.275</td>
<td>4.9</td>
<td>19.0</td>
</tr>
<tr>
<td>0.325</td>
<td>1.1</td>
<td>10.9</td>
<td>0.325</td>
<td>2.4</td>
<td>13.2</td>
</tr>
<tr>
<td>0.375</td>
<td>2.4</td>
<td>16.7</td>
<td>0.375</td>
<td>3.4</td>
<td>9.8</td>
</tr>
</tbody>
</table>

(a) $M_\infty = 0.30$ at $\alpha_0 = 1.5^\circ$, $M_{crit} = 0.648$

(b) $M_\infty = 0.70$ at $\alpha_0 = 1.5^\circ$, $M_{crit} = 0.648$

Table 2.2: Accuracy of DMD fit for forced pitching oscillations at frequencies that were not part of the DMD calibration for RAE2822 airfoil.

<table>
<thead>
<tr>
<th>Airfoil parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apparent-mass ratio</td>
<td>$\mu = 4m/\pi \rho_\infty c^2$</td>
</tr>
<tr>
<td>Inertia/mass ratio</td>
<td>$r_\theta = 4I_\theta/mc^2$</td>
</tr>
<tr>
<td>Elastic axis position</td>
<td>$x_{ea}/c$</td>
</tr>
<tr>
<td>In-vacuo heave natural frequency</td>
<td>$\omega_h = \sqrt{k_h/m}$</td>
</tr>
<tr>
<td>In-vacuo pitch natural frequency</td>
<td>$\omega_\theta = \sqrt{k_\theta/I_\theta}$</td>
</tr>
</tbody>
</table>

Table 2.3: Airfoil parameters used for flutter calculations in Figure 2.14, following Ref. 121.
Here, we focus on the flutter speed as a function of the center of gravity position. The eigenvalues of \((E^{-1}A)\) for a given \(x_{cg}\) are a function of \(V_\infty\). A bisection method is used to find that flutter speed index \(V_\mu\) for which the aeroelastic system becomes unstable, the results are shown in Figure 2.14. The flutter speed index \(V_\mu\) is a nondimensionalization of the freestream velocity at which flutter occurs, and is defined as \(V_\mu = 2V_\infty/\sqrt{\mu o c}\). This bisection search is performed online and only involves solving \(7 \times 7\) eigenvalue problems. Constructing one flutter boundary in Figure 2.14 for sixty different \(x_{cg}\) locations therefore only takes \(\sim 0.1\) s. A more detailed explanation of computing the flutter boundary is provided in Appendix A.

Figure 2.14 shows that for a center of gravity position far forward, a higher subsonic Mach number lowers the flutter speed. However, as the center of gravity moves aft, a higher subsonic Mach number increases the flutter speed. The behavior for transonic Mach numbers is quite different, especially for \(x_{cg}\) far forward. For instance, for \(x_{cg}\) far forward, the flutter speed for \(M_\infty = 0.80\) and \(M_\infty = 0.85\) is higher than for \(M_\infty = 0.70\), which is classic transonic flutter behavior and commonly described as the transonic dip.

2.3.2 VALIDATION FOR ISOGAI CASE A

We validate the accuracy of the model for the standard flutter benchmark Isogai Case A.\(^{33,122}\) We compare our results for the flutter boundary against four methods from the literature\(^{123-126}\) and against the flutter boundary found using SU2—using the same mesh as was used to calibrate the model—in Figure 2.15. We see that the agreement for the transonic flow regime is quite good up to \(M_\infty = 0.9\). The discrepancy between our model and SU2 is likely a result of numerical dissipation in the SU2 solver.

Isogai Case A is known for the characteristic S-shape of its flutter boundary. For high transonic Mach numbers, the airfoil becomes unsta-

![Figure 2.14: Flutter speed versus \(x_{cg}/c\) positions for different freestream Mach numbers.](image-url)
ble but if the flutter speed index is increased further, it becomes stable again, until it finally becomes unstable again around $V_\mu \approx 2.8$. Our model is intended to yield flutter predictions of optimal designs, as part of an aircraft design code. A good design should never operate in the second stability region and therefore it is not a problem that the model cannot capture this S-shape of the flutter boundary.

![Graph](image)

**Figure 2.15:** Comparison of transonic flutter boundary for Isogai Case A$^{3,122}$ as found by our method and methods from literature.$^{123-126}$

2.4 **INFLUENCE OF AIRFOIL PARAMETERS ON FLUTTER**

Airfoil parameters have a large influence on the flutter behavior of airfoils, as is demonstrated in the following. Specifically, we discuss the influence of the freestream Mach number $M_\infty$, airfoil family and baseline lift coefficient $c_\ell_0$, which are all inputs to the low-order aerodynamic model calibration.

![Graph](image)

**Figure 2.16:** Influence of baseline lift coefficient $c_\ell_0$ on the transonic flutter boundary for the 11% C-airfoil. For this case $\mu = 40$, $\omega_0/\omega_h = 0.9$, $r_0^* = 2.5$, $x_{ef}/c = -0.35$, $x_{cg}/c = 0.35$.

$^3$Note that the elastic axis is in front of the leading edge for this airfoil section—similar to Isogai Case A—to represent the dynamics of a swept-back wing section.
ibrated, whereas the lines in between are generated using splines for the entries in the aerodynamic state-space matrix. For low subsonic freestream Mach numbers, the dependence of the flutter boundary on baseline lift coefficient is small, however for the transonic regime, the baseline lift coefficient has a larger influence on the transonic flutter boundary. For this airfoil, a lower $c_{l_0}$ is associated with a higher flutter speed index for subsonic Mach numbers, but a lower flutter speed index for high transonic Mach numbers.

Bendiksen showed that the airfoil thickness can have a large effect on the flutter boundary in the transonic flow and that linear theory predicts that thinner wings would be less susceptible to flutter, but in practice the opposite has been observed in the transonic flow regime.$^{15}$ A flutter boundary for airfoils with different thicknesses of the same airfoil family is shown in Figure 2.18. These results indeed confirm that for subsonic freestream Mach numbers, a thicker airfoil is more susceptible to flutter. However, a thinner airfoil is more susceptible to flutter for high transonic freestream Mach numbers—in line with Bendiksen’s observations.

To highlight the sensitivity of the model to a particular airfoil, we compare the flutter boundary for two different transonic airfoils, which are both used in the design of the D8.x aircraft$^{127}$ (Figure 2.19). These

\[ \tau = 9\% \quad \tau = 10\% \quad \tau = 11\% \quad \tau = 12\% \quad \tau = 13\% \quad \tau = 14\% \]

Figure 2.17: Overview of C-airfoil family, used in the design of the D8.x aircraft.

Figure 2.18: Influence of airfoil thickness on the transonic flutter boundary for the C-airfoil family at $c_{l_0} = 0.6$. For this case $\mu = 40$, $\omega_0/\omega_h = 0.9$, $r_0^2 = 2.5$, $x_{ea}/c = -0.35$, $x_{cg}/c = 0.35$.

Figure 2.19: Comparison between C-airfoil and E-airfoil (12%).
airfoils are quite similar, and Figure 2.20 shows that their flutter boundary is almost the same for subsonic Mach numbers. However, for the transonic flow regime their flutter characteristics are quite different as the transonic dip for the E-airfoil is less strong than for the C-airfoil. For Mach numbers around 0.85, the flutter boundaries are essentially the same again. This demonstrates that the physics-based low-order model is able to pick up on the influence of slight differences in airfoil characteristics on the transonic flutter boundary.

![Figure 2.20: Comparison of flutter boundary between the E-airfoil and C-airfoil, both 12% thick at $c_{t0} = 0.6$. For this case $\mu = 40$, $\omega_0/\omega_h = 0.5$, $r_0^2 = 2.5$, $x_{e0}/c = -0.3$, $x_{cg}/c = 0.35$.](image)
TRANSONIC FLUTTER PREDICTION
FOR AIRCRAFT WINGS

The physics-based transonic flutter model for airfoils (Chapter 2) can also be applied to aircraft wings by applying corrections for three-dimensional flow and by using a beam structural model (Section 3.1). The accuracy of this transonic flutter prediction method is demonstrated by comparing the flutter boundary found using this model to wind tunnel results and higher-fidelity data from the literature (Section 3.2). To quantify the influence of transonic flutter on the overall aircraft design, this wing flutter model is implemented in a conceptual aircraft design tool (Section 3.3). The model is used to describe the influence of different wing configurations in transonic flow on the flutter boundary (Section 3.4) both for clamped wings and weight-converged aircraft (using the conceptual design tool), where it is found that transonic flow changes the aeroelastic behavior drastically, especially for sweep angle and engine weight variations. The conceptual design tool combined with the flutter model is used to investigate the influence of transonic flutter on the overall aircraft design (Section 3.5), where it is found that transonic flutter can limit the fuel efficiency gains from using better material technology, as this leads to more flexible wings and therefore more flutter problems.

3.1 WING FLUTTER MODEL

The aerodynamic model for airfoils in transonic flow, developed in Chapter 2, is extended to three-dimensional wings using strip theory (Section 3.1.1). This 3D aerodynamic model is coupled to a beam model (Section 3.1.2) to compute flutter characteristics of aircraft wings.

3.1.1 AERODYNAMIC MODEL

Three of the most common methods to predict unsteady loads on an aircraft are strip theory (2D unsteady airfoil theory with 3D corrections), the doublet-lattice method, and the unsteady vortex-lattice method (UVLM). Considering that our goal is to investigate aircraft concepts with high aspect ratios, the use of strip theory here is appropriate. In strip theory, one assumes that the flow along a cut in the wing—perpendicular to a span-wise axis of the wing—is two-dimensional (Figure 3.1). For this work, that means we can use our calibrated two-

129. Fung, An Introduction to the Theory of Aeroelasticity. 1993
130. Wright et al., Introduction to Aircraft Aerelasticity and Loads. 2015
132. Katz et al., Low-Speed Aerodynamics. 2001
dimensional flutter model (Chapter 2) also for three-dimensional wings. For our model, we therefore use the perpendicular-plane Mach number $M_\perp$ and lift coefficient $c_\ell_\perp$ for selecting the appropriate coefficients for the transonic aerodynamics model.

Commercial transport aircraft commonly have swept wings to enable higher cruise Mach numbers. We consider the wing geometry in Figure 3.2 with sweep angle $\Lambda$. The effective length of the wing $\bar{l} = l / \cos \Lambda$, and all section parameters such as chord length, moment of inertia, etc., are based on sections perpendicular to the elastic axis. We therefore assume that the elastic axis of the wing is straight.

For the aerodynamics model, we must account for the changes to the velocity distribution due to the swept wing. The vertical velocity at any point on the wing is found to be $v_{\perp}$.

---

25 Barmby et al., *Study of Effects of Sweep on the Flutter of Cantilever Wings*, 1951
\[ U_n(\bar{x}, t) = w + V_\infty \frac{\partial h}{\partial y} \sin \Lambda + V_\infty \theta \cos \Lambda + \bar{x} \omega + \bar{x} V_\infty \frac{\partial \theta}{\partial y} \sin \Lambda, \quad (3.1) \]

where \( \bar{x}, \bar{y} \) are the coordinates aligned with the swept wing. From Eq. (3.1), Barmby et al.\(^{25}\) found the circulatory lift per span for swept wings in incompressible flow to be

\[ L_c = \pi \rho V_\perp c C_{th}(\bar{\omega}) \left[ w + V_\perp \frac{\partial h}{\partial y} \tan \Lambda + V_\perp \theta + \frac{c}{4} \left( \omega + V_\perp \frac{\partial \theta}{\partial y} \tan \Lambda \right) \right], \]

where \( V_\perp \) is the wing-perpendicular velocity and \( C_{th}(\bar{\omega}) \) is Theodorsen’s function\(^{21}\) with \( \bar{\omega} \) the reduced frequency. Following the same approach, here the evolution equation for the circulation strength is defined to be,

\[ \Delta \Gamma = -\frac{A_T}{A_\Gamma} A_{\Gamma} \frac{\Delta \Gamma}{\Delta \kappa} + \frac{1}{A_\Gamma} \left( \Delta w + V_\perp \frac{\partial \Delta h}{\partial y} \tan \Lambda \right) + \frac{V_\perp}{A_\Gamma} \Delta \theta + \frac{c/4}{A_\Gamma} \left( \Delta \omega + V_\perp \frac{\partial \Delta \theta}{\partial y} \tan \Lambda \right). \]

Lastly, the non-circulatory forces and moments used in the model are\(^{25}\)

\[ \Delta L_{nc} = \frac{1}{4} \pi \rho_\infty c^2 \left[ \Delta \dot{h} + V_\perp \frac{\partial \Delta h}{\partial y} \tan \Lambda + V_\perp \Delta \dot{\theta} \right] - \frac{1}{4} \pi \rho_\infty c^2 \left( x_{ea} - \frac{c}{2} \right) \left[ \Delta \ddot{\theta} + V_\perp \frac{\partial \Delta \theta}{\partial y} \right], \]

\[ \Delta M_{nc} = \frac{1}{4} \pi \rho_\infty c^2 V_\perp \left[ \left( \frac{3}{4} c - x_{ea} \right) \Delta \dot{\theta} + \frac{1}{2} V_\perp \frac{\partial \Delta \theta}{\partial y} \tan \Lambda \right] - \frac{\rho_\infty \pi c^4}{128} \left[ \Delta \ddot{\theta} + V_\perp \frac{\partial \Delta \theta}{\partial y} \tan \Lambda \right] + \frac{1}{4} \rho_\infty \pi c^2 \left( x_{ea} - \frac{c}{2} \right) \left[ \Delta \dot{h} + V_\perp \frac{\partial \Delta h}{\partial y} \tan \Lambda \right] - \frac{1}{4} \rho_\infty \pi c^2 \left( x_{ea} - \frac{c}{2} \right)^2 \left[ \Delta \dot{\theta} + V_\perp \frac{\partial \Delta \theta}{\partial y} \tan \Lambda \right]. \]

Note that this formulation ignores the wing camber effect in the noncirculatory forces and moments, which is known to have a negligible effect on flutter.\(^{25}\)

Finally, note that—using the same assumptions as Barmby et al.\(^{25}\)—this unsteady aerodynamics model ignores (1) spatial variations in the trailing vorticity, (2) any variation of the airloads in the spanwise direction, and (3) any three-dimensional wing tip effects. These assumptions are quite reasonable for high reduced frequencies, because in these cases the wake does not have time to newly form with each oscillation.\(^*\) Flutter is typically observed for higher reduced frequencies, making these assumptions appropriate for this work.

\(^*\)This is typically the case for \( \bar{\omega} > 0.05 \) for medium to high aspect ratio wings.
For the structural part of the model, we use Bernoulli-Euler beam theory. Following Bisplinghoff et al.,\textsuperscript{113} the beam equations are,

\begin{align*}
m(\bar{y})\ddot{h}(\bar{y}, t) + S_{\gamma}(\bar{y}) \Delta \dot{\theta}(\bar{y}, t) + \frac{\partial^2}{\partial \bar{y}^2} \left[ EI(\bar{y}) \frac{\partial^2 \Delta h(\bar{y}, t)}{\partial \bar{y}^2} \right] &= -\Delta L(\bar{y}, t) \\
I_{\gamma}(\bar{y}) \Delta \dot{\theta}(\bar{y}, t) + S_{\gamma}(y) \Delta \dot{\theta}(\bar{y}, t) - \frac{\partial}{\partial \bar{y}} \left[ GJ(\bar{y}) \frac{\partial \Delta \theta(\bar{y}, t)}{\partial \bar{y}} \right] &= \Delta M_{ea}(\bar{y}, t)
\end{align*}

where \(m(\bar{y})\) is the mass per span, \(S_{\gamma}(\bar{y})\) is the mass unbalance per span, \(I_{\gamma}(\bar{y})\), \(\Delta \dot{\theta}(\bar{y}, t)\) is the mass moment of inertia per span, \(EI(\bar{y})\) is the bending stiffness, and \(GJ(\bar{y})\) is the torsional stiffness. \(L\) is the lift per span, and \(M_{ea}\) is the aerodynamic moment around the elastic axis per span. The states for this equation are the vertical displacement perturbation (positive downward) \(\Delta h\), and the pitch angle perturbation \(\Delta \theta\). The appropriate boundary conditions here are,

\begin{align*}
\Delta h(0, t) &= 0, & \frac{\partial \Delta h}{\partial \bar{y}}(0, t) &= 0, & \frac{\partial^2 \Delta h}{\partial \bar{y}^2}(\bar{I}, t) &= 0, & \frac{\partial^3 \Delta h}{\partial \bar{y}^3}(\bar{I}, t) &= 0, \\
\Delta \theta(0, t) &= 0, & \frac{\partial \Delta \theta}{\partial \bar{y}}(\bar{I}, t) &= 0.
\end{align*}

Figure 3.3: Discretized beam model used in structural part of the flutter model.
The beam equations can be rewritten as

\[
m(\bar{y}) \Delta \ddot{h}(\bar{y}, t) - S(\bar{y}) \Delta \ddot{\theta}(\bar{y}, t) + \frac{\partial \Delta S(\bar{y}, t)}{\partial \bar{y}} = -\Delta L(\bar{y}, t) \tag{3.2a}
\]

\[
I(\bar{y}) \Delta \ddot{\theta}(\bar{y}, t) + S(\bar{y}) \Delta \ddot{h}(\bar{y}, t) - \frac{\partial \Delta T(\bar{y}, t)}{\partial \bar{y}} = \Delta M_{ea}(\bar{y}, t) \tag{3.2b}
\]

where

\[
\frac{\partial \Delta M}{\partial \bar{y}} = \Delta S, \quad \frac{\partial \Delta \gamma}{\partial \bar{y}} = \frac{\Delta M}{EI}, \quad \frac{\partial \Delta h}{\partial \bar{y}} = \Delta y, \quad \frac{\partial \Delta \theta}{\partial \bar{y}} = \frac{\Delta T}{GJ},
\]

with boundary conditions

\[
\Delta h(0, t) = 0, \quad \Delta y(0, t) = 0, \quad \Delta M(\bar{I}, t) = 0, \quad \Delta S(\bar{I}, t) = 0,
\]

\[
\Delta \theta(0, t) = 0, \quad \Delta T(\bar{I}, t) = 0,
\]

where \( M \) is the internal bending moment, \( S \) is the internal shear force, \( T \) is the internal torsion moment, and \( \gamma \) is the local dihedral angle.

To solve the system in Eq. (3.2), the wing is discretized in several sections, as shown in Figure 3.3. The sectional properties are all computed using the assumed wing box shape as shown in Figure 3.4. The thickness ratio \( \tau \) is used to determine the coefficients of the aerodynamic model, as explained in Section 2.2. It is assumed that the elastic axis is at the center of the wing box. When the flutter model is used in a design setting, the wing box design variables such as spar and skin thicknesses are optimized in a conceptual design tool, and their values are then inputs to the flutter model.

The system in Eq. (3.2) is solved using finite differences, specifically the trapezoidal rule. However, Eq. (3.2) has \( 6 \times n_{beam} \) structural parameters in the aeroelastic system, of which only two have inertial terms (\( \Delta h \) and \( \Delta \theta \)). Such a large system slows down the eigenvalue computation. Instead, we can reduce the size of the system through Schur complements. Consider the discretized system of Eq. (3.2),

\[
E \begin{bmatrix} \Delta h \\ \Delta \dot{y} \\ \Delta \dot{M} \\ \Delta \dot{S} \\ \Delta \dot{\theta} \\ \Delta \dot{T} \end{bmatrix} + \bar{A} \begin{bmatrix} \Delta h \\ \Delta y \\ \Delta M \\ \Delta S \\ \Delta \theta \\ \Delta T \end{bmatrix} = \begin{bmatrix} -\Delta L \\ 0 \\ 0 \\ 0 \\ \Delta M_{ea} \\ 0 \end{bmatrix}, \tag{3.3a}
\]
where

$$\bar{E} = \begin{bmatrix} \bar{E}_{hh} & 0 & 0 & 0 & \bar{E}_{h\theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{E}_{\theta h} & 0 & 0 & 0 & \bar{E}_{\theta \theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(3.3b)

$$\bar{A} = \begin{bmatrix} \bar{A}_{hh} & \bar{A}_{h\theta} & 0 & 0 \\ \bar{A}_{\theta h} & \bar{A}_{\theta \theta} & 0 & 0 \\ 0 & 0 & \bar{A}_{MM} & 0 \\ 0 & 0 & \bar{A}_{SM} & \bar{A}_{SS} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(3.3c)

The matrices \(\bar{A}_{hh}, \bar{A}_{MM}, \bar{A}_{SM},\) and \(\bar{A}_{TT}\) are all tightly banded and therefore cheap to invert. This allows for taking the Schur complement of the system in Eq. (3.3) to obtain a system with only \(h\) and \(\theta\) as states,

$$\begin{bmatrix} \bar{E}_{hh} & \bar{E}_{h\theta} \\ \bar{E}_{\theta h} & \bar{E}_{\theta \theta} \end{bmatrix} \begin{bmatrix} \Delta h \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} A_{hh} & 0 \\ 0 & A_{\theta \theta} \end{bmatrix} \begin{bmatrix} \Delta h \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} -\Delta L \\ \Delta M_{ea} \end{bmatrix}.$$  

(3.4)

This structural model is then coupled to aerodynamic model described in Section 3.1.1 through the aerodynamic lift and moment, yielding a \(7 \times n_{\text{beam}}\) descriptor state-space system.

The engine adds a large amount of inertia and mass to the wing, and it is therefore important to represent in a flutter model. The engine is modeled as a point mass \(m_{\text{eng}}\) with inertia \(\bar{I}_{\text{eng}}\) and angular momentum \(\bar{H}_{\text{eng}}\), which are rotated into the swept coordinate system. As with the rest of the model, we only consider perturbations from the steady state. The thrust and moment generated by the engine are therefore not taken into account.

The influence of the engine on the structural model comes through a discontinuity in \(\Delta S, \Delta M,\) and \(\Delta T\) at the section to which the engine is attached, as shown in Figure 3.5. These discontinuities are captured with a zero-length grid interval at the engine attachment point.

The inertial-reaction forces and moments on the \(i\)th section on the wing due to the engine are,

$$\begin{align*}
\Delta F_{\text{eng}} &= -m_{\text{eng}} \Delta a_{\text{eng}} \\
\Delta M_{\text{eng}} &= \Delta r_{\text{eng}} \times \Delta F_{\text{eng}} - \Delta \omega_{\text{eng}} \times \bar{H}_{\text{eng}} \\
\Delta F_{\text{eng}} &= -\bar{I}_{\text{eng}} \Delta \omega_{\text{eng}} - \bar{I}_{\text{eng}} \Delta \omega_{\text{eng}} \times \bar{I}_{\text{eng}} \Delta \omega_{\text{eng}} \\
\end{align*}$$

(3.5a)

(3.5b)

where \(\Delta r_{\text{eng}} = [\Delta \bar{x}_{\text{eng}}, \Delta \bar{y}_{\text{eng}}, \Delta \bar{z}_{\text{eng}}]^T\) is the distance vector between the \(i\)th beam section and the engine, and \(\Delta F_{\text{eng}}\) and \(\Delta M_{\text{eng}}\) are the force and
moment, respectively, on the $i$th beam section. $\Delta \omega_i$ is the rotation rate perturbation of the $i$th beam section with respect to the aircraft body axes, defined as $\Delta \omega_i = [-\Delta \dot{y}, \Delta \dot{\theta}, 0]^T$. We observe that $\Delta \omega_i \times H_{\text{eng}}$ yields only an in-plane bending moment for unswept wings, while in-plane motion is not modeled here. Note that the last term in Eq. (3.5b) is a quadratic term in the perturbation rates, and can therefore be neglected. Finally, $\Delta a_{\text{eng}}$ is the inertial acceleration perturbation of the engine, defined as

$$\Delta a_{\text{eng}} = \Delta a_i + \Delta \dot{\omega}_i \times \Delta r_{\text{eng}} + \Delta \omega_i \times \left( \Delta \omega_i \times \Delta r_{\text{eng}} \right),$$

where $\Delta a_i$ is the acceleration perturbation of the $i$th beam section. Again, the last term in Eq. (3.5c) is a higher-order term in the perturbation rates and hence is omitted. Note that in this formulation, the aircraft fuselage is assumed to not participate significantly in the flutter modes.

The discontinuity in the shear force and bending moments results in additional terms in $E_{hh}, E_{h\theta}, E_{\theta h},$ and $E_{\theta\theta}$ in Eq. (3.4).

3.2 VALIDATION OF WING FLUTTER MODEL

The accuracy of the flutter model is demonstrated by comparing the computed aerodynamic response of a moderate aspect ratio wing in
transonic flow against a high-fidelity unsteady 3D Euler simulation and by comparing its flutter predictions against experimental data for a well-known benchmark case.

3.2.1 AERODYNAMIC RESPONSE OF PITCHING WING

Here, we demonstrate the accuracy of the aerodynamic model for a pitching wing with moderate aspect ratio. First, we calibrate the low-order model for the wing’s airfoil—in this case the NASA SC(2)-0414 airfoil (Figure 3.6)—using the methodology described in Chapter 2, with the data obtained from high-fidelity 2D unsteady Euler simulations. The results from this aerodynamic low-order model are then compared against the sectional lift coefficients obtained from a full 3D unsteady Euler simulation around the wing. The unsteady flow field around the wing is computed using SU2.\textsuperscript{134}

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure3_6}
\caption{Mach number field around NASA SC(2)-0414 airfoil at $M_{\infty} = 0.74$ and $\alpha = 0^\circ$. The NASA SC(2)-0414 airfoil is a 14\% thick supercritical airfoil, and is also used in the Benchmark Supercritical Wing (BSCW).}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure3_7}
\caption{Sectional lift coefficient perturbation at 60\% semi-span versus pitch angle for a wing with a NASA SC(2)-0414 airfoil and $AR = 8$.}
\end{figure}

A comparison between the low-order aerodynamic model and the SU2 solution is shown in Figure 3.7 for $M_{\infty} = 0.74$ and $\alpha = 0^\circ$. At these conditions, the NASA SC(2)-0414 airfoil has a large supersonic region over the upper surface (Figure 3.6), as the critical Mach number for the NASA SC(2)-0414 airfoil at $\alpha = 0^\circ$ is 0.648. The results in Figure 3.7 show good agreement in both magnitude and phase.

\textsuperscript{134} Economon et al., SU2: An Open-Source Suite for Multiphysics Simulation and Design. 2016
3.2.2 Flutter of Benchmark Supercritical Wing

The accuracy of the presented model is demonstrated using experimental results from the Benchmark Supercritical Wing (bscw). The bscw is also used for the Aeroelastic Prediction Workshop. The bscw wind tunnel model uses the NASA sc(2)-0414 airfoil and has no sweep or taper; further geometric parameters are described in Ref. 138. An illustration of the bscw in the wind tunnel is shown in Figure 3.8. Note that this model only has an aspect ratio of 4, which is substantially lower than applications for which this flutter model is intended.

The wind tunnel model is tested using the Pitch and Plunge Apparatus (PAPA) in the NASA Langley Transonic Dynamics Tunnel. The model itself is rigid; all motion is therefore controlled by the PAPA. The appropriate structural model for this test case is therefore a typical section model, as also described in Chapter 2.

Figure 3.8: The Benchmark Supercritical Wing (bscw) has an aspect ratio of 4 and is tested using a pitching and plunging apparatus.

Figure 3.9: Comparison between experimental flutter points, flutter points found using CFD, and flutter points found using the current method for the bscw.
quite well with the experimental results. For high transonic Mach numbers \( M_\infty > 0.80 \), the fit is also reasonable. The flutter boundary from the presented method exhibits a sharper transonic dip, whereas the wind tunnel results barely show a transonic dip, however the errors are still within 5%. We do note that the CFD results (full 3D unsteady RANS simulations) from the Aeroelastic Prediction Workshop shown in Figure 3.9 exhibit 5% or higher errors; thus our low-order model compares favorably with these high-fidelity benchmark results.

3.3 CONCEPTUAL AIRCRAFT DESIGN TOOL

As an illustration of how the flutter model can be incorporated into a conceptual design framework, we use the Transport Aircraft System OPTimization (TASOPT) tool for the conceptual design of novel aircraft concepts. TASOPT was developed at MIT as part of a NASA project to design aircraft to meet aggressive fuel burn, noise, and emission reduction goals for the 2035 timeframe. To assess the performance of next-generation aircraft concepts, this conceptual design tool is primarily developed from first principles rather than using historical correlations. It uses low-order physical models implementing fundamental structural, aerodynamic, and thermodynamic theory, and relies on historical correlations only for the weight of secondary structure and aircraft equipment.

TASOPT can both size the aircraft for a particular mission (range and payload) and optimize the aircraft by varying, for instance, cruise altitude, cruise lift coefficient, aspect ratio, wing sweep, etc., to minimize mission fuel burn. TASOPT can therefore be used to model existing aircraft, assess their off-design performance, and perform a sensitivity analysis for an aircraft. TASOPT is also used to assess the influence of new technologies, such as advanced materials, on an airframe design. Finally, this tool is used to design entirely new aircraft for a set of missions.

As with most aircraft conceptual design tools, dynamic aeroelasticity is currently not a design consideration in TASOPT. To assess the influence of transonic flutter on the conceptual design of next-generation aircraft, the developed transonic flutter model is implemented in TASOPT here.

Flutter is indicated using the eigenvalues of the aeroelastic system—the overall system that consists of the wing aerodynamics model and beam model. If the largest real part of the eigenvalues of the aeroelastic system—denoted here as \( \chi_{fl} \)—becomes positive, flutter occurs. We can include flutter constraints in a conceptual design tool by constraining the value of \( \chi_{fl} \) for several different points in the flight envelope.

The design routine in this conceptual design tool wraps an optimizer around a weight-convergence routine. During the optimization several design parameters, such as cruise altitude, wing sweep, aspect ratio, etc., are varied to minimize the fleet-wide fuel energy consumption per
payload-range. Constraints can be incorporated in this optimization loop, such as minimum balanced field length and maximum wing span. Here, we include the requirement for no flutter occurring for \( K \) operating points in the design process as a sum of constraints in this optimization loop via a penalty term in the objective function,

\[
f(x) = \tilde{f}(x) + \nu \sum_{k=1}^{K} \{ \max\{0, \chi_{\parallel,k} - \chi_{\parallel,max}\}\}^2,
\]

where \( x \) are the design variables, \( f(x) \) is the overall objective function, \( \tilde{f}(x) \) is a function that includes mission fuel burn with penalty terms for, e.g., balanced field length and maximum wing span, and \( \nu \) is a suitable weighting factor. \( \chi_{\parallel,k} \) is the largest real part of the eigenvalues of the aeroelastic system for the \( k \)th operating point, and \( \chi_{\parallel,max} \) is the largest value tolerated, typically \(-0.005/s\).

We include the flutter model in \textsc{tasopt} using a database of flutter coefficients with \( c_{\ell_0} \), \( M_\infty \), \( \tau \) as the database independent variables. We calibrate 2D low-order models, using the methodology described in Section 2.2, for several thickness ratios \( \tau \) of an airfoil family at different baseline lift coefficients \( c_{\ell_0} \) and Mach numbers \( M_\infty \) and store these in a database. An example of how such a flutter coefficient varies with \( c_{\ell_0} \) and \( M_\infty \) is shown in Figure 3.10. Whenever \textsc{tasopt} queries a flutter evaluation, the appropriate flutter coefficients are found through spline interpolation using the flutter coefficient database.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{image.png}
\caption{A database of flutter coefficients is generated for an airfoil family as a function of \( M_\infty \), \( c_{\ell_0} \), and \( \tau \). This example shows a flutter coefficient for the 11% thick C-airfoil.}
\end{figure}

\textsc{tasopt} uses a parametrized spanwise loading \( p_\parallel(\bar{y}) \), whose shape is either specified, or optimized for the best tradeoff between structural weight and induced drag (computed via a Trefftz-Plane analysis). Here, this same loading is used to obtain the local lift coefficient,

\[
c_{\ell_\perp}(\bar{y}) = \frac{2p_\parallel(\bar{y}) \cos \Lambda}{\rho_\infty V_\infty^2 c_\perp(\bar{y})},
\]

where \( c_\perp(\bar{y}) \) is the local chord length perpendicular to the elastic axis. The wing-perpendicular quantities \( c_{\ell_\perp} \) and \( M_\perp \) are then used to get the
appropriate flutter model for each section of the wing using the flutter coefficient database.

### 3.4 Influence of Wing Geometry on Flutter

The flutter model explained in Section 3.1 is used here to quantify the influence of several parameters of an aircraft wing on the flutter boundary. First, we compute the flutter boundary of a clamped wing in a wind tunnel-like set-up in Section 3.4.1. Second, we use the flutter model coupled with TASSOPT to evaluate the influence of the wing parameters on the flutter damping values in Section 3.4.2 where each aircraft on the flutter boundary is a weight-converged aircraft design.

#### 3.4.1 Clamped Wing

We consider a wind tunnel-like set-up for a straight, swept, tapered wing (Figure 3.11). This problem has twelve nondimensional parameters, which are listed in Table 3.1. Here, the apparent-mass ratio \( \mu \), flutter speed index \( V_\mu \), and radius of gyration \( r_\theta^2 \) are defined as,

\[
\mu = \frac{4m_{\text{smc}}}{\pi \rho_\infty c_{\text{smc}}^2}, \quad V_\mu = \frac{2 V_\infty |\chi_{\beta}=0}{\sqrt{\mu \omega_\theta c_{\text{smc}}}}, \quad r_\theta^2 = \frac{4I_{\text{y,smc}}}{m_{\text{smc}} c_{\text{smc}}^2}
\]

where,

\[
\omega_\theta = \sqrt{\frac{4(GJ)_{\text{smc}}}{m_{\text{smc}} b^2 c_{\text{smc}}^2}}, \quad \omega_h = \sqrt{\frac{(EI)_{\text{smc}}}{m_{\text{smc}} b^4}}.
\]

These non-dimensional parameters determine the planform shape and sectional properties of the Standard Mean Chord (SMC). To obtain the sectional properties over the entire wing, we scale the sectional properties with the local chord length: \( m \sim c^2 \), \( I_\beta \sim c^4 \), \( EI \sim c^4 \), and \( GJ \sim c^4 \).

First, we show the influence of wing taper on the flutter boundary (Figure 3.12). It is observed that the shape of the flutter boundary as

---

Figure 3.11: Geometry for clamped straight swept tapered wing, including the nondimensional parameters of the problem.
Parameter Description
---
\(A\) Aspect ratio
\(M_\infty\) Freestream Mach number
\(c_{t_0}\) Baseline lift coefficient
\(\tau\) Thickness ratio
\(\lambda\) Taper ratio
\(\Lambda\) Sweep angle
\(V_\mu\) Flutter speed index
\((\frac{x_{ea}}{c})\) Relative elastic axis position
\((\frac{x_{cg}}{c})\) Relative center of gravity position
\((\frac{\omega_\theta}{\omega_h})\) Ratio of uncoupled pitch and heave frequencies
\(r_\theta\) Pitch radius of gyration per semi-chord ratio
\(\mu\) Apparent-mass ratio

Table 3.1: Nondimensional parameters for straight, swept, tapered wing.

A function of Mach number is almost independent of the taper ratio and that the taper ratio only determines the offset, with wings having less taper being more susceptible to flutter. This is expected, as taper moves the wing mass inboard and also increases the overall bending and torsional stiffnesses, thus increasing the natural frequency of the wing.

The aspect ratio of the wing has a large influence on the flutter speed of the wing, which is a concern for next-generation aircraft designs. In order to look at a realistic variation of aspect ratio for an aluminum transport aircraft wing, we scale the properties of the wing as 

\[ b \sim \sqrt{A}, \quad c \sim 1/\sqrt{A}, \quad h \sim 1/\sqrt{A}, \quad m \sim A, \quad GJ \sim 1, \quad EI \sim 1, \quad I_y \sim 1. \]

This ensures that the wing has a fixed wing area, fixed lift, and fixed wing box stress.
Note that the nondimensional parameters are not kept constant with this scaling as a function of aspect ratio. Therefore we use the nondimensional parameters of $\mathcal{R} = 15$ to scale the results.

The flutter boundary as a function of aspect ratio is shown in Figure 3.13. We see that a higher aspect ratio reduces the flutter speed substantially, as expected. This influence does, however, plateau for high aspect ratios. Furthermore, the characteristic transonic dip is again observed.

![Flutter boundary](image)

(a) Flutter speed index versus Mach number

![Flutter speed index versus aspect ratio](image)

(b) Flutter speed index versus aspect ratio

The sweep of the wing has a large effect on the flutter boundary, particularly for transonic flows because it lowers the Mach number seen by the section perpendicular to the elastic axis. Flutter boundaries for various sweep angles and Mach numbers are shown in Figure 3.14. We observe that for low subsonic Mach numbers the flutter boundary follows a shifted parabola for incompressible flows, similar to what was found in Ref. 25. That behavior changes completely with transonic flow, though. Whereas for low subsonic flows the sweep angle generally increases the flutter speed, for transonic flows the sweep angle decreases the flutter speed. Figure 3.14a clearly shows that wing sweep delays the transonic dip to higher freestream Mach numbers, which is partly the reason for a decrease in flutter speed for transonic Mach numbers.

When the engine is included in the model, additional nondimensional parameters need to be considered. Specifically, we consider the mass ratio of the engine $\mu_{\text{eng}}$, the radius of gyration of the engine $r_{\text{eng}}$, the relative spanwise location of the engine $\eta_{\text{eng}}$, the relative chord-wise

Figure 3.13: Influence of aspect ratio and freestream Mach number on the flutter boundary. The nondimensional parameters for $\mathcal{R} = 15$ are $\Lambda = 0^\circ, \lambda = 0.75, \mu = 33, \omega_h/\omega_b = 40, r_{\text{eng}} = 0.125, (x_{\text{eng}}/c) = 0.300, (x_{\text{eng}}/c) = 0.301, \tau = 0.13$, and $c_{\text{eng}} = 0.6$ (C-airfoil family).

25. Barmby et al., Study of Effects of Sweep on the Flutter of Cantilever Wings. 1951
Increasing \( \Lambda \)
\[
\Lambda = 0^\circ \\
\Lambda = 10^\circ \\
\Lambda = 15^\circ \\
\Lambda = 20^\circ \\
\Lambda = 25^\circ \\
\Lambda = 30^\circ 
\]

(a) Flutter speed index versus Mach number

(b) Flutter speed index versus sweep angle

location of the engine \( \xi_{\text{eng}, x} \), and the relative normal location of the engine \( \xi_{\text{eng}, z} \). These parameters are defined as

\[
\mu_{\text{eng}} = \frac{m_{\text{eng}}}{m_{\text{smc}} b}
\]
\[
\bar{r}_{\text{eng}}^2 = \frac{4I_{\text{eng}}}{m_{\text{eng}} c_{\text{smc}}^2}
\]
\[
\eta_{\text{eng}} = \frac{y_{\text{eng}}}{b}
\]
\[
\xi_{\text{eng}, x} = \frac{\Delta x_{\text{eng}}}{c_{\text{smc}}}
\]
\[
\xi_{\text{eng}, z} = \frac{\Delta z_{\text{eng}}}{c_{\text{smc}}}
\]

The influence of the mass of the engine on the flutter boundary is shown in Figure 3.15. It is expected that adding engine mass for subsonic flow conditions increases the flutter speed, as the mass of the whole system is increased and the inertial loads increase compared to the aerodynamic loads. However, we observe a similar reversal of trends between low subsonic and transonic flows as for wing sweep; for low subsonic flows the additional mass of the engine increases the flutter speed, whereas for transonic conditions the additional mass suddenly decreases the flutter speed. The location of the transonic dip, however, does not change between different engine masses, the only difference is that the dip gets stronger for higher engine masses.
3.4.2 WEIGHT-CONVERGED AIRCRAFT

The results in Section 3.4.1 showed the influence of wing parameters on the flutter boundary for a wing clamped to the wall. However, for an aircraft design, any change in parameter of the wing cascades through the entire aircraft design, because the aircraft has to be weight converged. In other words, the lift the wing generates has to be equal to the weight of the aircraft. When this is taken into account, the flutter characteristics as a function of wing parameters also change. This section shows the influence of wing parameters on the flutter damping values while ensuring that the aircraft design is weight-converged at each flutter evaluation. These aircraft designs are generated using TSOPT.

We consider two different D8.x aircraft configurations, as described in Ref. 4. The D8.0 and D8.2 configurations are compared in Figure 3.16. The D8.0 is a “fuselage-only” modification to the Boeing 737, keeping the wing similar to the wing of the 737—the engines therefore hang under the wing. The D8.2 is different from the D8.0 in two major ways: the engines are moved to the back of the fuselage to enable boundary layer ingestion and the cruise Mach number is reduced to allow for a lower-sweep wing.

Flutter is evaluated for several worst-case points on the \((M_\infty, h_a)\) envelope, as shown in Figure 3.17. All of the flutter evaluation points are for the never-exceed dynamic pressure, because an increase in dynamic pressure always results in the aircraft being closer to flutter, even in transonic flight. Due to the transonic dip behavior, it is not clear which freestream Mach number is the worst-case for flutter. Therefore, several Mach numbers between the cruise Mach number and 10% above the end.
cruise Mach number are sampled. For each flutter evaluation point we also consider empty wing fuel tanks, half-full wing fuel tanks, and full wing fuel tanks, because these change the mass and center of gravity of the wing substantially.

Figure 3.18 shows the aeroelastic eigenvalues as function of wing aspect ratio $A$ and cruise Mach number $M_{\infty}$. Note that $\chi_{fl} > 0$ indicates flutter. These results consider a freestream Mach number 10% higher than $M_{CR}$ and full wing tanks as the flutter evaluation point. Note that the trends differ between the different flutter evaluation points shown in Figure 3.17, but the one shown in Figure 3.18 is one of worst cases. We see that the D8.0 and D8.2 have wildly different values for $\chi_{fl}$, while the trends are also dissimilar. For the D8.2 the increase in aspect ratio leads to an increase in $\chi_{fl}$ (i.e., closer to instability), whereas for the D8.0 the aspect ratio at first decreases $\chi_{fl}$ but for higher aspect ratios this trend reverses. The trend for the D8.2 is therefore similar to the results for the clamped wing configuration in Figure 3.13, where a higher aspect ratio decreases the flutter speed because the wing is more flexible for higher aspect ratios. For the D8.0, the wing also becomes more flexible for larger aspect ratios, but the engine also moves further outboard. For lower aspect ratios, putting the engine further outward moves the wing more...
towards stability. However, for higher aspect ratios the effect of the larger flexibility in the wing again moves the wing away from stability. An increase in design cruise Mach number puts the D8.2 closer to instability. However, for the D8.0 for lower aspect ratios the cruise Mach number decreases \( \frac{\chi_{fl}}{f_{l}} \), whereas for aspect ratios around \( AR = 11 \) higher cruise Mach numbers increase \( \frac{\chi_{fl}}{f_{l}} \).

Figure 3.18: Influence of aspect ratio \( AR \) on maximum aeroelastic eigenvalue \( \chi_0 \) for two different aircraft; the D8.0 has a wing with a large sweep angle and wing-mounted engines, the D8.2 has a lower-sweep wing and fuselage-mounted engines. These results are for a flutter evaluation at \( M_{\infty} = 1.10 M_{CR} \) with full wing fuel tanks.

Figure 3.19 shows the aeroelastic eigenvalues of weight-converged aircraft designs for different sweep angles of the wing \( \Lambda \) and different design cruise Mach numbers \( M_{CR} \). These results consider a freestream Mach number 10% higher than \( M_{CR} \) and empty wing tanks as the flutter evaluation point. The results for the D8.2 show that an increase in sweep angle moves the design away from instability, as does a decrease in cruise Mach number. For the D8.0, however, these trends are mostly reversed, showing the large influence of the inertia and mass of the engine on wing flutter. As explained in Section 3.1.2, with an increase in sweep angle, \( \Delta x_{eng} \) decreases and \( \Delta y_{eng} \) increases. The additional stability from changing the engine position therefore outweighs the push towards instability by the increase in sweep angle.
3.5 INFLUENCE OF FLUTTER ON AIRCRAFT DESIGNS

This section discusses the influence of transonic flutter on novel aircraft designs. We will focus here on several variants of the D8.0. All results in this section are generated using TAOPT, using the flutter evaluation points as described in Figure 3.17.

First, we show the influence of transonic flutter on the planform design of an advanced-technology version of the D8.0 (Figure 3.20). For this design, better engine technology, laminar-bottom wings, and carbon-
fiber structures are used, allowing for higher aspect ratio wings leading to a higher susceptibility to flutter. The flutter constraints essentially serve as a span constraint, as we have already seen that higher aspect ratios lead to lower flutter speeds in Figure 3.13. Due to the lower aspect ratio, the flutter-constrained design has worse aerodynamic performance, and therefore a 3.3% higher fuel burn. The lower span does result in a lower maximum take-off weight, but that does not offset the higher fuel burn due to poorer aerodynamic performance.

Second, we investigate specifically the influence of newer material technology on the performance of the D8.0 and describe the influence of transonic flutter constraints on this design (Figures 3.21 and 3.22). The specific allowable stress is used as a surrogate for new material technology, which is quantified here by a factor multiplying the baseline value, corresponding to aluminum in this case. The varying planform designs and influence on fuel burn and maximum take-off weight are shown in Figure 3.22 for specific allowable stress up to 50% higher values than aluminum.

As expected, an increase in specific allowable stress results in a lower fuel burn and maximum take-off weight, which was already observed by Drela\textsuperscript{43} for Boeing 737-class aircraft. The inclusion of the flutter constraints limits the efficiency gains seen by an increase in specific allowable stress; without a flutter constraint, a 50% increase in specific allowable stress decreases the fuel burn by as much as 12%, whereas the

\[ M_{CR} = 0.78 \]

\[ M_{CR} = 0.82 \]

Figure 3.21: Fuel burn increase as a result of transonic flutter constraints for different design cruise Mach numbers.
fuel burn decreases by “only” 10% when a flutter constraint is included. The fuel efficiency gains from better material properties are therefore limited by flutter constraints, as the aspect ratio is limited to mitigate flutter. The lower aspect ratio results in higher induced drag and therefore lower lift-to-drag ratios.

As already observed in Figure 3.20, the lower aspect ratio does result in a lower maximum take-off weight, but the fuel burn is still higher than for the design without flutter constraints. The trends in Figure 3.22 are similar to those described by Drela\textsuperscript{145} for the Boeing 737-class aircraft with and without a span constraint.

\textbf{Figure 3.22:} Variations of optimum fuel burn and maximum take-off weight with specific allowable stress.
Different design cruise Mach numbers change the influence of a flutter constraint (Figure 3.21). For instance, for $M_{CR} = 0.82$ the flutter constraint has less influence around 25% increase in $\sigma/\rho$ than for $M_{CR} = 0.80$ and $M_{CR} = 0.78$, but around 50% increase in $\sigma/\rho$ the influence of the flutter constraint has the largest influence for $M_{CR} = 0.82$. This again demonstrates the non-intuitive and subtle influence of transonic flutter constraints on conceptual aircraft design of next-generation transport aircraft.
DESIGN OF LATTICE STRUCTURES FOR ADDITIVE MANUFACTURING

Additive manufacturing allows for unprecedented complexity in part design, which enables the manufacturing of lattice structures. These structures are known for a high stiffness-to-weight ratio, while the large design freedom in the location of the nodes of the lattice and the size of its struts also allows for tailoring its stiffness in detail. The focus of this chapter is on developing an inexpensive approach to designing such lattice structures, while this approach will be used in Chapter 5 to tailor these structures to mitigate flutter. The lattice topology is designed using adaptive meshing techniques, which also allow for the incorporation of manufacturing constraints directly in the topology (Section 4.1). Once the lattice topology is designed, the cross-sectional areas of the lattice struts are optimized to achieve the minimum-weight lattice structure, subject to stress, compatibility, and buckling constraints (Section 4.2). This approach is demonstrated for a generic bracket design as well as an aircraft bracket, where an 80% weight reduction over the solid part is achieved using this methodology (Section 4.3).

4.1 DESIGN OF LATTICE TOPOLOGY

To achieve lightweight lattice structures, the lattice topology—i.e., the location of the lattice nodes and the connectivity between those nodes—needs to be aligned with the load direction and needs to be fine in areas of high stress, while it can be coarser in locations of lower stress. Moreover, the lattice topology should be defined such that it is actually manufacturable.

In this work, a Riemannian metric field (Section 4.1.1) is used to define the lattice topology, where the metric is informed by the stress tensor throughout the available design space of the problem, which is found by solving a linear elasticity problem on a fully solid design domain (Section 4.1.2). The metric field is based on this stress tensor, subject to corrections for direction and positive definiteness (Section 4.1.3). Moreover, information about manufacturing constraints and the build direction can be included in the Riemannian metric (Section 4.1.4). The complete approach is illustrated in Figure 4.1.
4.1.1 RIEMANNIAN METRIC FIELDS

A lattice structure can be described from a discrete or continuous viewpoint. In the discrete viewpoint, the lattice consists of nodes and struts, each with a certain length, thickness, and direction. In the continuous viewpoint, the properties of the lattice such as the local size—i.e., the volume that each element of the lattice encompasses—and directionality of the lattice can be described as a continuous field throughout the structure, as shown in Figure 4.2. The latter is a more elegant way of looking at the problem of designing a load-dependent lattice and also relaxes the intractability of solving a discrete optimization problem for the optimal topology. This continuous approach is demonstrated in the following.

The goal here is to design a load-dependent lattice topology—i.e., the location of the nodes and direction of the struts of the lattice are tailored to the specific loading condition(s) to which the part is subjected. To design such a load-dependent lattice, we leverage mesh adaptation
techniques from the CFD community. In CFD applications, meshes need to be locally refined in areas with large gradients, such as boundary layers and shock waves. Furthermore, these are areas where the gradient in one direction is large, while being small in a perpendicular direction, requiring the mesh to exhibit a certain anisotropy. One way of prescribing anisotropy is through the notion of Riemannian metric spaces, which were first used in Ref. 144 and 145 for mesh adaptation purposes in fluid flow problems. Such a metric is a tensor field that at any point in the domain describes mesh size and anisotropy. Mesh adaptation using a metric computed from solution error is demonstrated by, amongst others, Loseille and Alauzet,\textsuperscript{146,147} and Yano and Darmofal.\textsuperscript{148,149} Metric-based anisotropic mesh adaptation has also been used to increase the efficiency of SIMP-based topology optimization methods.\textsuperscript{150}

Most of these adaptive methods are developed for meshes with simplicial elements, which are also appropriate for structural applications. Triangles—or tetrahedra in three-dimensional structures—have good static structural properties and are therefore appropriate core elements of a lattice structure. Using such a metric to describe the triangulation is a natural way to include information of load paths into the lattice, and also to include manufacturing constraints, as we show in Section 4.1.4.

We now present the formal mathematical definition of a Riemannian metric field, closely following the notation in Ref. 146 and 147. A Riemannian metric field \( \{ M(x) \}_{x \in \Omega} \)—with \( x \) the physical coordinate—is a smoothly varying field of Symmetric Positive Definite (spd) matrices on the \( d \)-dimensional domain \( \Omega \subset \mathbb{R}^d \). The edge length of a segment \( \mathbf{a} \mathbf{b} \) from \( \mathbf{a} \in \Omega \) to \( \mathbf{b} \in \Omega \) under this Riemannian metric is given by

\[
\ell_M(\mathbf{a} \mathbf{b}) = \int_0^1 \sqrt{\mathbf{a} \mathbf{b}^T M(\mathbf{a} + s\mathbf{b}) \mathbf{a} \mathbf{b}} \, ds.
\]

The metric-conforming triangulation \( T \) is then such that all edges are close to unit length under the Riemannian metric field, \( \{ M(x) \}_{x \in \Omega} \).

\textbf{Figure 4.3:} Geometric interpretation of a metric. Note that the triangulation corresponding to the metric is not unique.
typically satisfying
\[
\frac{1}{\sqrt{2}} \leq \ell_M(e) \leq \sqrt{2} \quad \forall \ e \in \text{Edges}(T_h).
\]

A geometric interpretation of a metric and its corresponding triangulation are illustrated in Figure 4.3. As a further example of a metric field, consider the metric field in Figure 4.4, which was used to generate the lattice in Figure 4.2.

4.1.2 SOLID MECHANICS SOLVER

To find the continuous stress tensor, we solve the linear elasticity equations over the whole domain. The full linear elasticity equations are\(^{151}\)

\[
\nabla \cdot \sigma + f = 0 \quad (4.1a)
\]

\[
\varepsilon = \frac{1}{2} \left[ \nabla u + (\nabla u)^\top \right] \quad (4.1b)
\]

\[
\sigma = C : \varepsilon, \quad (4.1c)
\]

where \(\sigma\) is the Cauchy stress tensor, \(\varepsilon\) is the infinitesimal strain tensor, \(u\) is the displacement vector, \(C\) is the fourth-order stiffness tensor, and \(f\) is the body force per unit volume. Note that the : operator indicates a double contraction, such that Eq. (4.1c) can be written as \(\sigma_{ijkl} = C_{ijkl}\varepsilon_{kl}\) in index notation. \(C\) is defined as

\[
C_{ijkl} = \lambda_1 \delta_{ij} \delta_{kl} + \mu_s \left( \delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} \right),
\]

where \(\lambda_1\) is Lamé’s first parameter, and \(\mu_s\) is the shear modulus (or rigidity), both of which are elastic moduli, defined as

\[
\lambda_1 = \frac{E\nu}{(1 + \nu)(1 - 2\nu)} \quad \mu_s = \frac{E}{2(1 + \nu)},
\]

where \(E\) is the Young’s modulus, and \(\nu\) is Poisson’s ratio.

The linear elasticity equations are solved using a Finite Element Method (FEM), specifically a Hybridizable Discontinuous Galerkin (HDG) discretization, following the approach in Ref. 152. A detailed explanation of the solver used throughout this work is provided in Appendix B.

4.1.3 LOAD-DEPENDENT LATTICE BASED ON METRIC FIELD

To tailor the lattice topology to the load direction and areas of high stress, we base the lattice topology on the stress tensor throughout the solid domain. The stress tensor \(\sigma\) indicates which areas of the domain are highly loaded. Intuitively, it is desirable to have a finer lattice near highly loaded areas, and to have a coarser lattice near areas with low

\[\text{Figure 4.4: A metric is a field of SPD tensors, illustrated using ellipses here. This metric field is used to generate the lattice in Figure 4.2.}\]

\[151.\text{Shames et al., Introduction to Solid Mechanics, 2000}\]

\[152.\text{Soon et al., A Hybridizable Discontinuous Galerkin Method for Linear Elasticity, 2009}\]
stress. Furthermore, the lattice should align itself with the stress direction. Therefore, we adapt the lattice to the stress tensor using metric-based mesh adaptation techniques.\textsuperscript{144,148}

Our proposed method for computing the metric from the stress tensor applies corrections to ensure positive definiteness of the metric and to ensure that the direction of the metric aligns with the stress direction. The stress tensor is not necessarily positive definite because it distinguishes between compression and tension (Figure 4.5), whereas a metric is required to be positive definite. For adaptation purposes, however, it is not important to distinguish between compressive and tensile stresses since it is the magnitude of the stresses that drives the need for local refinement.\textsuperscript{*} Therefore, to guarantee the metric is positive definite, we take the magnitude of the principal stresses to be positive, such that this metric is positive definite—i.e. it has a positive determinant. In practice, this is implemented by taking the eigenvalue decomposition of the stress tensor and taking the absolute value of the eigenvalues to compute the metric.

Secondly, to align the lattice with the stress direction, the eigenvalues of the metric are altered. The magnitude of the local determinant of the metric is inversely proportional to the local size of the mesh. Therefore, when the determinant of the stress tensor is large in a particular part of the domain, the element sizes of the lattice are small in that part as well. To ensure the metric tensor is positive definite, we take the magnitude of the principal stresses, the bottom row illustrates what the lattice would look like if the eigenvalues of the final metric were taken to be the absolute principal stresses, the bottom row illustrates the desired lattice direction obtained by using scaled eigenvalues based on the principal stresses. $\mathcal{M}'$ is the metric $\mathcal{M}$ expressed in the coordinate system of the principal stresses.

\textbf{Figure 4.5:} The stress tensor is not necessarily positive definite, but an updated metric for adaptation purposes can be made positive definite by taking the absolute value of the principle stresses.

\textbf{Figure 4.6:} Using scaled eigenvalues for the final metric yields a lattice with correct orientation. The top row illustrates what the lattice would look like if the eigenvalues of the final metric were taken to be the absolute principal stresses, the bottom row illustrates the desired lattice direction obtained by using scaled eigenvalues based on the principal stresses. $\mathcal{M}'$ is the metric $\mathcal{M}$ expressed in the coordinate system of the principal stresses.

\textsuperscript{*}In designing the topology of the lattice, buckling is ignored.
well, which is desired. If the stress is high in one particular direction, the local lattice should be elongated along that direction, and thus the metric should be smaller in that direction. Therefore, the eigenvalues of the final metric are taken to be the inverse of the eigenvalues of the metric based on the stress tensor—while ensuring that the area under the metric stays constant. This is illustrated for two-dimensional problems in Figure 4.6.

For three-dimensional problems, we take the inverse of the principal stresses while scaling the metric such that the determinant of the metric stays the same. For three-dimensional problems, the metric is therefore computed as

\[
m_1 = \sqrt{S^2} \frac{1}{\sigma_1}, \quad m_2 = \sqrt{S^2} \frac{1}{\sigma_2}, \quad m_3 = \sqrt{S^2} \frac{1}{\sigma_3},
\]

where \( S \equiv \sigma_1 \sigma_2 \sigma_3 \) and \( m_1, m_2, \) and \( m_3 \) are the eigenvalues of the final metric \( M. \) The metric \( M \) is used to define the lattice topology.

4.1.4 MANUFACTURING CONSTRAINTS

Although additive manufacturing allows for large design freedom, few AM processes can manufacture any type of lattice structure. Most processes are for instance limited in terms of the overhang tolerated for parts. Typically, extrusion-based manufacturing processes have more stringent overhang constraints than powder-based manufacturing techniques, but almost all additive manufacturing processes have some form of overhang constraints.

The two most significant manufacturing constraints are the minimum geometric feature due to limited printer resolution, and the overhang angle constraint. The latter constraint is imposed because a new layer needs to have support from the structure below, as shown in Figure 4.7. Even in powder-based manufacturing methods this constraint remains due to heat transfer and warping issues. We also include bridge constraints, which limit the maximum horizontal “bridge” that can be printed.

\[ \beta \]

Figure 4.7: In this work, overhang angle constraints and bridge length constraints are considered.
There are several different ways to include the manufacturing constraints. The minimum geometric feature size limit is straightforward to include by setting the minimum mesh size through the metric. For the overhang angle constraint, we optimize the local metric such that it is as close as possible to the original metric in the Frobenius norm, while adhering to the manufacturing constraints. In order to solve that optimization problem, the orientation and shape of the metric needs to be described geometrically. Here, we choose *quaternions* \(^53,54\) to describe the axis orientation of the metric, and the length of the major axes \((h_1, h_2, h_3)\) to describe the shape of the ellipsoid (Figure 4.8). Quaternions are chosen over Euler or Tait-Bryan angles, because these have a singularity which can limit the convergence of the optimization algorithm. The optimization problem is described mathematically as

\[
\begin{align*}
\min_{\mathbf{q}, h_1, h_2, h_3} & \quad \| \mathcal{M} - \mathcal{M}_{\text{org}} \|_F \\
\text{subject to} & \quad \frac{n_1 + n_2}{|n_1 + n_2|} \cdot \mathbf{n}_{\text{printer}} \geq \cos \beta_{\text{max}} \\
& \quad \frac{n_1 + n_3}{|n_1 + n_3|} \cdot \mathbf{n}_{\text{printer}} \geq \cos \beta_{\text{max}} \\
& \quad \| \mathbf{q} \| = 1,
\end{align*}
\]

where \(\mathbf{q} = [q_1, q_2, q_3, q_4]^T\) are the quaternions, and \(\mathcal{M} = \mathbf{R} \Lambda \mathbf{R}^T\) with \(\mathbf{R}\) obtained from \(\mathbf{q}\) as

\[
\mathbf{R} = \begin{bmatrix}
1 - 2q_3^2 - 2q_4^2 & 2q_2q_3 - 2q_1q_4 & 2q_2q_4 + 2q_1q_3 \\
2q_2q_3 + 2q_1q_4 & 1 - 2q_2^2 - 2q_4^2 & 2q_3q_4 - 2q_1q_2 \\
2q_2q_4 - 2q_1q_3 & 2q_3q_4 + 2q_1q_2 & 1 - 2q_2^2 - 2q_3^2
\end{bmatrix}
\]

and \(\Lambda\) the eigenvalue matrix obtained from \(h_1, h_2, h_3\) as

\[
\Lambda = \begin{bmatrix}
1/h_1^2 & 0 & 0 \\
0 & 1/h_2^2 & 0 \\
0 & 0 & 1/h_3^2
\end{bmatrix}.
\]
The normal vectors of the major axes of the ellipse—\( \mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3 \)—are the columns of \( \mathbf{R} \). \( \mathbf{n}_{\text{printer}} \) is the \( z \)-axis of the printer expressed in the coordinate system of the part. Note that this optimization problem is non-convex, due to the \( \| \mathbf{q} \| = 1 \) constraint. A general nonlinear optimization solver is therefore used to solve this problem, in this case an interior point optimizer, Ipopt.\textsuperscript{155} A geometric interpretation of this optimization problem is that the metric has to be oriented such that it points into the manufacturable cone (Figure 4.8), while deviating as little as possible from the initial metric.

This approach to include manufacturing constraints into the metric is purely geometric, without a direct relationship to the loads through the lattice. However, the impact on the load bearing capability of the lattice is limited, since we find at each point in the domain a metric that is closest to the original metric obtained from the stress tensor, which ensures the lattice is still as closely aligned with the stress direction as possible. An example of the result of the optimization in 2D is illustrated in Figure 4.9, where in this case the metric is mostly rotated. We also show the influence of printer angle on the direction of the lattice for a column in compression in Figure 4.10, which clearly shows that the lattice is oriented along the printer \( z \)-axis.

\[
\left| \beta \pm \tan^{-1} \left( \frac{h_2}{h_1} \right) \right| > \beta_{\text{max}} \quad \text{and} \quad \left| \beta' \pm \tan^{-1} \left( \frac{h'_2}{h'_1} \right) \right| < \beta_{\text{max}}
\]

\textbf{Figure 4.9:} 2D example of including manufacturing constraints by optimizing the local metric.

Once the metric is optimized to include manufacturing constraints, the lattice topology is generated using a metric-based mesh generator. Throughout this work, we use \texttt{femto}.\textsuperscript{156–158} To generate the initial mesh for which we compute the stress tensor, any mesh generator can be used, in this work typically commercial mesh generation software is used.

Even though the lattice topology is now load-dependent and aware of manufacturing constraints through the metric, not all struts may follow manufacturable directions. Several heuristic approaches can be taken to alter the lattice to ensure 100% manufacturability. One could find struts in the lattice that violate the manufacturing constraints, and then perform edge swaps on them which in many cases may make the new strut manufacturable, see Figure 4.11. Such edge swaps are commonly

\begin{align*}
\left| \beta \pm \tan^{-1} \left( \frac{h_2}{h_1} \right) \right| > \beta_{\text{max}} & \quad \text{and} \quad \left| \beta' \pm \tan^{-1} \left( \frac{h'_2}{h'_1} \right) \right| < \beta_{\text{max}} \\
\end{align*}

\textsuperscript{156. Loseille et al., Anisotropic Adaptive Simulations in Aerodynamics. 2010}
\textsuperscript{157. Loseille, Metric-Orthogonal Anisotropic Mesh Generation. 2014}
\textsuperscript{158. Loseille et al., Unique Cavity-Based Operator and Hierarchical Domain Partitioning for Fast Parallel Generation of Anisotropic Meshes. 2017}
performed in meshing algorithms. For a 3D structure, however, we only perform these edge swaps on boundaries, as in the interior swapping edges may result in struts colliding with one another.

For struts that are perpendicular to the printer’s $z$-axis, we need to ensure that they have enough support from below. In this work, we support the “horizontal” strut by finding a node under the strut and then adding struts between that node and the horizontal strut (Figure 4.11).

The remaining struts that are not manufacturable could simply be removed, as is demonstrated by Shidid et al., but that does not necessarily guarantee manufacturability—one could remove a strut that was critical to support another strut—nor is there any guarantee that the resulting lattice is still stable. For now, we take Shidid’s approach, but a more integral approach is a topic of future work.

### Figure 4.10: The lattice is aligned with the printer direction for this column under compression.


4.2 **Optimal Cross-Sectional Area of Lattice Struts**

Once the lattice topology is defined, we can apply truss optimization techniques to optimize the size of each strut in that lattice. If one is interested in the minimum-weight solution with the cross-sectional area of the struts as design variables, the problem can be cast into a linear
programming form. Take \( \mathbf{a} \) and \( \mathbf{l} \) to be the vectors containing the areas and lengths, respectively, of the lattice struts. Then, for a lattice with \( n \) nodes and \( m \) struts in physical dimension \( d \), we minimize the volume of the lattice \( \mathcal{V} \) by solving\(^{159}\)

\[
\begin{align*}
\min_{\mathbf{a}, \mathbf{l}} \quad & \mathcal{V} = \mathbf{l}^\top \mathbf{a} \\
\text{subject to} \quad & \mathbf{C} \mathbf{a} = \mathbf{F} \\
& -\sigma_C a_j \leq f_j \leq \sigma_T a_j \\
& a_j \geq 0 \\
& f \in \mathbb{R}^m
\end{align*}
\]

where \( \mathbf{f} \) is a vector containing the forces in the lattice struts, and \( \mathbf{F} \) are the nodal forces—the definitions of these variables is also illustrated in Figure 4.12. \( \sigma_C \) and \( \sigma_T \) are the compressive and tensile stress limits, respectively. \( \mathbf{C} \in \mathbb{R}^{n \times m} \) is the connectivity matrix between nodes and struts, which is directly determined by the topology of the lattice. Each column of \( \mathbf{C} \) is the projection of a strut of the lattice on the degrees of freedom of the nodes on the lattice that are connected to that strut. Therefore, the matrix entry corresponding to the \( i \)th node and \( j \)th strut for a three-dimensional lattice is expressed as,

\[
\begin{align*}
C_{3(i-1)+1, j} &= \mathbf{n}_j \cdot \hat{x} \\
C_{3(i-1)+2, j} &= \mathbf{n}_j \cdot \hat{y} \\
C_{3(i-1)+3, j} &= \mathbf{n}_j \cdot \hat{z},
\end{align*}
\]

where \( \hat{x} \), \( \hat{y} \), and \( \hat{z} \) are the unit vectors in the \( x \), \( y \), and \( z \) directions, respectively. As an example, consider Figure 4.13 where strut \( k \) is orientated in the \( xy \)-plane and make a \( 60^\circ \) angle with the \( xz \)-plane.

This plastic formulation in Eq. (4.3) can be reformatted as a standard linear program and can therefore be solved very quickly. This formulation, however, can only be used for single load cases with no minimum area constraints, because the solution has to be a statically determinate structure, seeing as no stress-strain compatibility conditions are taken
into account. However, we require constraints to be placed on the cross-sectional areas of the struts. Therefore, a formulation which satisfies the stress-strain relations has to be used,

\[
\begin{align*}
\min_{\mathbf{a}, \mathbf{f}, \sigma} & \quad \mathcal{V} = \mathbf{l}^\top \mathbf{a} \\
\text{subject to} & \quad \mathbf{Cf} = \mathbf{F} \quad \text{(force balance)} \\
& \quad \mathbf{B}\sigma + \mathbf{C}^\top \mathbf{u} = \mathbf{0} \quad \text{(stress-strain compatibility)} \\
& \quad \bar{f}_j = \sigma_j a_j \quad \text{(stress definition)} \\
& \quad -\sigma_C \leq \sigma_j \leq \sigma_T \quad \text{(stress limits)} \\
& \quad a_{\min,j} \leq a_j \leq a_{\max,j} \\
& \quad \mathbf{u} \in \mathbb{R}^n, \quad \mathbf{f} \in \mathbb{R}^m
\end{align*}
\]

where \( \mathbf{B} \in \mathbb{R}^{m \times m} \) is a diagonal matrix where the \( j \)th diagonal entry corresponds to the deformation per unit force of the \( j \)th strut of the lattice,

\[
B_{jj} = \frac{l_j}{Ea_j},
\]

where \( l_j \) and \( a_j \) are the length and cross-sectional area of the \( j \)th strut, respectively, and \( E \) is the Young's modulus. This elastic formulation has been known to suffer from problems with vanishing constraints as \( a_j \to 0 \), resulting in \( \sigma_j \) becoming independent of \( f_j \). Fortunately, we specifically want to prescribe a minimum area for each strut to ensure support for each strut in the lattice, and therefore no vanishing constraints occur in this problem.

As written, Eq. (4.4) is not convex, due to the \( f_i = \sigma_i a_i \) constraint. If the objective were maximum stiffness, this problem could be written as a convex optimization problem. However, we are specifically looking for the minimum-weight solution, which results in a non-convex optimization problem. The problem is therefore solved as a generic nonlinear optimization problem using Ipopt.

Finally, Eq. (4.4) can be extended to include buckling constraints. The critical Euler buckling load \( f_{\text{crit},j} \) of the \( j \)th strut can be expressed as

\[
f_{\text{crit},j} = \frac{\pi E a_j^2}{4(k_e l_j)^2},
\]

where \( k_e \) is the column effective length factor. This constraint is also nonlinear in \( a_j \), which further adds to the nonlinearity of Eq. (4.4). This
constraint is added to the optimization statement in Eq. (4.4) as

\[ f_j \geq -f_{\text{crit}, j}. \]

4.3 EXAMPLE DESIGNS

The design methodology in Sections 4.1 and 4.2 is demonstrated here for several test cases. A simple bracket is designed in Section 4.3.1, which is used as a test case to compare the performance of an isotropic lattice to that of an anisotropic lattice and to describe the influence of the strut cross-sectional area optimization on the mass of a part. We further demonstrate the design methodology on a test case from General Electric in Section 4.3.2, which allows us to compare our methodology to methods from the literature.

4.3.1 GENERIC BRACKET

As a first example to demonstrate our design methodology, we design a generic bracket, for which the load case is shown in Figure 4.14. In the following, we compare the performance of a tailored lattice topology to that of an isotropic lattice topology and also characterize the influence of printer angle on the manufacturability of the part. We consider the design domain and load case shown in Figure 4.14. The part is to be manufactured using standard Acrylonitrile Butadiene Styrene (ABS) plastic. The Von Mises stress throughout the part is shown in Figure 4.15 for that load case. As expected the Von Mises stress is highest near the “neck” of the structure, both because that part has the lowest cross-sectional area and because of the discontinuity in the geometry of the bracket, which lead to stress concentrations.

From the stress tensor computed for the load case in Figure 4.14, we compute a metric field. The lattice topology is then generated using a metric-based mesh generator, in this case feflo.a.\textsuperscript{156} The cross-

\[ \begin{align*}
\text{Clamped} \\
0.25 \text{ MPa} \\
2.5 \text{ MPa}
\end{align*} \]

Figure 4.14: Load case for bracket design example. The largest dimension is in z-direction (150 mm).
sectional area of the struts in that resulting lattice are optimized to yield the minimum-weight structure, subject to stress and buckling constraints. In the optimization we limit the diameter of the struts to be between 3 mm and 5 mm. This lower limit is imposed since the bracket is to be manufactured using Fused Filament Fabrication (FFF), which requires a minimum strut diameter of around 3 mm. Under these constraints, the resulting optimized part weighs only 0.140 kg for a lattice with 401 struts. A render of this part is shown in Figure 4.16.

To investigate the importance of the lattice topology, we compare the minimum weight of the anisotropic lattice—which is aligned with stress directions—to the minimum weight of an isotropic lattice which is not tailored to the stress directions. For fair comparison, the anisotropic and isotropic lattices have the same number of struts. For the isotropic lattice then, we also optimize the cross-sectional area of each strut. For the optimization problem of the isotropic lattice, the diameter constraints of the struts have to be relaxed slightly compared to the anisotropic lattice to ensure a feasible design. The diameter of the struts is therefore constrained to be between 3 mm and 6 mm. The optimized isotropic lattice weighs 0.209 kg—a 37% increase compared to the anisotropic lattice. These designs are also compared in Figure 4.16. We see in Figure 4.16 that
the isotropic lattice has a few thick struts while most struts are assigned the minimum strut diameter. This could be expected, because there are only few struts that are aligned with the stress direction and therefore there are only a few struts that take up most of the load. The anisotropic lattice is therefore also expected to exhibit better load-bearing capability in case one or more struts fail. Also note that the manufacturability of the isotropic lattice is only 59%, whereas for the anisotropic lattice it is 100%—this again is expected, because the anisotropic lattice is actually designed with manufacturing constraints in mind. To demonstrate the manufacturability of the anisotropic lattice, the bracket is manufactured using an FFF desktop printer—the printed part is shown in Figure 4.17.

As a final comparison, we investigate the influence of optimizing the cross-sectional area of each strut in the lattice on the performance of a part. We compare the previous designs to lattices with a uniform cross-sectional area for all struts for both the anisotropic and isotropic lattice; we call these designs homogeneous lattices. The minimum-weight uniform cross-sectional area for the homogeneous lattices is found by a bisection method, which finds the minimum cross-sectional area for which one or more struts are at the stress limit. We compare these designs in Table 4.1, where we see that the homogeneous lattices are at least 51% heavier than their heterogeneous counterparts.

<table>
<thead>
<tr>
<th></th>
<th>Mass, kg</th>
<th>Manufacturability, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anisotropic</td>
<td>0.152</td>
<td>100</td>
</tr>
<tr>
<td>Anisotropic (homogeneous)</td>
<td>0.230</td>
<td>100</td>
</tr>
<tr>
<td>Isotropic</td>
<td>0.232</td>
<td>59.2</td>
</tr>
<tr>
<td>Isotropic (homogeneous)</td>
<td>0.367</td>
<td>59.2</td>
</tr>
</tbody>
</table>

Table 4.1: Manufacturability and performance for anisotropic and isotropic lattices.

Finally, we also compare the influence of the design for manufacturability on the actual manufacturability of the resulting lattice. For this comparison, we design a lattice for various manufacturing directions by varying $\beta$ for the design and then check the manufacturability of the
part under different \( \beta \) manufacturing angles. Manufacturability here is defined as the ratio between the number of struts that follow manufacturable directions to the total number of struts in the part. The results in Table 4.2 show that whenever the part is to be manufactured in the same direction as it is designed, the manufacturability is 100\%, validating the approach. However, when we vary the angle at which the part is manufactured, the manufacturability quickly drops to levels as low as 53\%. Incorporating manufacturing constraints for the lattice into the design is therefore crucial.

<table>
<thead>
<tr>
<th>( \beta ) (manufacturing)</th>
<th>in %</th>
<th>-45\°</th>
<th>-25\°</th>
<th>0\°</th>
<th>25\°</th>
<th>45\°</th>
</tr>
</thead>
<tbody>
<tr>
<td>-45\°</td>
<td>100</td>
<td>97.1</td>
<td>99.3</td>
<td>87.7</td>
<td>60.0</td>
<td></td>
</tr>
<tr>
<td>-25\°</td>
<td>77.5</td>
<td>100</td>
<td>98.6</td>
<td>94.2</td>
<td>67.8</td>
<td></td>
</tr>
<tr>
<td>0\°</td>
<td>78.5</td>
<td>97.2</td>
<td>100</td>
<td>94.9</td>
<td>74.1</td>
<td></td>
</tr>
<tr>
<td>25\°</td>
<td>72.9</td>
<td>94.7</td>
<td>98.7</td>
<td>100</td>
<td>78.3</td>
<td></td>
</tr>
<tr>
<td>45\°</td>
<td>53.4</td>
<td>86.5</td>
<td>98.8</td>
<td>96.6</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: Manufacturability (\%) for manufacturing and designing a bracket with different manufacturing orientations.

### 4.3.2 AIRCRAFT ENGINE BRACKET

In 2013, General Electric (GE) launched a competition to redesign an aircraft engine bracket to take advantage of additive manufacturing capabilities. Although the deadline for the competition has long since passed,
this design problem still serves as a useful test case for this work. Tang et al.\textsuperscript{164} used this bracket case study to demonstrate their lattice optimization algorithm, the result of which we compare our methodology against. Note that in this study only one load case is used—a 42.5 \textit{kN} bearing load, see Figure 4.18. This bracket is to be manufactured using Titanium Ti-6Al-4V, which has a tensile yield stress of 880 MPa, a compressive yield stress of 970 MPa, and a Young’s modulus of 113.8 GPa.

The Von Mises stress through this bracket is shown in Figure 4.19, where it can be seen that this bracket is fairly lightly loaded, with the maximum stress in the hooks around 350 MPa. The rest of the structure is even more lightly loaded, and therefore provides a good opportunity to minimize the weight using a lattice structure.

![Von Mises stress through domain of bracket for load case in Figure 4.18.](image)

The results in Figure 4.19 are used to generate the lattice topology, after which the cross-sectional area for each strut is optimized for minimum weight, subject to buckling and compatibility constraints. In this optimization, the radius of the struts is constrained to be between 0.3 mm and 1 mm, the same constraints as in Ref. 164.

The resulting lattice structure is shown in Figure 4.20, which has a mass of 0.408 kg—an 80\% reduction in mass compared to the solid part. The lattice structure from Tang et al.\textsuperscript{164} had a mass of 0.54 kg. The difference is explained by the hooks and bolt holes being solid in that work, as well as by their use of a fine isotropic lattice. It should also be noted that Tang et al. did not include buckling of the lattice struts in their

\textsuperscript{164} Tang et al., Bidirectional Evolutionary Structural Optimization (BESO) Based Design Method for Lattice Structure to be Fabricated by Additive Manufacturing. 2015
analysis, while that has a large influence on the mass—without including buckling, our analysis shows that the bracket would weigh 0.286 kg. Use of stress-aligned lattice structures thus yields a large mass reduction.

These examples have demonstrated the design methodology for additive manufacturing using lattice structures, which can result in large weight savings. Furthermore, the design freedom in the weight distribution and stiffness of these lattices make them ideal for aeroelastic tailoring. This design methodology is used in the next chapter to tailor internal wing lattice structures to mitigate (transonic) flutter.

Figure 4.20: Optimized lattice structure for aircraft bracket. This part weighs 0.408 kg.
AEROELASTIC TAILORING USING LATTICE STRUCTURES

This chapter combines the methods presented in Chapters 2 to 4 by designing a lattice structure for the internal structure of a wing to mitigate (transonic) flutter with minimal mass increase. To achieve this, the dynamics of the lattice structure have to be computed. This has to be done efficiently in order to include a flutter constraint in a design process, therefore a low-order model is constructed (Section 5.1). It is demonstrated that this low-order model can accurately capture the bending and torsion modes of the full-order model (Section 5.2). This design methodology is then demonstrated on a wing in transonic flow (Section 5.3).

5.1 DESIGN FOR FLUTTER MITIGATION

This section details the aeroelastic tailoring approach, which is summarized in Section 5.1.1. The structural model for the flutter model is described in detail in Section 5.1.2. Section 5.1.3 describes the aeroelastic tailoring of a lattice structure through optimizing the cross-sectional area of each of the lattice’s struts.

5.1.1 AEROELASTIC TAILORING APPROACH

The complete aeroelastic tailoring approach can be summarized as follows. This work uses a physics-based low-order model for transonic flutter. The first step, therefore, is to calibrate the aerodynamic coefficients of that model using 2D unsteady Euler simulations for the airfoil family used in the wing design. This approach is described in detail in Chapter 2.

Second, the load distribution over the wing is computed to obtain the baseline lift coefficient $c_{\ell_0}$ at each beam section—which is needed to select the correct aerodynamic coefficients for the flutter model—and to generate a pressure field over the wing that is subsequently used as input to the structural analysis of the wing internal structure. There are several ways of computing the flow over the wing, e.g., using a vortex-lattice method or a higher-fidelity Euler simulation. In the current work, we use an Euler CFD simulation computed with the SU2 tool suite.118

Part of the research presented in this chapter has been published as Opgoenoord M. M. J. and Willcox, K. E., "Aeroelastic Tailoring using Additively Manufactured Lattice Structures," 2018 AIAA Multidisciplinary Analysis and Optimization Conference, Atlanta, Georgia, June 2018.
The pressure distribution over the wing is used as the input for the structural FEM analysis of the internals of the wing, which allows for computing the stress tensor everywhere in the domain. From the stress tensor, then, a Riemannian metric field is computed, in which manufacturing constraints—such as overhang angle constraints or minimum feature size constraints—are incorporated. That metric is used to generate a lattice topology that is optimal and manufacturable using a metric-based mesh generator. This design methodology is described in detail in Chapter 4.

Finally, the size of each strut of this lattice is determined by solving an optimization problem that takes into account compatibility conditions, buckling, and flutter. To determine whether or not flutter occurs for such a lattice structure, we build up a low-order structural model for the lattice and combine that with the low-order aerodynamic model.

![Diagram of lattice models](image)

**Figure 5.1:** The full-order dynamic lattice model is reduced to a low-order lattice model to be used in the lattice optimization.

### 5.1.2 Low-Order Structural Model for Lattices

To compute the dynamic characteristics of lattice structures for the internal structure of a wing efficiently, we develop a low-order model for the lattice structure dynamics. This dynamic low-order model is defined in terms of the vertical and angular displacement of the wing sections (Figure 5.1), which are needed for the aerodynamic model.

For a lattice structure with $n$ nodes and $m$ struts, the dynamics are described by

$$M\ddot{u}(t) + Ku(t) = F(t),$$

(5.1)

where $M \in \mathbb{R}^{3n \times 3n}$ is the mass matrix, $K \in \mathbb{R}^{3n \times 3n}$ is the stiffness matrix,
\( \mathbf{F}(t) \in \mathbb{R}^{3n} \) is a dynamic force vector, and \( \mathbf{u}(t) \in \mathbb{R}^{3n} \) is the displacement vector of each node in the lattice.

\( \mathbf{K} \) and \( \mathbf{M} \) are potentially quite large (sparse) matrices, resulting in a large aeroelastic system. As part of the lattice optimization, the eigenvalues of this aeroelastic system need to be computed repeatedly, which can become quite expensive. Therefore, we reduce the size of the structural system by using an equivalent beam model (Figure 5.1), which describes the vertical and torsional deflection of several wing sections, rather than the deflection of each lattice node. The dynamics of the beam model are described by

\[
\begin{bmatrix}
\mathbf{M} & \mathbf{S}_\gamma \\
\mathbf{S}_\gamma & \mathbf{I}_\gamma
\end{bmatrix}
\begin{bmatrix}
\ddot{\mathbf{h}} \\
\ddot{\theta}
\end{bmatrix} +
\begin{bmatrix}
\mathbf{K}_{hh} & \mathbf{K}_{h\theta} \\
\mathbf{K}_{\theta h} & \mathbf{K}_{\theta\theta}
\end{bmatrix}
\begin{bmatrix}
\mathbf{h} \\
\theta
\end{bmatrix} =
\begin{bmatrix}
-\mathbf{L} \\
\mathbf{M}_{c/2}
\end{bmatrix},
\]  

(5.2)

where \( \mathbf{h} \) are the vertical deflections of the wing sections and \( \theta \) are the torsional deflections of the wing sections. \( \mathbf{L} \) is the lift acting at each wing section, and \( \mathbf{M}_{c/2} \) is the moment around the mid-chord at each wing section. The matrices \( \mathbf{M}, \mathbf{S}_\gamma \), and \( \mathbf{I}_\gamma \) are computed from the full-order mass matrix \( \mathbf{M} \), and \( \mathbf{K}_{hh}, \mathbf{K}_{h\theta}, \mathbf{K}_{\theta h}, \) and \( \mathbf{K}_{\theta\theta} \) are computed from the full-order stiffness matrix \( \mathbf{K} \).

The full-order mass matrix \( \mathbf{M} \) is obtained by lumping half the weight of each strut connecting to a node into the mass of that node, while having massless connectors (Figure 5.2).

We define the low-order mass matrix \( \mathbf{M} \) as a diagonal matrix, where the \( i \)th diagonal entry corresponds to the mass per span of the \( i \)th beam section. Similarly, \( \mathbf{S}_\gamma \) is a diagonal matrix where the diagonal represents the mass unbalance per span at each beam section. Finally, \( \mathbf{I}_\gamma \) is a

![Figure 5.2: Mass matrix is generated by lumping the mass from the edges into the nodes.](image)
diagonal matrix of the mass moment of inertia per span at each beam section.

The low-order mass matrix $\mathcal{M}$ is mapped from the full-order mass matrix $M$ as
\[
\text{diag}(\mathcal{M}) = T_{M\to\mathcal{M}} \text{diag}(M),
\]
where $T_{M\to\mathcal{M}}$ is defined as
\[
T_{M_k\to\mathcal{M}_j} = \frac{\varphi_j(\bar{y}_k)}{\int_0^1 \varphi_j(\bar{y})d\bar{y}},
\]
with $\varphi_j$ the basis function of the $j$th node and $\bar{y}$ the coordinate along the elastic axis of the wing. Throughout this work, linear nodal basis functions are used for $\varphi_j$. All quantities necessary for the mapping between the beam model and the full-order model are defined in Figure 5.3.

The mass unbalance matrix $\mathcal{S}$ is found from a similar mapping from the mass matrix $M$,
\[
\text{diag}(\mathcal{S}_{\bar{y}}) = T_{M\to\mathcal{S}} \text{diag}(M)
\]
with
\[
T_{M_k\to\mathcal{S}_j} = \frac{\varphi_j(\bar{y}_k)}{\int_0^1 \varphi_j(\bar{y})d\bar{y}} (\bar{x}_k - \bar{x}_j).
\]
Finally, the mass matrix \( M \) maps to the mass moment of inertia matrix \( I_\gamma \) as

\[
\text{diag}(I_\gamma) = T_{M\to I} \text{diag}(M)
\]

with

\[
T_{M_k\to I_j} = \frac{\varphi_j(\bar{y}_k)}{\int_\gamma \varphi_j(\bar{y}) \, d\bar{y}} \left[ (\bar{x}_k - \bar{x}_j)^2 + (\bar{z}_k - \bar{z}_j)^2 \right].
\]

To fully define the beam model (5.2), we also need to compute the low-order stiffness matrices. The stiffness matrix \( K \) is obtained from the connectivity of the lattice \( (C) \) and stress-strain relations \( (B) \) as

\[
K = CB^{-1}C^T.
\]

For this beam model, we only consider bending around the \( \bar{x} \) axis and torsion around the \( \bar{y} \) axis. Therefore, we only have to consider displacements in \( z \)-direction. To find the stiffness matrices for the beam model, we consider the system

\[
\begin{bmatrix}
K & 0 & 0 \\
T_{u\to h} & -I & 0 \\
T_{u\to \theta} & 0 & -I
\end{bmatrix}
\begin{bmatrix}
u \\ h \\ \theta
\end{bmatrix} =
\begin{bmatrix}
T_{L\to F} \\
0 \\
0
\end{bmatrix} L +
\begin{bmatrix}
T_{M\to F} \\
0 \\
0
\end{bmatrix} M_{c/2},
\]

(5.3)

where \( T_{u\to h}, T_{u\to \theta}, T_{L\to F}, \) and \( T_{M\to F} \) are mappings between the low-order and full-order model, which will be defined later in this section. Through Schur complements of Eq. (5.3), this system can be rewritten as

\[
\begin{bmatrix}
T_{u\to h}K^{-1}T_{L\to F} & T_{u\to h}K^{-1}T_{M\to F} \\
T_{u\to \theta}K^{-1}T_{L\to F} & T_{u\to \theta}K^{-1}T_{M\to F}
\end{bmatrix}
\begin{bmatrix}
L \\ M_{c/2}
\end{bmatrix} =
\begin{bmatrix}
h \\ \theta
\end{bmatrix},
\]

(5.4)

which combines Eq. (5.1) with mappings between the full displacement vector \( u \) and the vertical displacement (positive downward) \( h \) and the angular displacement \( \theta \), as well as mappings between the full force vector \( F \) and the lift vector \( L \) and the moment around the mid-chord \( M_{c/2} \). To find the low-order stiffness matrices, the block inverse of Eq. (5.4) is computed, yielding the system

\[
\begin{bmatrix}
K_{hh} & K_{h\theta} \\
K_{\theta h} & K_{\theta\theta}
\end{bmatrix}
\begin{bmatrix}
h \\ \theta
\end{bmatrix} =
\begin{bmatrix}
-L \\ M_{c/2}
\end{bmatrix}
\]

(5.5)

To map the full displacement vector \( u \) to the deflection of the beam model \( h \), we consider the average \( z \)-displacement near the beam node,

\[
T_{u\to z\to h} = \frac{\mathcal{A}_k \varphi_j(\bar{y}_k)}{\sum \mathcal{A}_i \varphi_j(\bar{y}_i)},
\]

where \( \mathcal{A}_k \) is the projected area in \( z \)-direction of the \( k \)th lattice node (Figure 5.3). The full displacement vector \( u \) can also induce a pitch
deflection of the beam model; the mapping between \( u \) and \( \theta \) is defined to be
\[
T_{u,z,k} \rightarrow \theta_j = \frac{1}{2 \sum_i \ell_i \varphi_j (y_i)} \frac{\ell_k \varphi_j (y_k)}{\bar{x}_k - \bar{x}_j},
\]
where \( \ell_k \) is the length along the leading or trailing edge for the lattice node \( k \).

The lift per span \( L \) only acts on the surface of the lattice, and is mapped to the full force vector \( F \) by considering the average pressure around the \( j \)th beam node,
\[
T_{L, j} \rightarrow F_{k,z} = \frac{1}{c_j} \varphi_j (y_k) A_k.
\]

Finally, the moment around mid-chord per span \( M_{c/2} \) is mapped to the full force vector \( F \) as,
\[
T_{M, j} \rightarrow F_{k,z} = \ell_k \varphi_j (y_k) \frac{1}{\bar{x}_k - \bar{x}_j}.
\]

These mappings fully define the low-order structural model (5.2). That beam model has four states per beam section: the vertical (downward) deflection \( h_i \), the vertical (downward) velocity \( w_i \), the pitch angle \( \theta_i \) and the pitch rate \( \omega_i \). Combined with the aerodynamic states \((\Delta \Gamma, \Delta \kappa_x, \Delta \dot{\kappa}_x)\), the aeroelastic system therefore has seven states per beam section.

### 5.1.3 Lattice Optimization with Flutter Constraint

To find the optimal area of each strut in the lattice, we solve a nonlinear optimization problem, which includes limits on the stresses in each strut, buckling constraints, and constraints on flutter behavior. This optimization problem for a lattice with \( m \) struts and \( n \) nodes in the physical dimension \( d \) is written as
\[
\min_{u,a,f,\sigma} V = \Gamma^T a \tag{5.6}
\]
subject to \( \mathbf{C}f = \mathbf{F} \) (force balance),
\[
\mathbf{B}\sigma + \mathbf{C}^T \mathbf{u} = \mathbf{0} \quad \text{(stress-strain compatibility)}
\]
\[
f_j = \sigma_j a_j \quad \text{(stress definition)}
\]
\[
-\sigma_C \leq \sigma_j \leq \sigma_T \quad \text{(stress limits)}
\]
\[
f_j \geq -\frac{\pi E a_j^2}{4(k e_j)^2} \quad \text{(buckling)}
\]
\[
\chi_{fl} < \chi_{fl,\max} \quad \text{(flutter)}
\]
\[
a_{\min,j} \leq a_j \leq a_{\max,j}
\]
\[
\mathbf{u} \in \mathbb{R}^{n_d}, \quad \mathbf{f} \in \mathbb{R}^m,
\]
where $V$ is the volume of the lattice, $a_j$ is the cross-sectional area of the $j$th lattice strut, $f_j$ is the force in the $j$th lattice strut, $\sigma_j$ is the stress in the $j$th lattice strut, and $l_j$ is the length of the $j$th strut. $\sigma_T$ is the maximum allowable tensile stress and $\sigma_C$ is the maximum allowable compressive stress. $\chi_f$ is the maximum real part of the eigenvalues of the aeroelastic system, which is constrained to be lower than $\chi_{f,\text{max}}$ (typically $-0.005/s$) for aeroelastic stability, similar to the work of Ringertz. Finally, $k_{e}$ is the column effective length factor, which is taken to be 1.2 here.

The optimization problem in Eq. (5.6) is non-convex, due to the constraint $f_j = \sigma_j a_j$ and the flutter constraint being non-convex. The problem is therefore solved using a generic NLP solver, in this case Ipopt.

Since we are solving Eq. (5.6) using a generic nonlinear optimizer, providing gradients of the objective and constraints can greatly improve the speed and accuracy of the optimization. For the objective function, computing the gradient is straightforward as the objective function is linear. All constraints—except the flutter constraint—are also all linear or at most quadratic, for which it is straightforward to derive the gradients. However, the flutter constraint is a highly nonlinear function of the cross-sectional area of each strut.

Computing the gradient through finite differences or forward differentiation is too expensive for the flutter constraint, as gradient computation using such forward methods scales with the input dimension, and there can be thousands of struts in the wing lattice. Adjoint differentiation on the other hand, scales with the output dimension. Here, we are only interested in one output: the aeroelastic eigenvalue. We therefore implement an adjoint to efficiently compute the gradient of the maximum eigenvalue with respect to the cross-sectional areas of the struts. We derive the adjoint for this problem from adjoints of elementary matrix operations, based on the work by Giles. The adjoint derivation is discussed in detail in Appendix D.

Griewank states that under realistic assumptions, computing a gradient of a function should be no more than five times as expensive as the underlying function. In our implementation the adjoint computation is only around 3.5× more expensive than the evaluation of the maximum eigenvalue of the system, starting from updated cross-sectional areas for the lattice.

5.2 VERIFICATION OF LOW-ORDER STRUCTURAL MODEL

The low-order beam model has to be validated against its full-order lattice model. We compare the frequencies and mode shapes for a long and slender beam, as computed by the full-order lattice model and the low-order beam model. A symmetric lattice that is used in the first part of this section is shown Figure 5.4. This lattice is symmetric with respect

165. Ringertz, On Structural Optimization with Aeroelasticity Constraints. 1994


166. Giles, Collected Matrix Derivative Results for Forward and Reverse Mode Algorithmic Differentiation. 2008

167. Giles, An Extended Collection of Matrix Derivative Results for Forward and Reverse Mode Automatic Differentiation. 2008

to the $yz$ plane with the elastic axis and center of gravity coinciding with the mid-chord line.

The frequencies of the model are listed in Table 5.1 for the first four bending modes and first two torsion modes using a beam model with 10 beam sections. The error between low-order and full-order model is below 4% in all cases, which is deemed sufficiently accurate in this study, as the error is substantially lower than the spacing of frequencies between different modes. Note that the error increases for higher mode numbers, which is the result of the discretization in the beam model. If the number of beam sections in the beam model is increased, the error in the higher modes goes down substantially.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Lattice model, Hz</th>
<th>Beam model, Hz</th>
<th>Error, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bending (I)</td>
<td>0.9075</td>
<td>0.9055</td>
<td>0.22</td>
</tr>
<tr>
<td>Bending (II)</td>
<td>5.263</td>
<td>5.306</td>
<td>0.81</td>
</tr>
<tr>
<td>Torsion (I)</td>
<td>9.152</td>
<td>9.111</td>
<td>0.45</td>
</tr>
<tr>
<td>Bending (III)</td>
<td>14.24</td>
<td>14.52</td>
<td>1.9</td>
</tr>
<tr>
<td>Bending (IV)</td>
<td>26.73</td>
<td>27.82</td>
<td>3.9</td>
</tr>
<tr>
<td>Torsion (II)</td>
<td>28.38</td>
<td>28.42</td>
<td>0.14</td>
</tr>
</tbody>
</table>

Table 5.1: Comparison between frequencies obtained from the full-order lattice model and the low-order beam model.
The mode shapes are also compared between the low-order beam model and full-order lattice model. The second bending mode shape is shown in Figure 5.5. These shapes also match well between the low-order model and full-order model. A comparison between the low-order model and full-order model for the first torsion mode is shown in Figure 5.6, and again these shapes match quite well between the low-order and full-order model.

Lastly, we also compare the structural frequencies between the low-order and full-order model for a lattice where neither the elastic axis, nor the center of gravity position coincides with the mid-chord line. This lattice has an asymmetric orientation for its struts as well as different thicknesses for several struts (Figure 5.7).

<table>
<thead>
<tr>
<th>Mode</th>
<th>Lattice model, Hz</th>
<th>Beam model, Hz</th>
<th>Error, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode I</td>
<td>1.103</td>
<td>1.112</td>
<td>0.81</td>
</tr>
<tr>
<td>Mode II</td>
<td>6.620</td>
<td>6.398</td>
<td>3.5</td>
</tr>
<tr>
<td>Mode III</td>
<td>9.716</td>
<td>9.433</td>
<td>3.0</td>
</tr>
<tr>
<td>Mode IV</td>
<td>17.59</td>
<td>16.94</td>
<td>3.7</td>
</tr>
</tbody>
</table>
The structural frequencies for the lattice in Figure 5.7 obtained from both the low-order model (10 beam sections) and full-order model are compared in Table 5.2. Again, the maximum error is less than 4%, with the error increasing for higher frequencies.

5.3 TAILORING OF WING IN TRANSONIC FLOW

This section demonstrates the aeroelastic tailoring of lattice structures to mitigate flutter. First, we optimize the internal lattice of the wing without considering flutter and show the flutter boundary and structural modes of that design. Second, we optimize the internal lattice again, but now including a flutter constraint in the optimization. We then compare the flutter characteristics of the two wings as well as their relative mass. It is found that optimizing the wing to obtain a 15% higher flutter speed only makes the structure 1.8% heavier.

Figure 5.8: Von Mises stress throughout design space for wing in transonic flow.

Figure 5.9: Original lattice that is optimized without considering aeroelastic instabilities.
For the load case considered, the Von Mises stress through the solid internal domain of the wing is shown in Figure 5.8, as computed using the solver discussed in Appendix B. The root is clearly highly loaded—especially near the center of the wing—whereas the tips are lightly loaded. The stress tensor for this load case is then used to generate a metric field, which in turn is used to generate the lattice topology. The struts of that lattice are subsequently optimized for minimum weight of the wing, subject to stress constraints—note that we will consider flutter constraints later in this section. The original optimized lattice—without considering flutter—is shown in Figure 5.9. In this lattice the largest strut radius is allowed to be $2.5 \times$ larger than the smallest strut radius.

The flutter boundary for the original lattice is shown in Figure 5.10. The model clearly exhibits typical transonic dip behavior, where the flutter speed is lowest for transonic Mach numbers and higher for subsonic and high transonic Mach numbers.$^{15}$

![Figure 5.10: Flutter boundary for both the original lattice and the aeroelastically tailored lattice.](image)

Problematically, the flutter speed $V_f$ in Figure 5.10 is equal to the dive speed $V_{dive}$, whereas typically the flutter speed needs to be 15% higher than the dive speed.$^{16}$ We therefore optimize the lattice again, but now add a flutter constraint that ensures the wing’s flutter speed is at least 15% higher than the dive speed. The resulting aeroelastically tailored lattice is only 1.8% heavier than the original lattice. The flutter boundaries for both the original and the aeroelastically tailored structure are shown in Figure 5.10, which shows that the entire flutter boundary moves up for the aeroelastically tailored lattice. Figure 5.11 shows the final optimized

![Figure 5.11: Aeroelastically tailored lattice. The cross-sectional area of each red strut is more than 2.5% larger than the cross-sectional area of the corresponding strut in the original lattice, whereas the cross-sectional area of each blue strut is smaller than the corresponding original strut by 1% or more.](image)
lattice, where its struts are colored according to whether their cross-sectional area increased or decreased compared to the original lattice. We see that especially in the outboard region of the wing and near the trailing edge, the cross-sectional areas are increased. The center of gravity position is therefore changed considerably, as is the overall stiffness of the wing, leading to different aeroelastic behavior. In particular, the stiffness of the outboard section of the wing is increased, as that part of the wing is most flexible considering it has fewer struts, which are also fairly thin (Figure 5.9). The change in stiffness of the internal structure of the wing, can be clearly deduced from the structural frequencies of the eigenmodes (Figure 5.12), which are all at least 2.4% higher for the aeroelastically tailored lattice.

Figure 5.12: Eigenmodes for optimized wing lattice. These are essentially identical to the eigenmodes of the original wing lattice.

(a) 1st mode (bending), frequency 3.3% higher for tailored lattice

(b) 2nd mode (coupled bending-torsion), frequency 3.4% higher for tailored lattice

(c) 3rd mode (torsion), frequency 2.4% higher for tailored lattice
Finally, another way the flutter speed can be increased is by increasing the mass of a structure uniformly. It is known that the flutter speed scales with $\sqrt{m_{\text{wing}}}$. Therefore, to increase the flutter speed for this problem by adding mass, we would need to make the entire lattice at least 32% heavier. However, because during the optimization the mass of each strut is optimized—which changes the mass of the structure, its stiffness, and its center of gravity position—the mass increase for the aeroelastically tailored lattice is substantially lower. Because there is a great deal of design freedom to do that in the lattice—in this example the lattice has 2,870 struts—the aeroelastic properties of the wing can be changed with only a minimum mass addition.
CONCLUSIONS AND OUTLOOK

This thesis presented low-order methods for transonic flutter prediction and flutter mitigation. First, a physics-based low-order model was developed for predicting aeroelastic instabilities in transonic flow to be used in the conceptual design of next-generation commercial transport aircraft. Second, we developed a design methodology for additively-manufactured lattice structures which allows for mitigating flutter with only minimal mass addition. The main findings and contributions are highlighted here, together with avenues for further research as well as additional potential applications of this work.

6.1 CONTRIBUTIONS

For the design of next-generation transport aircraft, it is crucial to include transonic flutter considerations early in the conceptual design, as well as have efficient strategies to mitigate flutter. In this thesis, therefore, we developed a physics-based low-order model for transonic flutter prediction that is cheap enough to be used in the conceptual design phase, and we have used this model to describe the influence of transonic flutter on the design of next-generation transport aircraft. To mitigate any potential flutter problems efficiently, we have developed a design methodology for lattice structures that takes advantage of the geometric design freedom offered by additive manufacturing, and have used this design methodology to design an internal wing structure while mitigating flutter with a minimal mass increase.

The physics-based low-order model is based on the lowest moment of vorticity—the circulation strength—and the lowest \( x \)-moment of the volume-source field—the doublet strength. These two state variables are used to construct an aerodynamic state-space formulation with several unknown parameters. These unknown parameters can be calibrated using dynamic mode decomposition from high-fidelity unsteady Euler simulations over pitching and heaving airfoils for several Mach numbers, baseline lift coefficients, and reduced frequencies. The approach has been demonstrated by calibrating the model with unsteady CFD simulations of a pitching and heaving RAE2822 in subsonic and transonic flow. The calibrated model was evaluated for its ability to predict the unsteady response over a relatively wider range of flows than was used in the calibration set, where it was confirmed that the circulation strength
and doublet strength match up between the low-order and full-order model. Furthermore, the accuracy of the flutter prediction capability of the low-order model is confirmed for a classic flutter test problem from the literature—Isogai Case A—where the flutter boundary from the low-order model matched up with the flutter boundary from much more expensive CFD simulations. To demonstrate the benefits of using this model in the conceptual design phase, the influence of baseline lift coefficient, freestream Mach number, and airfoil geometry on the flutter boundary is investigated using the low-order model, where even the influence of small geometry changes in the airfoil were picked up by the model.

The low-order model for flutter prediction of airfoils can also be applied to the flutter prediction of aircraft wings through strip theory. In this approach, the wing is discretized in several sections and an aeroelastic system for the whole wing is formed using calibrated flutter coefficients for transonic flow. Such a system is low-dimensional, allowing for fast computation of the aeroelastic eigenvalues, and therefore making it applicable for use in a conceptual design tool. The wing flutter model's accuracy is confirmed for the Benchmark Supercritical Wing from the Aeroelasticity Prediction Workshop, as this low-order flutter model is as accurate as much more expensive 3D Euler or Navier-Stokes flutter solutions. Furthermore, the model’s predictive capability is demonstrated by finding transonic flutter boundaries for several wing configurations, showing the influences of Mach number, aspect ratio, taper ratio, engine location, and sweep on the aeroelastic behavior of clamped wings. The effect of transonic flow on the flutter behavior of swept wings is particularly strong.

By implementing the wing flutter model in a physics-based conceptual aircraft design tool, the influence of transonic flutter on next-generation aircraft designs can be quantified. The flutter behavior for two different aircraft configurations—one with a high-sweep wing with an engine attached below, one low-sweep wing with no engines attached to the wing—is quite different, and several trends of flutter damping values with respect to wing design parameters are reversed between the two configurations. For example, a larger sweep angle pushes the aircraft with wing-mounted engines towards instability, whereas for the aircraft with fuselage-mounted engines a larger sweep angle moves the design away from instability. Furthermore, it was found that the inclusion of flutter constraints in the aircraft design optimization limits the aspect ratio of the wings, resulting in a 3.3% higher fuel burn. Transonic flutter therefore limits the performance gains seen by using more advanced materials in the wing. Note that these trends are not generally applicable to any aircraft design; the onset of flutter is quite subtle and is sensitive to small changes in wing geometry, engine parameters, or aerodynamic
forces. This confirms that it is crucial to include an accurate, fast aeroelastic analysis early in the conceptual design phase of next-generation transport aircraft.

As a proof-of-concept study of how novel manufacturing techniques can aid in flutter mitigation, we turn to lattice structures manufactured using additive manufacturing. Lattice structures in particular are of interest, because of the large design freedom in their stiffness and center of gravity—making them ideal for aeroelastic tailoring—as well as their high stiffness-to-weight ratio. For this reason, we developed a design methodology for lattice structures to be manufactured using additive manufacturing. In this design method, first the lattice topology is designed using adaptive meshing techniques. A metric field is obtained from the stress tensor through the allowable design domain, which then informs a Riemannian metric field. Through this Riemannian metric field, a lattice topology can be designed that is fine near areas of high stress, coarse near areas of low stress, and is aligned with the stress direction. This approach takes advantage of the large design freedom allowed by additive manufacturing. However, some manufacturing constraints remain for additive manufacturing, such as minimum feature size and overhang angle constraints. Designing around these manufacturing constraints is imperative, and particular focus has therefore been placed on incorporating those in the design methodology. Information about manufacturing constraints can in fact be encoded in the metric field, by orienting the metric in manufacturable directions. The lattice topology is then generated from the metric field using a metric-based mesh generator, typically used for CFD meshes. Once the lattice topology is designed, the cross-sectional area of each lattice strut is optimized for minimum weight subject to stress, compatibility, and buckling constraints. This design methodology is applied to a small bracket to show the manufacturability of the part, where it was found that the design for manufacturability improves the manufacturability of the lattice substantially. Lastly, an aircraft bracket from the literature is redesigned to achieve an 80% weight improvement over the original solid bracket design.

This design methodology can also be used to design lattice structures for an internal structure of a wing, while tailoring the area of each strut in that lattice for minimum weight and while mitigating aeroelastic instabilities. To keep the cost of evaluating the aeroelastic stability of the wing manageable, a low-order structural model is developed for the lattice structure. This low-order model is coupled to the physics-based low-order aerodynamic model to evaluate the aeroelastic eigenvalues of the system. The low-order structural model is derived using mappings between the deflections of the full-order lattice model and the equivalent beam model, and mappings between the forces on each node of the
lattice to the moments and forces on the beam model. The structural frequencies of the low-order beam model have been compared against the structural frequencies of the full-order lattice model, where a maximum error of only 4% was observed for higher modes. This flutter model is then used to predict the flutter of the lattice structure and is included in the optimization problem to find the optimal cross-sectional area of each of the members of the lattice, while avoiding aeroelastic instabilities. This approach to aeroelastic tailoring is applied to the design of the internal structure of a wing in transonic flow. It was found that including a flutter constraint in the optimization increases the flutter speed by 15%, while the weight of the structure is increased by only 1.8%, compared to the design where flutter was not constrained.

It is clear that transonic flutter plays an important part in the design of next-generation transport aircraft, and the research presented in this thesis enables incorporating flutter requirements into early-stage design. Advances in manufacturing techniques play an important role as well in future aircraft designs, which the research presented in this thesis exploits to design more efficient structures and to aeroelastic tailor wing structures efficiently.

6.2 OPPORTUNITIES FOR FURTHER RESEARCH

In this thesis, a low-order flutter prediction method is developed and its use is demonstrated for a conceptual design study. That model can therefore be readily implemented in other (industry-based) conceptual design tools as well. The design methodology for additive manufacturing as well as the aeroelastic tailoring approach are more proof-of-concept studies and there are therefore a number of avenues for future research.

The accuracy of the physics-based transonic flutter model could be further improved by training the low-order method using unsteady Reynolds Averaged Navier-Stokes (URANS) solutions to capture shock-boundary layer interactions. The model developed in this thesis is able to capture such effects because the vorticity of the boundary layer can be lumped into the circulation strength. However, in order to generate a batch of high-fidelity URANS solutions accurately—i.e., where the solution for each baseline lift coefficient, Mach number, and Reynolds number is converged and has the correct $y^+$ value—a space-time adaptive solution is most likely necessary. Such a solution procedure could be built on the work of Fidkowski et al.\textsuperscript{170,171} and Barral et al.\textsuperscript{172} although it should also be investigated whether a $(2 + 1)d$ space-time adaptive solution is more efficient.

To improve the fidelity of the flutter prediction method for aircraft wings, more vibration modes could be considered. In the current model, the fuselage is considered rigid, whereas the fuselage flexibility can lead to

\textsuperscript{170} Fidkowski et al., Output-Based Space–Time Mesh Adaptation for the Compressible Navier–Stokes Equations. 2011

\textsuperscript{171} Kast et al., Output-Based Mesh Adaptation for High-Order Navier–Stokes Simulations on Deformable Domains. 2013

\textsuperscript{172} Barral et al., Time-Accurate Anisotropic Mesh Adaptation for Three-Dimensional Time-Dependent Problems with Body-Fitted Moving Geometries. 2017
coupled fuselage and wing modes, particularly for long fuselages. Given that the fuselage cross-sectional area of the D8.x aircraft is larger than current commercial aircraft, this is expected to be less of a problem for that concept, however, the fuselage flexibility could be more important for other next-generation transport aircraft. Furthermore, the flexibility in the engine strut should also be taken into account, as this flexibility typically gives rise to a vibration mode with low damping at low speed. The engine strut is considered rigid in this work, but its elastic behavior can be included in the beam model by adding states to the dynamic system.

Specifically for the design methodology for additive manufacturing of lattice structures, there are multiple further avenues to explore. The current methodology allows for fast generation of lattice structures and obtain an estimate of the material weight and cost, which is appropriate for early design stages. However, several approaches can be taken to improve the methodology for early design stages, as well as for the detailed design phase.

To guarantee 100% manufacturability consistently, additional manufacturability constraints should be included in the design process to guarantee manufacturability for any additive manufacturing process. Warping of the part during manufacturing due to thermal stresses, for instance, is often the cause of print failure, especially for metal additive manufacturing processes. Optimizing the part together with the toolpath of the additive manufacturing process with limits on the thermal stresses could solve this problem. Furthermore, to improve the design for additive manufacturing aspect, further methods should be explored to guarantee 100% manufacturability of the struts by supporting internal struts better, rather than deleting them from the final structure.

With regard to manufacturing constraints, better data is also needed on exactly what these constraints are for each manufacturing process, perhaps even for each machine. An overhang constraint of 45°, for instance, is quite generic and usually conservative. Usually, there is a length limit beyond which an overhang constraint actually matters. A manufacturing constraint is therefore more likely a function of both the overhang angle and the thickness, shape, and length of a feature. Developing an experimental strategy by which these constraints can be inferred for different additive manufacturing machines is therefore crucial to guarantee a design that is both optimal and manufacturable.

Beyond manufacturing constraints, the influence of the build direction on failure modes should also be taken into account in the design. We have assumed that the material properties are isotropic, whereas especially for wire-feed manufacturing process, parts have anisotropic behavior in the print direction. Further, the build direction can also have an influence on crack propagation. Surface irregularities promote
crack growth, while additively manufacturing parts typically exhibit such irregularities. Rans et al. recently studied the effect of build direction on crack propagation in additively manufactured metal parts under a static load, where it was found that for that case no influence of build direction was noticeable. Such studies should be performed for multiple materials and manufacturing processes—also under cyclic loading to investigate the fatigue behavior—and the results from such studies should be included in the part design. This allows for optimizing for the correct part orientation, such that a trade-off between part performance, part build time, failure modes, and amount of support is possible.

Regarding the toolpath used by the manufacturing process, this could be optimized as well together with the infill pattern, as has been demonstrated by Clausen et al. for a density-based topology optimization method. Furthermore, we have only considered solid struts in this work, whereas using hollow struts would probably result in a more efficient final design. However, for a powder-based or stereolithography manufacturing process, care then has to be taken that powder or liquid polymer can escape the internals of the part after the print is done.

For the detailed design of these lattice structures, the fidelity of the lattice optimization needs to be improved. Right now, the lattice structure is optimized by solving a truss optimization problem where it is assumed that all struts are two-force members—an assumption that can get violated especially under dynamic loads. It is more realistic to use a model that takes bending and torsion of the lattice struts into account. If large deformations are expected—for instance in impact structures or morphing applications—a nonlinear lattice model should be used, for instance using Crosserat rods. Such lattice models, however, do not take stress concentrations into account, whereas lattice structures are typically susceptible to stress concentrations due to the complicated node geometry. To alleviate this concern, fillets around the lattice nodes should be added, which further complicates the lattice geometry generation.

To design the lattice topology, we rely heavily on metric-based mesh generators. We have used a mesh generator that does not project the lattice back to the original geometry and does not curve the faces of the mesh to match the geometry. Instead, a large number of nodes are placed near areas of high curvature to conform to the geometry. This results in material being added to a part of the structure just because a face is curved, rather than because it is highly loaded in that area. A curved mesh instead would allow for curving the struts as well to conform to the geometry—note that this would also require a higher-fidelity structural lattice model as those curved struts are no longer two-force members. Some preliminary work in this area has recently been presented by Feuillet et al.

Throughout this work, the lattice structures were designed using...
triangles or tetrahedra as base elements. However from these meshes, we can also generate a Voronoi diagram,\textsuperscript{178} which in turn also has a notion of nodes and edges that can be used to generate a lattice structure. It should be investigated whether using such a Voronoi structure results in weight savings. Furthermore, it is expected that lattices using such a Voronoi structure are more useful for parts where larger deformations are required, such as in impact structures or morphing applications.

To reduce the dimensionality of the design problem, we have split the design of the lattice topology from the strut area optimization. However, to obtain the absolute minimum part weight, most likely the full design problem needs to be considered. As an initial guess, however, our design method for the lattice topology can still be used. Further, we assign each strut the minimum manufacturable cross-sectional area to guarantee manufacturability. However, not all of these struts are necessary. More performance can be extracted from the parts if some struts are allowed to be deleted, while still guaranteeing manufacturable of the parts. To design such an optimization strategy, one can likely draw on ideas from dynamic programming\textsuperscript{179} to guarantee that each strut is supported by at least one strut below.

In this work, we fill the entire available design space with a lattice structure and use a coarse lattice near areas of low stress. However, the performance of the part will likely be improved if the outer shape of the geometry is allowed to change. Such a shape optimization approach can for instance be used in the \textit{g\_e} bracket example, where the back of the part can likely be moved inwards as that region does not carry any load. This shape optimization approach of the internal structure can then also be combined with aerodynamic performance of a wing, leading to an aerostructural optimization problem to improve the overall fuel efficiency of an aircraft.

In the design of the internal structure of a wing, we have assumed that the wing skin does not take any load, which is typical for a structure where the wing is wrapped in a thin film of plastic. However, for most uavs or passenger aircraft, the wing skin is actually highly stressed, leading to lighter internal structures. We can take the effect of a wing skin into account by using a shell model for the boundary faces of the lattice and assigning them a thickness. Our lattice design methodology is built on mesh data structures—which have a notion of faces—and we can therefore include the thickness of faces into the optimization. Beyond wing skins, combining these lattice structures with shell models can result in even higher performance parts, as closed wall parts are typically stiffer structures than lattices alone according to Sigmund et al.\textsuperscript{180} By extending this design methodology to delete struts that are not necessary and to allow for closed walled parts, this approach could perhaps get close to the result of Aage et al.\textsuperscript{70} at a fraction of the cost.
6.3 Potential Additional Applications

The computational methods presented in this thesis have potential applications beyond the example applications used in this thesis. In the following, we highlight some potential future applications, although this is by no means an exhaustive list.

Transonic Flutter Prediction

Following a similar approach to our physics-based low-order model development, a physics-based low-order flutter prediction method could be developed for rotorcraft, propellers, or turbine blades. Furthermore, the aeroelastic tailoring approach using lattice structures may also be of interest for these applications. However, if this method is applied to the design of aircraft with propellers, care has to be taken to include whirl flutter in the design.\textsuperscript{181}

Flutter Prediction for Other Aircraft Concepts

Several next-generation transport aircraft concepts feature high-aspect ratio wings—besides the D8.x aircraft—for which our flutter prediction method could be used. Especially when those concepts feature four engines—for which flutter is typically a critical design constraint—using an accurate transonic flutter model is paramount. However, the current model could not be used for the Truss-Braced Wing concept,\textsuperscript{9} because the current structural model does not take the strut into account. Furthermore, the influence of the strut on the flutter behavior needs to be investigated. The position of the strut could be another calibration parameter in the physics-based transonic flutter model.

Design of Self-Supporting Lattice Structures

Lattice structures are of interest for their high stiffness-to-weight ratio, but also because they typically allow multiple load paths throughout a structure, which are critical for a damage tolerant part.\textsuperscript{182} Using a more formal way of incorporating such fail-safety in the design would help speed up the certification of additively manufactured parts for safety-critical parts. Furthermore, this could make lattice structures of interest to applications where vehicles are able to keep operating even in non-pristine conditions, as long as the amount of damage is known or well approximated. This helps save maintenance costs and limits downtime of vehicles.

Because lattice structures allow for multiple load paths through a structure, they are typically also attractive as impact structures. As struts in the lattice start to fail, other struts take more load, and as these struts fail, others take the load, and so on, until almost all struts have failed.

\textsuperscript{181} Kunz, Analysis of Proprotor Whirl Flutter: Review and Update. 2005

\textsuperscript{9} Bradley et al., Subsonic Ultra Green Aircraft Research: Phase I Final Report. 2011

\textsuperscript{182} Venkataraman et al., Investigating Alternate Load Paths and Damage Tolerance of Structures Optimized for Multiple Load Cases. 2009
This leads to a structure that can take up a large amount of energy before completely failing. However, impact structures are typically also heavy—even when lattice structures or honeycomb material is used. Using our design methodology could make these structures lighter by tailoring the lattice structures to only take up energy in the direction of expected impact.

Lattice structures are also used for thermal applications, as they have a high surface-to-volume ratio.† An interesting future application of our design methodology is to design the lattice topology such that there is maximum flow through the lattice or such that the heat dissipation of the structure is at its maximum.

Lastly, this design methodology could also be applied in different industries, in particular the medical industry. The medical industry has been at the forefront of using additive manufacturing for final products, as it allows for just-in-time manufactured patient-specific parts. Lattice structures are used for bone implants, because they can be tailored to match the stiffness of the bone around them and are porous, which promotes bone tissue growing on them.¹⁸³ Our design methodology could improve the patient-specific stiffness tailoring, as well as improve the manufacturability of the implants.

### Mesh Generation for Lattice Structures

The geometry generation algorithm in Appendix C is used here to generate STL files such that the lattice structures can be manufactured. However, the same algorithm can also be used to generate a surface mesh, which in turn can be used to generate an internal FEM mesh to solve for a high-fidelity stress distribution throughout the lattice.

### Aeroelastic Tailoring

Besides flutter mitigation, the aeroelastic tailoring approach can also be used to achieve a desired deflection at a given load. For the automotive/motorsport industry in particular this could be of interest for adaptive cooling at different speeds by deforming turning vanes or for improving the effectiveness of wings by deflecting them closer to the ground at high speed.

### Wind Tunnel Models

Designing wind tunnel models with correct aeroelastic properties can be costly and time-consuming.¹⁸⁴ Our design methodology for lattice structures could be used to design the internal structure for wind tunnel models to ensure that the overall wind tunnel model matches aerodynamic and aeroelastic nondimensional parameters. Such an approach could then also be used to design wind tunnel models specifically for

---

†The high surface-to-volume ratio can also be a disadvantage, as this makes the structure more susceptible to corrosion.
flutter validation of numerical methods to tailor the structure to exhibit particular flutter modes.

**INTERNAL STRUCTURE FOR WINGS**

In this thesis, the aeroelastic tailoring approach has been used for a notional internal wing structure, which demonstrated the effectiveness of the approach. However, that lattice was still fairly coarse. Because the methodology is based on adaptive meshing techniques—which can easily handle tens of thousands of elements—the design methodology actually scales very well with the number of nodes in the lattice. This methodology can therefore be used to design quite fine lattice structures with fine details where point loads are introduced into the wing, such as near an engine pylon or flap attachment.‡

It will take some time to reach the level of technical maturity required to build such intricate lattice structures for as the internal structure of a commercial jetliner wing. However, however, building the internal structure of a (smaller) UAV wing is much more feasible with current or near-future additive manufacturing tools. This would allow for on-site production and repair in the field—think of printing parts of at distribution centers for delivery drones or at a response center for search and rescue missions—especially as no support structures are necessary, limiting the need for post-processing.

These future applications highlight the start of what is possible in the realm of aeroelasticity in years to come, taking advantage of new computational methods and manufacturing technologies. This thesis has laid the groundwork for the realization of these future applications.

‡ Note that in such large lattice structures, the cross-sectional area optimization of the lattice struts is likely the most expensive step, which needs to be addressed using for instance first-order optimization methods.
BIBLIOGRAPHY


cited on p. 28


cited on p. 28


cited on p. 28


doi: 10.2514/3.61226

cited on pp. 28, 53, 54


cited on pp. 28, 29


doi: 10.2514/3.60855

cited on pp. 28, 29


doi: 10.2514/3.25127

cited on p. 29


doi: 10.1137/130932715

cited on p. 29


doi: 10.2514/1.9839

cited on p. 29


doi: 10.1090/qam/910462

cited on p. 29


doi: 10.1146/annurev.fluid.25.1.539

cited on p. 29

cited on p. 29


cited on p. 29


cited on p. 29


cited on p. 29


cited on p. 29


cited on p. 29

cited on pp. 29, 39, 41

cited on pp. 29, 39, 41, 62


cited on p. 30


cited on p. 30


cited on p. 30


BIBLIOGRAPHY


cited on p. 86


cited on p. 88


cited on p. 89


cited on p. 89


cited on p. 90


cited on p. 91


cited on pp. 94, 154


cited on p. 103

166 Giles, M. B., *Collected Matrix Derivative Results for Forward and Reverse Mode Algorithmic Differentiation*, *Advances in Automatic Differentiation*, Springer Berlin Heidelberg, 2008, pp. 35–44. DOI: 10.1007/978-3-540-68942-3_4

cited on pp. 103, 163, 164

167 Giles, M., “An Extended Collection of Matrix Derivative Results for Forward and Reverse Mode Automatic Differentiation,” 08/01, Computing Laboratory, Oxford University, Oxford, United Kingdom, 2008.

cited on pp. 103, 163, 164, 165


cited on p. 103


cited on p. 107


DOI: 10.1214/aoms/1177730148


ISBN: 9780471986331

DOI: 10.1002/bimj.4710310722
Throughout Chapters 2, 3 and 5 flutter boundaries for varying airfoil or wing parameters are computed. In this appendix, we provide a more detailed explanation of how to compute such a flutter boundary. Crucial to computing the flutter boundary cheaply is the data fitting from CFD, which has to be done nondimensionally. Once the model is calibrated nondimensionally, the flutter boundary can be computed quickly using a bisection method.

Transonic inviscid unsteady flow for a given oscillating airfoil can be uniquely described using just three nondimensional parameters. Typically, the Mach number $M_\infty$, angle of attack $\alpha$, and Strouhal number $St$ are used. In this work, instead of the angle of attack, we use the baseline lift coefficient $c_{\ell_0}$, and instead of the Strouhal number $St$, we use the reduced frequency $\overline{\omega}$. Note that if viscous effects are taken into account, the Reynolds number $Re$ is required as an additional nondimensional parameter to describe the flow field uniquely. Therefore, the flutter coefficients $A(\cdot)$ and $B(\cdot)$ also have to be a function of $M_\infty$ and $c_{\ell_0}$. Note that by assuming a linear state-space system for the aerodynamic model (2.12), these coefficients cannot be a function of $\overline{\omega}$, but the influence of $\overline{\omega}$ is of course taken into account in the state-space system itself.

To fit the flutter coefficients nondimensionally, nondimensional state variables and nondimensional input parameters need to be used in the DMD fitting. The nondimensional $x$-doublet strength $\kappa_x$ and circulation strength $\Gamma$ are defined as

$$\kappa_x = \frac{\kappa_x}{V_\infty c^2}, \quad \Gamma = \frac{2\Gamma}{V_\infty c}.$$  

The nondimensional input variables then are the $\Delta \theta$, $\Delta \omega \ b / V_\infty$, and $\Delta h / V_\infty$. The independent variable—in this case, time $t$—also has to be nondimensionalized as $\bar{t} = t \ V_\infty / b$. If this independent variable and these input and output parameters are used, the resulting flutter coefficients are only dependent on the freestream Mach number $M_\infty$, the baseline lift coefficient $c_{\ell_0}$, and the airfoil shape, while the flutter coefficients are independent of the chord $c$ and freestream velocity $V_\infty$. The (nondimensional) aerodynamic response therefore is only a function of $M_\infty$, $c_{\ell_0}$, airfoil shape, and the reduced frequency $\overline{\omega}$.

To include the nondimensionalized aerodynamic system in the overall aeroelastic state-space system, either the structural model can be changed to use the nondimensional time as its independent parameter,
or the aerodynamic system can be converted into dimensional form. In this work, the latter approach is taken.

To compute the flutter boundary then, we need to find that value of the freestream velocity for which the aeroelastic system has a maximum real part equal to 0. We employ a bisection method for this purpose. This approach is quite straightforward: starting from a window of freestream velocities within which we know the aeroelastic system turns unstable, we successively narrow this window as we evaluate the stability of the aeroelastic system at the center of this window. If the system is unstable there, the upper bound is changed to the freestream value at which we just evaluated the stability. If the system is stable, the lower bound is changed to that freestream value. This continues until the size of the window is below some tolerance. This process is also illustrated in Figure A.1. Note that this approach is quite cheap, as it only involves eigenvalue computations of $7 \times 7$ systems in the 2D case. The flutter boundary in Figure A.1 is computed in only $0.09s$ for 28 different Mach numbers, which requires 1228 different aeroelastic stability evaluations.

![Figure A.1: Illustration of using the bisection method to find the flutter speed at $M_\infty = 0.65$.](image-url)
FINITE ELEMENT SOLVER
FOR LINEAR ELASTICITY

This appendix discusses the FEM solver for the linear elasticity equations used in Chapters 4 and 5. The solver’s HDG discretization is discussed in detail (Appendix B.1), while the solver’s accuracy has also been verified for test problems in 2D and 3D (Appendix B.2).

The full linear elasticity equations are\(^\text{151}\)

\[
\nabla \cdot \sigma + f = 0 \\
\epsilon = \frac{1}{2} \left( \nabla u + (\nabla u)^T \right) \\
\sigma = C : \epsilon
\]

(4.1a revisited)  
(4.1b revised)  
(4.1c revised)

where \(\sigma\) is the Cauchy stress tensor, \(\epsilon\) is the infinitesimal strain tensor, \(u\) is the displacement vector, \(C\) is the fourth-order stiffness tensor, and \(f\) is the body force per unit volume. \(C\) is defined as

\[
C_{ijkl} = \lambda_1 \delta_{ij} \delta_{kl} + \mu_s \left( \delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} - \frac{2}{3} \delta_{ij} \delta_{kl} \right)
\]

with \(\lambda_1\) Lamé’s first parameter, and \(\mu_s\) the shear modulus (or rigidity), both of which are elastic moduli, defined as

\[
\lambda_1 = \frac{E \nu}{(1 + \nu) (1 - 2\nu)} \\
\mu_s = \frac{E}{2 (1 + \nu)}.
\]

B.1 DISCRETIZATION

We use a Finite Element Method (FEM) to solve the linear elasticity equations, specifically a Discontinuous Galerkin (DG) method. DG methods are particularly applicable to problems where the solution exhibits complex structures or difficult-to-approximate structures, because these methods have a built-in stabilization mechanism resulting in a robust method with no loss in accuracy.\(^\text{185,186}\) A Hybridizable Discontinuous Galerkin (HDG) discretization is a more efficient variant within the DG methods, as only the displacements on the element borders are globally coupled degrees of freedom—compare Figure B.1b to Figure B.1c.

The resulting global stiffness matrix is symmetric, positive definite, and has a block-wise sparse structure. Figure B.1c shows the globally
coupled degrees of freedom together with the local degrees of freedom for the HDG method. Figure B.1d shows the discretization for the Embedded Discontinuous Galerkin (EDG) method which further reduces the number of degrees of freedom, but does not retain all convergence properties from HDG. For second-order elliptic problems, HDG should therefore be used over EDG. Thus, this work uses an HDG method.

A linear elasticity problem with Dirichlet and Neumann boundary conditions on the domain \( \Omega \) can be written in Einstein notation as

\[
\begin{align*}
\sigma_{ij,j} + f_i &= 0, \quad \forall x \in \Omega \\
\epsilon_{ij} - \frac{1}{2}(u_{i,j} + u_{j,i}) &= 0, \quad \forall x \in \Omega \\
\sigma_{ij} - C_{ijkl}\epsilon_{kl} &= 0, \quad \forall x \in \Omega \\
u_i = \vec{n}_i, & \quad \forall x \in \partial\Omega_D \\
\sigma_{ij}n_j &= t_i, \quad \forall x \in \partial\Omega_N
\end{align*}
\]

where \( \Omega \) is the domain on which the problem is solved, \( \partial\Omega_D \) is the part of the boundary with a Dirichlet boundary condition, and \( \partial\Omega_N \) is the part of the boundary with a Neumann boundary condition.

Following Soon et al. the triangulation of the domain \( \Omega \) is denoted as \( T^h := \{ T \} \), where the elements of \( T \) are simplices, and the faces of this
triangulation are denoted as \( \mathcal{E}^h \). The boundaries of the elements are in \( \partial T^h = \{ \partial T : T \} \). For the integrals in the HDG formulation, the following notation is used: \( (a, b)_T := \int_T a \, b \, d\Omega \) and \( (a, b)_{\partial T^h} := \sum_{T \in \partial T^h} (a, b)_T \). Similarly, \( (a, b)_{\partial T} := \int_{\partial T} a \, b \, d\Gamma \) and \( (a, b)_{\partial T^h} := \sum_{T \in \partial T^h} (a, b)_{\partial T} \).

In the HDG method, we attempt to find an approximation \( u^h \) to the exact displacement \( u \), \( \sigma^h \) to the exact stress tensor \( \sigma \), and \( \epsilon^h \) to the exact strain tensor \( \epsilon \). These approximations exist in the spaces \( (u^h, \sigma^h, \epsilon^h, \hat{u}^h) \in V^h \times W^h \times Z^h \times M^h \) which are defined as

\[
V^h = \{ v \in L^2(\Omega) : v_i|_T \in \mathcal{P}(T) \ \forall T \in T^h, \ i = 1, 2, 3 \} \]

\[
W^h = \{ w \in L^2(\Omega) : w_{ij}|_T \in \mathcal{P}(T) \ \forall T \in T^h, \ i, j = 1, 2, 3 \} \]

\[
Z^h = \{ z \in L^2(\Omega) : z_{ij}|_T \in \mathcal{P}(T) \ \forall T \in T^h, \ i, j = 1, 2, 3 \} \]

\[
M^h = \{ \mu \in L^2(\epsilon^h) : \mu_i|_T \in \mathcal{P}(\epsilon) \ \forall e \in \mathcal{E}^h, \ i = 1, 2, 3 \}, \]

where \( L^2(\Omega) = \{ L^2(\Omega) \}^n \), \( L^2(\Omega) = \{ L^2(\Omega) \}^{n \times n} \) and \( \mathcal{P}(\mathcal{A}) \) is the space of polynomials of degrees \( k \) defined on the set \( \mathcal{A} \).

The approximate solutions are then found by solving the following weak statements\cite{soon2009}

\[
\begin{align}
(v_{ij}, \sigma_{ij}^h)_{T^h} - (v_{ij}, \sigma_{ij}^h n_j)_{\partial T^h} - (v_{ij}, f_i)_{T^h} &= 0 \quad \text{(B.1a)} \\
\left( w_{ij}, \epsilon_{ij}^h \right)_{T^h} - \frac{1}{2} \left( w_{ij}, \left( \hat{u}_{ij}^h n_j + \hat{u}_j^h n_i \right) \right)_{\partial T^h} + \frac{1}{2} \left( w_{ij}, u_j^h \right)_{T^h} &= 0 \quad \text{(B.1b)} \\
\left( z_{ij}, \sigma_{ij}^h \right)_{T^h} - \left( z_{ij}, C_{ijkl} \sigma_{kl}^h \right)_{T^h} &= 0 \quad \text{(B.1c)} \\
\left( \mu_i, \sigma_{ij}^h n_j \right)_{\partial T^h \setminus \partial \Omega_D} - (\mu_i, t_i)_{\partial \Omega_N} &= 0 \quad \text{(B.1d)} \\
\left( \mu_i, \hat{u}_i^h \right)_{\partial T^h \setminus \partial \Omega_D} - (\mu_i, \bar{u}_i)_{\partial \Omega_N} &= 0 \quad \text{(B.1e)}
\end{align}
\]

for all \( (v, w, z, \mu) \in V^h \times W^h \times Z^h \times M^h \), where

\[
\hat{\sigma}_{ij}^h = \sigma_{ij}^h - \tau_{ijkl} \left( u_{ik}^h - \hat{u}_k^h \right) n_i, \quad \forall x \in \partial T^h. \quad \text{(B.1f)}
\]

The system in Eq. (B.1) can be written in matrix form as

\[
\begin{bmatrix}
A & 0 & B & C \\
0 & D & H & J \\
K & M & 0 & 0 \\
N & 0 & P & Q
\end{bmatrix}
\begin{bmatrix}
S \\
E \\
U \\
G
\end{bmatrix}
= \begin{bmatrix}
F \\
0 \\
0 \\
0
\end{bmatrix}. \quad \text{(B.2)}
\]

The system in Eq. (B.2) is the full system for the problem. However, because only \( \hat{u}_h \) are globally coupled unknowns, we can dramatically reduce the size of the global system through a Schur complement yielding a global system of the form

\[
\hat{Q} \hat{U} = \hat{G}. \quad \text{(B.3)}
\]
Solving the reduced system in Eq. (B.3) yields the solution for $\hat{u}^h$. The solution to $u^h$, $\sigma^h$, and $\varepsilon^h$ is then found by unwrapping the Schur complement in each element separately—this process can therefore be performed completely in parallel. In this work, the solver is implemented in the technical computing language Julia.\textsuperscript{188}

B.2 VERIFICATION

To verify that the implementation of the HDG method is correct, we check the convergence rates of the discrete solution to the exact solution. Here we show the convergence order for problems with Dirichlet and Neumann boundary conditions in both 2D and 3D. We use the method of manufactured solutions for computing the convergence rates.

For a problem with only Dirichlet boundary conditions on a unit square in 2D, the exact solution used here is

\[
\begin{align*}
    u_1 &= 10 \left( x_2 - x_2^2 \right) \sin(\pi x_1) \left( 1 - x_1 \right) \left( 1 - \frac{x_2}{2} \right) \\
    u_2 &= 2 \left( x_1 - x_1^2 \right) \sin(\pi x_2) \left( 1 - x_2 \right) \left( 1 - \frac{x_1}{2} \right).
\end{align*}
\]

The convergence rates of the $L_2$ error in all quantities are shown in Table B.1a and Figure B.2a. We expect $p + 1$ convergence on all quantities, where $p$ is the polynomial order used in the solution. We see that for a problem with only Dirichlet boundary conditions, those convergence rates are observed.

<table>
<thead>
<tr>
<th></th>
<th>$p = 1$</th>
<th>$p = 2$</th>
<th>$p = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>1.99</td>
<td>2.99</td>
<td>3.99</td>
</tr>
<tr>
<td>$u_2$</td>
<td>1.96</td>
<td>2.99</td>
<td>4.00</td>
</tr>
<tr>
<td>$\sigma_{11}$</td>
<td>1.97</td>
<td>2.99</td>
<td>3.99</td>
</tr>
<tr>
<td>$\sigma_{12}$</td>
<td>1.93</td>
<td>2.96</td>
<td>3.97</td>
</tr>
<tr>
<td>$\sigma_{22}$</td>
<td>1.92</td>
<td>2.96</td>
<td>3.99</td>
</tr>
</tbody>
</table>

(a) Only Dirichlet boundary conditions

<table>
<thead>
<tr>
<th></th>
<th>$p = 1$</th>
<th>$p = 2$</th>
<th>$p = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>1.97</td>
<td>2.99</td>
<td>4.00</td>
</tr>
<tr>
<td>$u_2$</td>
<td>1.99</td>
<td>2.99</td>
<td>3.99</td>
</tr>
<tr>
<td>$\sigma_{11}$</td>
<td>1.95</td>
<td>2.98</td>
<td>3.98</td>
</tr>
<tr>
<td>$\sigma_{12}$</td>
<td>1.92</td>
<td>2.97</td>
<td>3.96</td>
</tr>
<tr>
<td>$\sigma_{22}$</td>
<td>1.96</td>
<td>2.98</td>
<td>3.99</td>
</tr>
</tbody>
</table>

(b) Dirichlet and Neumann boundary conditions

For a 2D problem with both Dirichlet and Neumann boundary conditions on a unit square, the exact solution used here is

\[
\begin{align*}
    u_1 &= \left( x_2 - x_2^2 \right) \cos(\pi x_1) \left( \frac{1}{2} - x_1 \right) \left( 1 - \frac{1}{2} x_2 \right) \\
    u_2 &= 2 \left( x_1 - x_1^2 \right) \sin(\pi x_2) \left( 1 - x_2 \right) \left( 1 - \frac{1}{2} x_1 \right),
\end{align*}
\]

where an inhomogeneous Neumann boundary condition is used at $x_1 = 0$ and $x_1 = 1$ and a homogeneous Dirichlet boundary condition is used on the rest of the boundary. The convergence rates are shown in Table B.1b and Figure B.2b. Again, the correct convergence rate $(p + 1)^*$ is observed.

\textsuperscript{188} Bezanson et al., Julia: A Fresh Approach to Numerical Computing, 2017.

\textsuperscript{*}(p + 1) because we show the $L_2$ solution error.
For 3D problems, the \((p + 1)\) convergence rate is also observed, as is demonstrated in the following. For a 3D problem on a unit cube with only homogeneous Dirichlet boundary conditions, the exact solution used here is

\[
\begin{align*}
    & u_1 = 10 \left( x_2 - x_2^3 \right) \left( x_3 - x_3^3 \right) \sin(\pi x_1) \left( 1 - x_1 \right) \left( 1 - \frac{1}{3} x_2 \right) \left( 1 - \frac{1}{3} x_3 \right) \\
    & u_2 = 2 \left( x_1 - x_1^3 \right) \left( x_3 - x_3^3 \right) \sin(\pi x_2) \left( 1 - x_2 \right) \left( 1 - \frac{1}{3} x_1 \right) \left( 1 - \frac{1}{3} x_3 \right) \\
    & u_3 = 5 \left( x_1 - x_1^3 \right) \left( x_2 - x_2^3 \right) \sin(\pi x_3) \left( 1 - x_3 \right) \left( 1 - \frac{1}{3} x_1 \right) \left( 1 - \frac{1}{3} x_2 \right).
\end{align*}
\]

Table B.2a and Figure B.3a show the convergence rates for this problem, where we clearly observe the \((p + 1)\) convergence rates.

<table>
<thead>
<tr>
<th>(p = 1)</th>
<th>(p = 2)</th>
<th>(p = 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(u_1)</td>
<td>1.98</td>
<td>3.00</td>
</tr>
<tr>
<td>(u_2)</td>
<td>2.00</td>
<td>2.98</td>
</tr>
<tr>
<td>(u_3)</td>
<td>1.98</td>
<td>2.99</td>
</tr>
<tr>
<td>(\sigma_{11})</td>
<td>1.94</td>
<td>2.96</td>
</tr>
<tr>
<td>(\sigma_{12})</td>
<td>1.94</td>
<td>2.90</td>
</tr>
<tr>
<td>(\sigma_{13})</td>
<td>1.94</td>
<td>2.96</td>
</tr>
<tr>
<td>(\sigma_{22})</td>
<td>1.93</td>
<td>2.95</td>
</tr>
<tr>
<td>(\sigma_{23})</td>
<td>1.88</td>
<td>2.91</td>
</tr>
<tr>
<td>(\sigma_{33})</td>
<td>1.92</td>
<td>2.91</td>
</tr>
</tbody>
</table>

(a) Only Dirichlet boundary conditions

\[
\begin{align*}
\text{(a) Only Dirichlet boundary conditions} & & \text{(b) Dirichlet and Neumann boundary conditions} \\
\end{align*}
\]

Figure B.2: \(L_2\) solution error convergence for linear elasticity problem in 2D.

Table B.2: Rates of convergence of \(L_2\) solution error for different discretization orders in 3D.
Finally, for a 3D problem with both Dirichlet and Neumann boundary conditions, we consider the following exact solution:

\[
\begin{align*}
    u_1 &= 10 \left( x_2 - x_2^2 \right) \left( x_3 - x_3^2 \right) \sin(\pi x_1) \left( 1 - x_1 \right) \left( 1 - \frac{1}{3} x_2 \right) \left( 1 - \frac{1}{4} x_3 \right), \\
    u_2 &= 2 \left( x_1 - x_1^2 \right) \left( x_3 - x_3^2 \right) \sin(\pi x_2) \left( 1 - x_2 \right) \left( 1 - \frac{1}{3} x_1 \right) \left( 1 - \frac{1}{4} x_3 \right), \\
    u_3 &= 5 \left( x_1 - x_1^2 \right) \left( x_2 - x_2^2 \right) \cos(\pi x_3) \left( 1 - \frac{1}{3} x_3 \right) \left( 1 - \frac{1}{3} x_1 \right) \left( 1 - \frac{1}{2} x_2 \right),
\end{align*}
\]

where an inhomogeneous Neumann boundary condition is used at \( x_3 = 0 \) and \( x_3 = 1 \) and a homogeneous Dirichlet boundary condition on the rest of the boundary. The convergence rates for this problem are shown in Table B.2b and Figure B.3b, where again the \((p+1)\) convergence rate is observed.

![Figure B.3: \(L_2\) solution error convergence for linear elasticity problem in 3D.](image)
GEOMETRY GENERATION FOR LATTICE STRUCTURES

After the lattice topology is designed and the optimal cross-sectional area for each strut in the lattice is found—as explained in Chapter 4—the design of the lattice is essentially a list of nodes coordinates and a list of cross-sectional areas of struts connecting those nodes. In order to manufacture a part, this information has to be translated into a solid representation of the part. This appendix describes how we generate such solid representations, and how such a geometry can be additively manufactured. The solid representation of a lattice is discussed in Appendix C.1, while the generation of a CAD geometry is described in Appendix C.2. Building such a structure may require a considerable computational effort for large structures and the resulting CAD files can become quite large. The resulting model then needs to be sliced in order to print. However, we could also directly slice the lattice based on the coordinates of the nodes and the cross-sectional areas of each strut, which is discussed in Appendix C.3.

C.1 SOLID REPRESENTATION OF LATTICE STRUCTURES

Throughout the design of the lattice topology and the cross-sectional area optimization, we use the lattice representation of the physical part. This representation consists of the coordinates of the nodes of the lattices, the connectivity of the struts and the nodes, and the cross-sectional area of each strut—see Figure C.1. Such a representation is a very cheap representation of a complicated model. However, such a lattice representation is not manufacturable. In order to manufacture the part, a solid part needs to be generated which can then be used to print the lattice. This section explains how to generate a solid geometry for a lattice structure.

Drawing the lattice structure “by hand” based on the lattice representation in a commercial CAD package might sound like an attractive option, because it uses the same infrastructure as is used in conventional manufacturing. Unfortunately, current commercial CAD packages are inadequate for such a task, because it takes a long time to draw a lattice and results in massive data files.189,190 For example, drawing a $10 \times 10 \times 10$ octet truss in CATIA can take almost half an hour and generates CAD files on the order of several hundred Mbs.191

Generating the geometry efficiently—and automatically from the
lattice representation—for parts designed for additive manufacturing is therefore of great interest to researchers and practitioners. The requirements for a geometry generation algorithm for additive manufacturing are that (1) the resulting solid representation is generated efficiently, (2) the resulting geometry is watertight, and (3) the algorithm is able to handle anisotropic lattices.

For the solid representation of lattice structures, *Boolean operations* are often used. In this approach, the struts are modeled using cylinders and the nodes using spheres and these are then joined together using Boolean union operations. Such an approach, however, requires a massive amount of computational power to perform so many Boolean operations. Boolean union operations are also not robust; they can fail at any point in the structure due to tolerancing issues.

Wang et al.\textsuperscript{192} therefore investigated using a hybrid approach: connecting the struts and nodes together node by node, tessellating nodes with half-length struts, and then connect the tessellations. In this approach, no parametric \textsc{cad} file is generated, but instead the whole part is already tessellated. This approach suffers from less problems with tolerancing, but still relies heavily on slow Boolean operations.

Srinivasan et al.\textsuperscript{193} instead fully tessellate a lattice structure by tessellating the struts and then connect the nodes using convex hulls, resulting in a watertight structure. This approach is demonstrated in Figure C.2. This approach is for instance used in the \textit{IntraLattice} software.\textsuperscript{164} However, when the lattices have high anisotropy, the resulting structure has very thick nodes—essentially blobs of material—because the node radius that is used is that radius for which none of the struts overlap at a single node.

Recently, Massarwi et al.\textsuperscript{194} demonstrated a parametric \textsc{cad} representation for microstructures which uses trivariate deformation functions to map between parameter and object space. For lattice structures this can be used to map a unit cell over a whole domain in a water-tight fashion. This approach is therefore purely focused on structured geometries.

A different approach is taken by Rothenberg\textsuperscript{195} which essentially

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{latticerepresentation.png}
\caption{Lattice representation with 6 nodes and 13 struts.}
\end{figure}

\textsuperscript{192} Wang et al., *A Hybrid Geometric Modeling Method for Large Scale Conformal Cellular Structures*. 2005

\textsuperscript{193} Srinivasan et al., *Solidifying Wireframes*. 2005

\textsuperscript{164} Tang et al., *Bidirectional Evolutionary Structural Optimization (BESO) Based Design Method for Lattice Structure to be Fabricated by Additive Manufacturing*. 2015

\textsuperscript{194} Massarwi et al., *Hierarchical, Random and Bifurcation Tiling with Heterogeneity in Micro-Structures Construction via Functional Composition*. 2018

\textsuperscript{195} Rothenberg, *Dynamic Cellular Microstructure Construction*. 2016
defines the structure implicitly through a level-set and redefines union and subtraction operations such that they can be executed quickly on the implicit surface and without the previously-mentioned robustness issues. This approach is for instance used by Mici et al. to embed cooling channels into 3D-printer parts. Because the surface is defined implicitly, all operations on the computational geometry of the structure have to be redefined, such as visualization, meshing, and solid model generation for manufacturing. It was therefore deemed impractical for this research, but this approach is promising for future work.

In the following, we explain our algorithm for computing the solid geometry for lattices. McMillan et al. explored similar ideas, but their algorithm is restricted to using period unit cells, whereas we specifically require an algorithm for anisotropic lattices. Appendix C.1.1 explains the geometry for the nodes and Appendix C.1.2 explains the geometry generation for the struts.

### C.1.1 NODE GEOMETRY

Again, in order to generate the solid representation, we start from the lattice representation (Figure C.1), which is the cheapest representation of a lattice structure. The lattice representation holds information about the coordinates of each node, the cross-sectional area of each strut, and connectivity information, as is shown in the following data structure in the Julia language.

```julia
struct Lattice
    p::Vector{SVector{Float,3}} # Node coordinates
    a::Vector{Float} # Cross-sectional area of each strut
    s::Vector{SVector{Int,2}} # Strut to node connectivity
    n2s::Vector{Vector{Int}} # Node to strut connectivity
    bound::Vector{Vector{Int}} # List of edges on boundaries
end
```

To translate that information into a solid representation, we go through several steps. The first step is to find the maximum cross-sectional area associated with a node, and to find that radial distance from the node for which the struts do not collide.
The second step is to—for each strut separately—find the cutting planes between the struts. For instance, the normal vector of the cutting plane between strut $i$ and $j$ is found as

$$n_{ij} = \frac{n_i - n_j}{\|n_i - n_j\|}.$$ 

As an example, the cutting planes for node 1 of Figure C.1—with $i = 1$—are shown in Figure C.3.

The next step is then to find the intersecting vertices of any pair of cutting planes with the surface of the current strut, as is shown in Figure C.3. There are two intersecting vertices per pair of cutting planes.

Figure C.3: Cutting planes and intersecting vertices for strut 1 on node 1 of Figure C.1.

Figure C.4: Active vertices for intersections of strut 1.
The question now is which of these intersecting vertices are actually active. A vertex is “active” when that vertex lies on the surface of the node. This is important, because vertices from the strut have to be projected onto a line segment between active vertices to guarantee watertightness. To find out whether a vertex is active, we project that node onto each cutting plane and if the projected vertex is the same as the intersecting vertex and that vertex is the closest to the strut, the vertex is active. The active vertices of node 1 as seen by strut 1 for the example in Figure C.3 are shown in Figure C.4.

![Figure C.5: Points on the strut are projected onto the node using the active vertices information.](image)

The final step is to project the strut onto the node using the active segments (segments between active vertices), as is shown in Figure C.5. The projected vertices can be used to tessellate the surface of the lattice, or to build splines for the parametric CAD definition.

Note that when the struts do not have the same cross-sectional area everywhere, we use the largest strut diameter for each of the connecting

![Figure C.6: The struts are tapered from the node to the correct cross-sectional area.](image)
struts. Therefore, the struts taper from the maximum cross-sectional area to the correct cross-sectional area along the strut (Figure C.6). This makes finding the intersections between struts near the node more straightforward. Furthermore, this helps with load introduction into a node with several struts of different diameter.

Often the lattice structure interfaces with a different part and requires a side of the lattice to be completely flat. This is also important for additive manufacturing as ideally a build is started from a flat base. A flat base can be included in the geometry generation process, by adding the flat plane to the other intersecting planes (see Figure C.3) and detecting when a strut is aligned with that flat plane. As an example, in Figure C.7 we consider node 1 from Figure C.1 where the flat plane is \( z = 0 \).

![Figure C.7: A flat base is useful for interfaces with other parts or to start a print from a flat surface.](image)

### C.1.2 Strut Geometry

Once the node geometry is generated, the solid strut geometry is needed to connect them. Care has to be taken to connect all these nodes correctly in a watertight fashion, especially when the nodes are rotated with respect to one another (Figure C.8) or have a different number of active vertices per strut.

![Figure C.8: Struts connect the nodes, but care has to be taken such that the vertices are connected correctly in a watertight fashion.](image)
C.2 CAD GENERATION

The geometry has been fully parametrized in Appendix C.1, but that information needs to be stored in a CAD format to be readable by AM machines. The nodes and struts have been parametrized by the active vertices and the edges connecting those vertices, and now we generate a solid representation from the surface of that lattice.

C.2.1 TRIANGULATED REPRESENTATION

The STL file format is still the most popular file format for both model definition and the slices definition used for manufacturing.\textsuperscript{198} Using this format, a surface is discretized in triangles of which the vertices and connectivity is stored in the STL file. The surface therefore has to be triangulated.

The struts are tessellated using the points generated along the active segments, as shown in Figure C.5, resulting in a geometry like Figure C.9. Examples of lattices generated using this approach are shown in Figure C.10, where lattices with hundreds of struts are generated in less than 1 second.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{struts_tessellation.png}
\caption{Tessellation of struts.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{lattices.png}
\caption{Examples of lattices generated using the geometry generation algorithm.}
\end{figure}

C.2.2 PARAMETRIC CAD REPRESENTATION

The STL representation of the node is useful for visualization and for additive manufacturing. However, if the lattice structure is to interface

\textsuperscript{198} Jacobs, Rapid Prototyping & Manufacturing: Fundamentals of Stereolithography. 1992
with other parts or manufacturing features—such as tapped holes—need to be added, a parametric CAD representation is more appropriate.

Here, we generate a parametric CAD representation of a lattice structure by generating edges along the active segments using B-splines and then generating faces along the strut. All faces are then sewed together to yield the final CAD part. This is a more robust and faster approach than using Boolean union operations. Throughout this work, we use egads\(^{199}\) which is based on the openCASCADE kernel\(^{200}\). An example of a lattice structure which has been modified in CATIA to add tapped holes is shown in Figure C.11.

### Figure C.11: A parametric CAD representation allows for taking advantage of manufacturing-based CAD software to apply, for example, threaded holes to a lattice structure.

#### C.3 SLICING OF LATTICE STRUCTURES FOR PRINTING

Additive manufacturing processes build up parts in a layer-wise fashion, requiring the CAD representation of those parts to be split into thousands of different slices along the build direction. In particular for lattice structures, this slicing procedure can become quite expensive as there are often thousands of struts in a lattice, resulting in large CAD file sizes. Instead, we can also directly slice the lattice frame representation, which is a cheap representation of the lattice. This can be achieved using the lattice frame representation and computing the active vertices for each lattice node—which are also needed to compute the solid representation of the lattice.

For a given slicing plane, we first find which struts are intersected by the plane by looping over all struts. Then we determine whether the lattice nodes are intersected by the slicing plane by determining whether all active vertices from that node are on the same side of the slicing plane. If they are not, the node is intersected. The slice through the node is

\(^{199}\) Haimes et al., *On the Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design*, 2012

\(^{200}\) OpenCASCADE, *OpenCASCADE Technology, 3D Modeling & Numerical Simulation*, 2017
computed by computing the intersections with the edges that connect the active vertices together.

An example of slicing the lattice frame representation directly is shown in Figure C.12 for a cube lattice structure.

Figure C.12: Slicing the lattice frame representation directly with seven slicing planes for a cube lattice structure.
ADJOINT FOR LOW-ORDER STRUCTURAL MODEL OF LATTICE STRUCTURES

The approach in Chapter 5 to aeroelastically tailor a lattice structure results in a high-dimensional optimization problem, for which efficient gradient computation is critical. This appendix details the gradient computation for that high-dimensional optimization problem, focusing on computing the gradient of the maximum aeroelastic eigenvalue with respect to the cross-sectional area of the struts of the lattice. To do this efficiently an adjoint for the low-order structural model in Section 5.1.2 is derived and implemented. To derive this adjoint, adjoints for different elementary matrix operations need to be known, which are discussed in Appendix D.1. The approach to obtain the adjoint for the low-order structural model builds on these results, as discussed in Appendix D.2.

D.1 ADJOINTS FOR ELEMENTARY MATRIX OPERATIONS

The results in this appendix are based on the work of Giles,166,167 which in turn is mostly based on the seminal work by Dwyer and Macphail.201 We follow the notation of Giles closely here, for a more detailed explanation please consult Ref. 166 and 167.

To explain the process of reverse differentiation or adjoint differentiation, we consider a function which starts with some scalar input $s_I$ and outputs some scalar $s_O$. Consider a matrix $A$ which is used in some intermediate step in this function. Following standard Automatic Differentiation (AD) terminology, the derivative of each entry in $A$ with respect to the input $s_I$ is the matrix $\dot{A}$, or

$$\dot{A}_{i,j} = \frac{\partial A_{i,j}}{\partial s_I},$$

where $\dot{A}$ has the same size as $A$. On the other hand, the derivative of $s_O$ with respect to each entry in $A$ is denoted as $\overline{A}$, or

$$\overline{A}_{i,j} = \frac{\partial s_O}{\partial A_{i,j}},$$

where $\overline{A}$ again has the same dimensions as $A$. $\overline{A}$ is usually referred to as the adjoint of $A$.

As an example, consider an intermediate step in a function,

$$C = F(A, B).$$
An infinitesimal perturbation to $A$ and $B$ results in

$$dC = \frac{\partial F}{\partial A} dA + \frac{\partial F}{\partial B} dB.$$  \hfill (D.1)

Assuming the perturbation is the result of changes to the input variable $s_i$, we obtain

$$\dot{C} = \frac{\partial F}{\partial A} \dot{A} + \frac{\partial F}{\partial B} \dot{B}.$$  

Instead, if the derivative of $s_O$ with respect to $A$ and $B$ is to be computed, the differentiation process starts in reverse, i.e., starting at the end of the function and working back. By definition,

$$ds_O = \sum_{i,j} C_{i,j} dC_{i,j} = \text{Tr} \left( \overline{C}^T dC \right).$$  \hfill (D.2)

Substituting Eq. (D.1) in Eq. (D.2) yields,

$$ds_O = \text{Tr} \left( \overline{C}^T \frac{\partial F}{\partial A} dA \right) + \text{Tr} \left( \overline{C}^T \frac{\partial F}{\partial B} dB \right).$$

If we assume that $A$ and $B$ are not used in any other intermediate steps in the function $F$, then

$$\overline{A} = \left( \frac{\partial F}{\partial A} \right)^T \overline{C}, \quad \overline{B} = \left( \frac{\partial F}{\partial B} \right)^T \overline{C}.$$  

This approach to finding $\overline{A}$ and $\overline{B}$ is similar to the work by Magnus and Neudecker.\textsuperscript{202}

In deriving $\overline{A}$ and $\overline{B}$ for different expressions of $F$, the following identities prove useful:

$$\text{Tr} (A^T) = \text{Tr} (A)$$
$$\text{Tr} (A + B) = \text{Tr} (A) + \text{Tr} (B)$$
$$\text{Tr} (ABC) = \text{Tr} (CAB) = \text{Tr} (BCA)$$

As an example for deriving the adjoint for different forms of $F$ we show the adjoint differentiation for the matrix inverse—taken directly from Ref. 166—and restate the adjoint for an eigenvalue problem—taken from Ref. 167. We then derive the adjoint for a matrix triple product and a block inverse, which are required for deriving an adjoint for the structural low-order model.

**INVERSE**

Following Giles,\textsuperscript{166} if $C = A^{-1}$, then $CA = I$ and hence

$$dC A + Cd A = 0 \quad \rightarrow \quad dC = -C dA C.$$  

Thus

$$\text{Tr} (\overline{C}^T dC) = \text{Tr} (-\overline{C}^T A^{-1} dA A^{-1}) = \text{Tr} \left( -A^{-1} \overline{C}^T A^{-1} dA \right),$$

from which we obtain the adjoint

$$\overline{A} = -A^{-T} \overline{C} A^{-T} = -\overline{C}^T \overline{C}^T.$$  

\textsuperscript{202} Magnus et al., Matrix Differential Calculus with Applications in Statistics and Econometrics. 1999

\textsuperscript{166} Giles, Collected Matrix Derivative Results for Forward and Reverse Mode Algorithmic Differentiation. 2008

\textsuperscript{167} Giles, An Extended Collection of Matrix Derivative Results for Forward and Reverse Mode Automatic Differentiation. 2008
Consider a square matrix $A$ with distinct eigenvalues $\lambda_k$ and eigenvectors $r_k$. We define $\Lambda$ to be a diagonal matrix of the eigenvalues $\lambda_k$ and $R$ a matrix where the columns correspond to the eigenvectors $r_k$. The adjoint for the eigenvalues and eigenvectors of $A$ is found to be

$$
\bar{A} = R^{-\top} (\bar{\Lambda} + F \odot [R^\top \bar{R}]) R^\top,
$$

where $F_{i,j} = (\lambda_i - \lambda_j)^{-1}$ and “$\odot$” indicates the Hadamard product for two matrices of the same size, which is an element-wise product between two matrices, i.e., $(A \odot B)_{i,j} = A_{i,j} \cdot B_{i,j}$. For the full derivation of this result please consult Ref. 167.

Note that in this work, we are only interested in the value of the largest aeroelastic eigenvalue. In that case then, the adjoint becomes

$$
\bar{A} = R^{-\top} \bar{\Lambda} R^\top, \quad \tag{D.3}
$$

where

$$
\bar{\Lambda} = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0
\end{bmatrix},
$$

if the eigenvalues are ordered such that the first eigenvalue of $A$ has the largest positive real part.

**Triple Matrix Inverse Product**

By employing the Schur complement of Eq. (5.3) in Eq. (5.4), products of the form

$$
D = CA^{-1}B \tag{D.4}
$$

have to be computed. For this product, we can write

$$
dD = dC A^{-1}B + C d (A^{-1}) B + CA^{-1} dB = dC A^{-1}B - CA^{-1} dA A^{-1}B + CA^{-1} dB.
$$

For the adjoint differentiation—using Eq. (D.2)—then,

$$
\text{Tr} (\bar{D}^\top dD) = \text{Tr} \left( \bar{D}^\top dC A^{-1}B \right) - \text{Tr} \left( \bar{D}^\top CA^{-1} dA A^{-1}B \right) + \text{Tr} \left( \bar{D}^\top CA^{-1} dB \right) \\
= \text{Tr} (A^{-1}B \bar{D}^\top dC) - \text{Tr} (A^{-1}B \bar{D}^\top CA^{-1} dA) + \text{Tr} (\bar{D}^\top CA^{-1} dB).
$$

Therefore, the adjoint of Eq. (D.4) is

$$
\bar{A} = -A^{-\top} C^\top \bar{D} B^\top A^{-\top}, \quad \tag{D.5}
$$

$$
\bar{B} = A^{-\top} C^\top \bar{D}
$$

$$
\bar{C} = \bar{D} B^\top A^{-\top}.
$$
The inverse of $2 \times 2$ block matrix can be expressed as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix} \tag{D.6}$$

where

$$E = (A - BD^{-1}C)^{-1}$$

$$F = -(A - BD^{-1}C)^{-1}BD^{-1}$$

$$G = -(D - CA^{-1}B)^{-1}CA^{-1}$$

$$H = (D - CA^{-1}B)^{-1}.$$  

In the following we derive the adjoint for $E$, and simply state the result for the other submatrices. The results are combined in the end to yield the complete adjoint of Eq. (D.6).

For $E$, we can write an infinitesimal perturbation as

$$dE = d\left(A - BD^{-1}C\right)^{-1} = -E d\left(A - BD^{-1}C\right)E$$

$$= -E dA E + E dB D^{-1}CE\,$$

$$- EBD^{-1}dD D^{-1}CE + EBD^{-1}dCE$$

For the adjoint then,

$$\text{Tr}\left(\bar{E}^T dE\right) = \text{Tr}\left(-E \bar{E}^T E dA\right) + \text{Tr}\left(D^{-1}CE \bar{E}^T E dB\right)$$

$$+ \text{Tr}\left(E \bar{E}^T EBD^{-1} dC\right) + \text{Tr}\left(-D^{-1}CE \bar{E}^T EBD^{-1} dD\right).$$

Therefore, we can write the adjoint as,

$$\bar{A}_E = -E^T \bar{E} E^T$$

$$\bar{B}_E = -\bar{A}_E C^T D^{-T}$$

$$\bar{C}_E = -D^{-T} B^T \bar{A}_E$$

$$\bar{D}_E = -\bar{C}_E C^T D^{-T}.$$  

Similarly, for $F$

$$\bar{A}_F = E^T \bar{F} D^{-T} B^T E^T$$

$$\bar{B}_F = -\bar{A}_F C^T D^{-T} - E^T \bar{F} D^{-T}$$

$$\bar{C}_F = -D^{-T} B^T \bar{A}_F$$

$$\bar{D}_F = -\bar{C}_F C^T D^{-T} + D^{-T} B^T E^T \bar{F} D^{-T}.$$  

For $G$

$$\bar{D}_G = H^T \bar{G} A^{-T} C^T H^T$$

$$\bar{C}_G = -\bar{D}_G B^T A^{-T} - H^T \bar{G} A^{-T}$$

$$\bar{B}_G = -A^{-T} C^T \bar{D}_G$$

$$\bar{A}_G = -\bar{B}_G B^T A^{-T} + A^{-T} C^T H^T \bar{G} A^{-T}.$$
Finally, for $\bar{H}$

\[
\begin{align*}
\bar{D}_{\bar{H}} &= -H^{\top} HH^{\top} \\
\bar{C}_{\bar{H}} &= -\bar{D}_{\bar{H}} B^{\top} A^{-\top} \\
\bar{B}_{\bar{H}} &= -A^{-\top} C^{\top} \bar{D}_{\bar{H}} \\
\bar{A}_{\bar{H}} &= -\bar{B}_{\bar{H}} B^{\top} A^{-\top}.
\end{align*}
\]

The complete adjoint for the inverse of a 2×2 block matrix (D.6) combines the previous results as

\[
\begin{align*}
\bar{A} &= \bar{A}_{E} + \bar{A}_{F} + \bar{A}_{G} + \bar{A}_{H} \\
\bar{B} &= \bar{B}_{E} + \bar{B}_{F} + \bar{B}_{G} + \bar{B}_{H} \\
\bar{C} &= \bar{C}_{E} + \bar{C}_{F} + \bar{C}_{G} + \bar{C}_{H} \\
\bar{D} &= \bar{D}_{E} + \bar{D}_{F} + \bar{D}_{G} + \bar{D}_{H}.
\end{align*}
\]

### D.2 Approach for Structural Model

The optimization problem for aeroelastically tailoring a lattice structure is described in Eq. (5.6). Apart from the flutter constraint, the derivatives for the objective function and constraints are obtained in a straightforward manner, as the object and these constraints are either linear or quadratic. To obtain the gradient of the maximum aeroelastic eigenvalue $\chi_{\text{fl}}$, we compute the adjoint of $\chi_{\text{fl}}$. Trying to compute the gradient using finite differences or tangent differentiation is too expensive as that would require building up the low-order model and computing the eigenvalues as many times as the number of struts. Instead if we use an adjoint, we only have to compute the adjoint once to find the gradient of $\chi_{\text{fl}}$ with respect to the cross-sectional areas of the struts.

\[
\begin{align*}
\min_{u, a, f, \sigma} \quad & \mathcal{V} = l^{\top} a \\
\text{subject to} \quad & Cf = F \quad \text{(force balance)} \\
& B\sigma + C^{\top} u = 0 \quad \text{(stress-strain compatibility)} \\
& f_{j} = \sigma_{j} a_{j} \quad \text{(stress definition)} \\
& -\sigma_{C} \leq \sigma_{j} \leq \sigma_{T} \quad \text{(stress limits)} \\
& f_{j} \geq -\frac{\pi E a_{j}^{2}}{4(k_{e} l_{j})^{2}} \quad \text{(buckling)} \\
& \chi_{\text{fl}} < \chi_{\text{fl,max}} \quad \text{(flutter)} \\
& a_{\min,j} \leq a_{j} \leq a_{\max,j} \\
& u \in \mathbb{R}^{n_{d}}, \quad f \in \mathbb{R}^{m}.
\end{align*}
\]

To compute the adjoint of $\chi_{\text{fl}}$, the low-order model construction has to be traversed in reverse. The first step is therefore to compute
the adjoint of the low-order mass matrix and stiffness matrix—using Eq. (D.3)—from which the adjoints of \( \hat{M}, \hat{S}_y, \) and \( \hat{I}_y \) are found.

The part of the adjoint of the cross-sectional areas that stems from the mass matrix \( M \) follows from the adjoint of the low-order mass matrices and the mapping between the low-order mass matrix and the full mass matrix.

For the stiffness matrices, we require the adjoint of a block inverse to compute the adjoints of the blocks in Eq. (5.4), using the results in Appendix D.1. We would then like to compute the adjoint of \( K \) using Eq. (D.5), which requires the inverse of \( K \). However, \( K \) is a large sparse matrix, the inverse of which should never be computed directly as that would be too expensive. To avoid this, we can reorder the adjoint equations.

Focusing here on the upper left block of Eq. (5.4) which is here denoted as \( K_{11} \)—such that \( K_{11} = T_{u-h}K^{-1}T_{L-F} \)—the adjoint of \( K \) can be expressed as

\[
\frac{\partial \chi}{\partial K} = -K^{-\top}T_{u-h}^{-\top} \frac{\partial \chi}{\partial K_{11}} T_{L-F}^{-\top} K^{-\top}
\]

where we used the result in Eq. (D.5). We can then group each \( K^{-\top} \) with a projection matrix to reduce the computational cost:

\[
\frac{\partial \chi}{\partial K} = \left( K^{-\top}T_{u-h}^{-\top} \right) \frac{\partial \chi}{\partial K_{11}} \left( K^{-1}T_{L-F}^{-\top} \right)^{\top}.
\]

In this expression, each of those projection matrices is oriented such that it is a tall slender matrix. This approach is therefore much cheaper than computing the inverse of \( K \) directly, as only few matrix inverse vector products have to be computed. The part of the adjoint of the cross-sectional areas that stems from the mass matrix \( K \) then follows from the adjoint of \( B^{-1} \)—defined in Chapter 4—in a straightforward manner as each diagonal entry of \( B^{-1} \) is linearly dependent on the cross-sectional area of one strut.

The contributions of the mass and stiffness matrix on the adjoint of the cross-sectional areas of the struts are added to yield the gradient of the maximum eigenvalue with respect to the cross-sectional areas of the lattice struts. In our implementation, computing the adjoint is only 3.5\times more expensive than evaluating the maximum aeroelastic eigenvalue, starting from low-order model construction.

\footnote{Note that this expression considers only the contribute of \( K_{11} \) to the adjoint, and therefore assumes that \( \frac{\partial \chi}{\partial K_{12}}, \frac{\partial \chi}{\partial K_{21}}, \) and \( \frac{\partial \chi}{\partial K_{22}} \) are all zero.}
CODE LISTINGS

This chapter lists the tools that have been developed for this thesis and have been made publicly available.

RESEARCH

**luteos.jl**

Julia implementation of a higher-order HDG method for 2D and 3D geometries, focused on linear elasticity.

_This code is used to compute the initial stress tensor throughout the solid domain, which then informs the metric on which the lattice is based._

[github.com/mopg/luteos.jl](http://github.com/mopg/luteos.jl)

**intrico.jl**

Julia implementation of the geometry generation algorithm, discussed in detail in Appendix C.

_This code is used to generate CAD geometries for all lattice structures designed in this thesis._

[github.com/mopg/intrico.jl](http://github.com/mopg/intrico.jl)

**egads.jl**

Julia package that wraps egads, a library that links to OpenCASCADE. This code is used for the parametric CAD representation of lattice structures in Appendix C.2.2.

[github.com/mopg/egads.jl](http://github.com/mopg/egads.jl)

GRAPHICS

**Tikz.jl**

Julia package to output \LaTeX\ Tikz code in .tex format.

_This code is used for both line plots or drawings in this thesis._

[github.com/mopg/Tikz.jl](http://github.com/mopg/Tikz.jl)
INDEX

Adjoint, 103, 163
Benchmark Supercritical Wing, 65
Bernoulli-Euler beam theory, 60
Bisection method, 53, 145, 146
Boolean CAD operations, 154
Circulation strength, 39, 145
D8.x aircraft, 25, 55, 72, 118
  D8.2, 72, 73
  D8.0, 72, 73, 75, 76
Direct Energy Deposition, 31
Discontinuous Galerkin, 147
  Embedded, 148
  Hybridizable, 147–150
Doublet-lattice method, 38, 57
Dynamic mode decomposition, 44
Electron Beam Melting, 31
Flow simulation
  Euler, 29, 42, 48, 64, 97, 111, 112
  RANS, 42, 112
  URANS, 114
Flutter boundary, influence of
  Airfoil thickness, 55
  Aspect ratio, 69
  Baseline lift coefficient, 54
  Engine mass, 71
  Sweep angle, 70
  Taper ratio, 68
Flutter validation
  Benchmark Supercritical Wing, 65
  Isogai Case A, 53
Fused Filament Fabrication, 31
Helmholtz decomposition, 38
Kutta-Joukowsky theorem, 39
Lattice optimization, 87
  with flutter constraints, 102, 167
Linear elasticity equations, 82, 147
Powder-based additive manufacturing process, 31
Quaternions, 85
STL file, 159
Selective Laser Melting, 31
Selective Laser Sintering, 31
Stereolithography, 31
Strip theory, 57
SU2, 45, 97
TASOFT, 66
Theodorsen theory, 28, 39, 41, 51, 59
Topology optimization, 30, 81
Truss-Braced Wing, 25, 27, 118
Typical section, 42
Volume-source density, 38
Vortex-lattice method, 38, 40, 57
Vorticity, 38
Wire-feed additive manufacturing process, 31
x-doublet strength, 39, 145
COLOPHON

This thesis has been typeset using \LaTeX, and compiled using pdflatex. Minion Pro is used as both the text and display typeface. The sans-serif text uses the Myriad Pro typeface, whereas monospaced text is typeset in Bitstream Vera Mono. Line figures and illustrations were predominantly generated using the Tikz \LaTeX package, \textit{FEM} data plots using Tecplot, and renders using Blender.

Several books have influenced the style, typography, and graphic design of this thesis, in particular:

Jean-Luc Doumont's \textit{Trees, Maps, and Theorems} (2009)
Robert Bringhurst's \textit{Elements of Typography} (1992)

\textit{Transonic Flutter Prediction and Aeroelastic Tailoring for Next-Generation Transport Aircraft}
Max Opgenoord, September 2018

© Massachusetts Institute of Technology 2018. All rights reserved.
Ph.D. Dissertation
Max M. J. Opgenoord

Transonic Flutter Prediction and Aeroelastic Tailoring for Next-Generation Transport Aircraft

~ MMXVIII ~