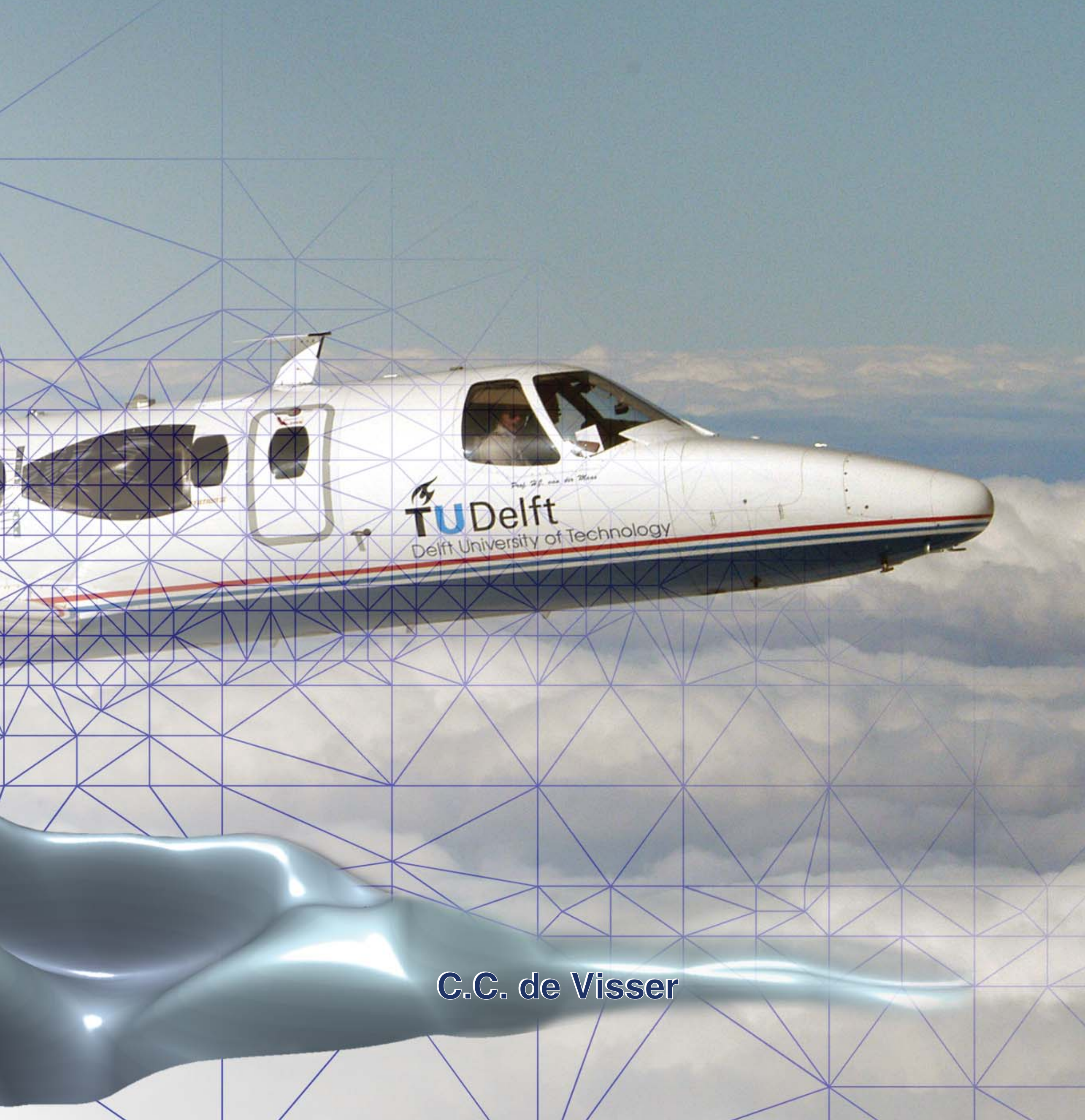
Global Nonlinear Model Identification with Multivariate Splines

Application to Aerodynamic Model Identification of the Cessna Citation II



C.C. de Visser

Global Nonlinear Model Identification
with
Multivariate Splines

Application to Aerodynamic Model Identification of the Cessna Citation II

C.C. de Visser

ISBN 978-90-8570-770-7

Published and distributed by C.C. de Visser.

Printed by Wöhrmann Print Service, Zuthpen, The Netherlands.

Cover design by R.M. Wessels.

Copyright © 2011 C.C. de Visser. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission in writing from the proprietor.

Global Nonlinear Model Identification
with
Multivariate Splines

Application to Aerodynamic Model Identification of the Cessna Citation II

PROEFSCHRIFT

Ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op maandag 4 juli 2011 om 12.30 uur

door

Cornelis Coenraad DE VISSER

ingenieur luchtvaart en ruimtevaart
geboren te Groningen.

Dit proefschrift is goedgekeurd door de promotor:

Prof.dr.ir. J.A. Mulder

Copromotor:

Dr. Q.P. Chu

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. J.A. Mulder,	Technische Universiteit Delft, promotor
Dr. Q.P. Chu,	Technische Universiteit Delft, copromotor
Prof.Dr.-Ing. F. Holzapfel,	Technische Universität München
Prof.Dr.-Ing. R. Luckner,	Technische Universität Berlin
Prof.dr.ir. B. Koren,	Centrum Wiskunde & Informatica Amsterdam
Prof.dr.ir. M.H.G. Verhaegen,	Technische Universiteit Delft
Prof.dr.ir. M. Mulder,	Technische Universiteit Delft
Prof.dr.ir. B.A.C. Ambrosius,	Technische Universiteit Delft, reservelid

Summary

Global Nonlinear Model Identification with Multivariate Splines

At present, model based control systems play an essential role in many aspects of modern society. Application areas of model based control systems range from food processing to medical imaging, and from process control in oil refineries to the flight control systems of modern aircraft. Central to a model based control system is a mathematical model of the physical system or process that is being controlled. The field of science concerned with the identification of models of physical systems is called system identification. In this thesis, a new methodology is proposed for the identification of models of nonlinear systems using multivariate simplex splines. This new methodology has the potential to increase the performance of any model based control system by improving the quality of system models.

Modeling systems with nonlinear dynamics is a challenging task, and currently only a handful of methods exist that are capable of creating sufficiently accurate models of such systems. The four most widely known of these methods are neural networks, kernel methods, polynomial blending methods, and tensor product spline methods. All these methods are able to produce models of an arbitrarily high approximation power on a global model scale. However, these methods currently suffer from a number of inherent shortcomings. Neural networks are essentially black-box models and use global basis functions, resulting in complex, nontransparent, and inefficient computational schemes for their training and evaluation. Kernel methods are non-parametric in nature, which means that in principle there are as many kernel functions as there are data points, leading to inefficient computational schemes for large datasets. Polynomial blending methods use fuzzy logic techniques to blend local polynomial models into a single global model. The tuning of the fuzzy blending operation is done based on expert knowledge, with the result that it is unlikely to ever become a fully automated technique. Polynomial spline methods have been successfully used in the past for the modeling of nonlinear systems. However, these spline methods employed multivariate tensor product B-splines which are limited to rectangular domains, and are incapable of fitting widely occurring scattered data.

The new methodology proposed in this thesis is based on multivariate simplex splines, which are a recent type of multivariate spline that have a number of important advantages over the above mentioned methods. Firstly, simplex splines have a local polynomial basis, which implies that only small subsets of parameters and basis polynomials need to be considered during estimation and evaluation, resulting in efficient computational schemes. Secondly, simplex spline models are parametric models, which allows for efficient approximation of very large datasets. Thirdly, the simplex splines are linear in the parameters, meaning that linear regression methods can be used for their estimation. Fourthly, the simplex splines are defined on non-rectangular domains and can be used to approximate scattered data. And finally, the quality of simplex spline based models can be assessed using a number of unique and powerful model quality assessment methods.

Multivariate simplex splines consist of polynomial basis functions, called B-form polynomials, which are defined on geometric structures called simplices. Every simplex supports a single B-form polynomial which itself consists of a linear combination of Bernstein basis polynomials. Each individual Bernstein basis polynomial is scaled by a single coefficient called a B-coefficient. The B-coefficients have a special property in the sense that they have a unique spatial location inside their supporting simplex. This spatial structure, also known as the B-net, provides a number of unique capabilities that add to the desirability of the simplex splines as a tool for data approximation. For example, the B-net simplifies local model modification by directly relating specific model regions to subsets of B-coefficients involved in shaping the model in those regions. This particular capability has the potential to play an important role in future adaptive model based control systems. In such a control system, an on-board simplex spline model can be locally adapted in real time to reflect changes in system dynamics.

The approximation power of the multivariate simplex splines can be increased by joining any number of simplices together into a geometric structure called a triangulation. Triangulations come in many shapes and sizes, ranging from configurations consisting of just two simplices to configurations containing millions of simplices. Triangulations can be optimized by locally increasing or decreasing the density of simplices to reflect local system complexity. In principle, the total number of simplices in a triangulation is bounded only by the available computational resources. This thesis shows, however, that there is an important practical limit to the size and resolution of a triangulation. This practical limit is the result of every simplex requiring a minimum data content which is determined by the degree and continuity order of the basis polynomials. It was shown in this thesis that this data coverage problem requires a new approach towards triangulation optimization, as methods in the existing literature do not consider per-simplex data coverage as an optimization parameter. The newly proposed method for triangulation optimization produces triangulations that are

specifically suited for use with simplex splines by ensuring that every individual simplex in a triangulation contains a minimum amount of data.

While multivariate simplex splines have been used in the past to model scattered nonlinear data in two and three dimensions, no methodology existed for their use inside a framework for system identification. The unique properties of the simplex splines, together with the above mentioned advantages over existing data approximators, makes them highly desirable for use within such a framework. It is the main objective of this thesis to present a new methodology for system identification based on multivariate simplex splines. This new methodology encompasses the three main aspects of system identification; model structure selection, parameter estimation, and model validation. The aspect of model structure selection for the multivariate simplex splines consists of two parts. The first part is the geometric model structure selection which consists of the selection of the spline model dimensions and the creation of a triangulation embedded in this set of dimensions. The second part is the determination of the polynomial model structure. For the aspect of parameter estimation, a new formulation of the standard linear regression model structure was developed. In this formulation, the B-form polynomials of the simplex splines form the regressors. Using the new regression model structure, a number of different parameter estimation techniques can be employed to estimate the B-coefficients of the B-form polynomials. This thesis introduces two such methods for parameter estimation. The first is a generalized least squares estimator, which enables the estimation of B-coefficients on simplices containing measurement noise of varying magnitudes. The second parameter estimator is a differentially constrained recursive least squares estimator which allows, in real-time, the reconfiguration of spline models using incoming observations. During the aspect of model validation, the quality of the estimated spline models is assessed using existing methods based on an analysis of model residuals and parameter variances. Additionally, a number of completely new quality assessment methods are enabled by the use of the B-form polynomials. For example, the variances of the B-coefficients can be pinpointed to specific locations within the model. This means that regions of high parameter variance can be isolated within the global model and subjected to further analysis. These unique and powerful properties together may result in a new perspective on system identification and parameter estimation, potentially leading to further innovations in the field.

This thesis introduces three major theoretical innovations in the field of multivariate spline theory. These innovations were essential in the creation of an effective method for system identification with simplex splines. The first of these innovations was the definition of the differential constraints, which are used to constrain the directional derivatives of the simplex splines at selected locations within the spline domain. The differential constraints enable bounded model extrapolation and limit polynomial divergence near the bounds of the spline domain. Additionally, the differential constraints can be applied to impose boundary conditions like Dirichlet or Cauchy conditions on the simplex spline functions, thereby

enabling the approximation of solutions to boundary value problems using simplex splines. The second innovation was the development of a theory for the quantification of B-net propagation, a new effect observed in large scale triangulations. B-net propagation is the spreading of local disturbances from the B-net of one simplex to that of its neighbors. It was proved that B-net propagation effectively transforms a simplex spline function from a local approximator into an global approximator if its continuity order is high with respect to its polynomial degree, and when it is defined on the most widely used triangulation type. A final innovation was a new formulation of the B-form in global Cartesian coordinates instead of local barycentric coordinates. The Bernstein basis polynomials of the simplex splines are functions in terms of local barycentric coordinates, which means that their global interpretation is meaningless. The new formulation of the B-form polynomials in global coordinates adds a global interpretation capability. Additionally, and more importantly, the new formulation enables the optimization of triangulation and B-coefficients in a single step, thereby avoiding the need for separate triangulation optimization.

Aircraft aerodynamics are notoriously nonlinear, and the identification of accurate aerodynamic models from flight data has historically been a challenging task. Aerodynamic models are crucial in the correct functioning of flight simulators and flight control systems. The higher the quality of an aerodynamic model, the more accurate its predictions on the aerodynamic forces and moments acting on an aircraft. For flight simulator applications, this directly translates into increased simulator fidelity, and consequently a better training environment for pilots. For flight control systems this results in a more accurate reference signal tracking performance, and an increased tolerance to damage events. Ultimately, high quality aerodynamic models have an important societal relevance by benefiting flight safety. The societal relevance of accurate aerodynamic models, together with the technical challenge of their identification from flight data, presents the ideal case for demonstrating the new methodology proposed in this thesis.

Two identification experiments in the field of aerodynamic model identification were conducted with the new methodology. The first experiment was the identification of an aerodynamic model for the F-16 fighter aircraft using a NASA wind tunnel dataset. The internal structure of this wind tunnel model was known, and as such it provided a controlled environment for testing and validating the new methodology. In the second identification experiment a complete set of aerodynamic models for the Cessna Citation II laboratory aircraft were identified using flight data obtained during seven test flights conducted between 2006 and 2010. In total, 247 flight test maneuvers were flown which together provided a significant coverage of the flight envelope of the Citation II. The complete identification dataset consisted of millions of measurements on more than sixty flight parameters. For this real-life experiment it was necessary to consider the aspects of model structure selection, parameter estimation, and model validation. The geometric model structure selection was performed using a novel approach based on the occurrence of hysteresis in the time trace of the aerodynamic force and moment coefficients. Using

the hysteresis analysis method, a number of candidate dimension sets was defined. For each candidate dimension set, a triangulation of the hypercube was created that minimally envelopes the flight test data. The polynomial model structure was selected by validating the performance of a number of prototype simplex spline functions of different degree and continuity order on the hypercube triangulation. More than 2000 prototype spline models were identified using a newly developed, highly optimized software implementation of the simplex spline identification algorithm. The final geometric and polynomial model structures were selected based on the further optimization of the best performing prototype model.

The identified simplex spline based aerodynamic models are phenomenological models, that is, models that are based directly on observational data. Using the developed methods for simplex spline model validation it is proved that the models are both accurate and of guaranteed numerical stability inside the spline domain. The identification and validation results of the simplex spline models were compared with those of ordinary polynomial models identified using standard identification methods. These results showed that the multivariate simplex spline based aerodynamic models were of significantly higher quality than the aerodynamic models based on ordinary polynomials.

The research performed in the framework of this thesis leads to three principal recommendations. First, it was found that the greatest practical limit in the application of multivariate simplex splines in real life data approximation is per-simplex data coverage. To alleviate this problem, and further improve the practical utility of the simplex splines, a software tool should be developed for checking, in real time, the coverage of the system operating domain with measurements. In the case of aerodynamic model identification, such a software tool would provide cues to the pilots for executing specific maneuvers. The second recommendation is that a general triangulation optimization method should be developed that is specifically suited for system identification with simplex splines. Such a method could be based on the global formulation of B-form polynomials provided in this thesis, and would close an important gap in current simplex spline theory. A final recommendation is a real-life implementation of an adaptive model based control system which employs the recursive B-coefficient estimator introduced in this thesis. For aerospace applications, this would result in a fault tolerant flight control system with a built in flight envelope prediction functionality. Installed in future aircraft, simplex spline adaptive model-based flight control systems could increase flight safety by turning catastrophic events into survivable incidents thereby saving human lives.

Contents

Summary vii

1 Introduction 1

1.1 Modeling Reality 1

1.2 Models, Identification, and Control 4

1.3 Introduction to Aerodynamic Model Identification 7

1.4 A Case for Multivariate Splines 10

1.5 Thesis Goals and Research Approach 12

1.5.1 Goals of this thesis 12

1.5.2 Research Approach 14

1.5.3 Scope and limitations 15

1.6 Contributions 15

1.7 Outline of the Thesis 16

2 Introduction to Multivariate Simplex Splines 19

2.1 Introduction to Multivariate Simplex Splines 20

2.1.1 Univariate B-Splines 21

2.1.2 The unclear path from univariate to multivariate splines 25

2.1.3 Tensor product splines 26
2.1.4 Thin plate splines 29
2.1.5 Polyhedral splines 31
2.1.6 Simplex splines 35
2.2 The Basis of the Multivariate Simplex Spline 39
2.2.1 The Simplex 39
2.2.2 Barycentric Coordinates 44
2.2.3 Bernstein basis polynomials 49
2.2.4 The B-form 52
2.2.5 A Vector form of the B-form 57
2.2.6 The B-coefficient net 58
2.2.7 Spline Spaces 60
2.3 Triangulations of Simplices 62
2.3.1 Simplex metrics 62
2.3.2 General definition 64
2.3.3 Type I/II triangulations 66
2.3.4 Delaunay triangulations 70
2.3.5 B-nets on triangulations 73
2.3.6 Simplex Stars 75
2.3.7 Geometric Triangulation optimization 76

3 Advances in Simplex Spline Theory 87
3.1 The de Casteljau algorithm 88
3.1.1 The ordinary de Casteljau algorithm 89
3.1.2 A multi-degree formulation of the de Casteljau recursion . . . 90
3.1.3 A one-step matrix form of the de Casteljau algorithm 92
3.2 Calculations with B-form Polynomials 96
3.2.1 Full triangulation vector form of B-form polynomials 96
3.2.2 Sums of B-form polynomials 98
3.2.3 Integrals of B-form polynomials 99

3.2.4 Inner products of B-form polynomials 99
3.2.5 Degree raising 99
3.2.6 Directional derivatives of B-form polynomials 101
3.2.7 Bounds on approximation with B-form polynomials 105
3.3 Smoothness Constraints 108
3.3.1 Definition of the continuity conditions 108
3.3.2 A general formulation for the continuity conditions 112
3.3.3 The effects of continuity 120
3.3.4 B-net propagation 124
3.3.5 The Smoothness Matrix 132
3.4 Differential Constraints 136
3.4.1 Definition of differential constraints 137
3.4.2 Differentially constraining polynomials on subsimplices 140
3.4.3 Differential constraints for bounded model extrapolation 141
3.5 A B-form in Global Coordinates 148
3.5.1 Barycentric coordinates revisited 149
3.5.2 A B-form in global coordinates 151
3.5.3 Globally interpreting B-form polynomials 157

4 System Identification with Simplex Splines 161

4.1 Model Structure Selection 162
4.1.1 Geometric model structure selection 163
4.1.2 Polynomial model structure selection 165
4.1.3 A linear regression model for B-form polynomials 165
4.2 Spline Model Estimation 167
4.2.1 Generalized Least Squares with simplex splines 167
4.2.2 Constrained recursive regression with simplex splines 173
4.3 Model Quality Assessment 177
4.3.1 Model residue analysis 178
4.3.2 Statistical model quality assessment 180

4.3.3 Stability analysis . . . 181
4.4 Scattered Data Modeling with Simplex Splines . . . 182
4.4.1 Experiment setup . . . 183
4.4.2 Model residue analysis . . . 186
4.4.3 Statistical model quality analysis . . . 186
4.4.4 Stability analysis . . . 189
4.4.5 Empirical confidence bounds . . . 192
4.4.6 Statistical confidence bounds . . . 193

5 Aerodynamic Modeling of the Cessna Citation II 197
5.1 Introduction . . . 199
5.1.1 A procedure for AMI with simplex splines . . . 199
5.1.2 The Cessna Citation II and its FTIS . . . 200
5.2 Flight Test Design . . . 203
5.2.1 Aircraft Equations of motion . . . 203
5.2.2 Determination of the force and moment coefficients . . . 205
5.2.3 Initial assumptions on model structure . . . 206
5.2.4 Test point selection . . . 210
5.2.5 Maneuver input design . . . 213
5.3 Flight Test Results . . . 216
5.3.1 Flight test results . . . 216
5.3.2 Flight envelope coverage . . . 223
5.3.3 IMU sensor noise analysis . . . 227
5.3.4 IMU center of gravity offset correction . . . 229
5.4 Flight Path Reconstruction . . . 234
5.4.1 Flight path reconstruction procedure . . . 235
5.4.2 The iterated extended Kalman filter . . . 235
5.4.3 Kalman filter model structure . . . 239
5.4.4 Investigation of state observability . . . 244
5.4.5 Engine model . . . 246

5.4.6 Mass model 247
5.4.7 Flight path reconstruction results 248
5.5 Spline Model Estimation 261
5.5.1 A framework for AMI with simplex splines 261
5.5.2 Geometric model structure selection 262
5.5.3 Polynomial model structure selection 266
5.5.4 Prototype phase results 268
5.5.5 Prototype phase final model structure selection 270
5.5.6 Model refinement 285
5.6 Spline Model Validation 285
5.6.1 Raw validation results 286
5.6.2 Global model inspection 288
5.6.3 Model residual analysis 292
5.6.4 Stability analysis 300
5.6.5 Statistical analysis 307
5.6.6 Final model quality assessment 316

6 Conclusions 319

6.1 Simplex spline theory 320
6.2 System identification with simplex splines 322
6.3 Aerodynamic model identification with simplex splines 324

7 Recommendations 327

7.1 Simplex spline theory 327
7.2 System identification with simplex splines 328
7.3 Aerodynamic model identification with simplex splines 329

A Reference Frame Definitions 333

B IMU Geometry 337

Bibliography 339

Dutch Summary 357

Acknowledgments 363

Curriculum Vitae 365

Chapter 1

Introduction

1.1 Modeling Reality

A model is an abstraction of physical reality in which mathematics are used to reduce the complexity of reality into a conceptual structure [79]. It is in the approach towards this abstraction that there exist two different philosophies, which are the phenomenological modeling approach on the one hand and the physical modeling approach on the other. Both approaches have an identical goal, which is achieving the most accurate approximation to physical reality that produces the best possible predictions using the simplest possible model structure.

The physical modeling approach can be classified as a 'theory first' approach in which model structure and dynamics are derived directly from a set of *first principles* in the form of the laws of physics. The phenomenological modeling approach is a more pragmatic one, an 'observation first' approach, in which empirical observations of phenomena are linked together using a mathematical structure which is not directly reducible to first principles [78, 79, 83]. The phenomenological modeling approach has proven itself as a powerful tool when the underlying principles of a system are unknown. The difference between the two approaches is therefore that the physical modeling approach requires a firm foundation on the fundamental laws of physics, while the phenomenological approach does not require such a foundation.

While both model approaches have been used successfully throughout history, the physical model approach has invariably been valued more than the phenomenological approach. Because the physical model approach is based on the laws of physics, it is assumed to produce models that achieve a higher power of prediction than phenomenological models, which do not have such a deep underlying philosophy. The physical model approach has

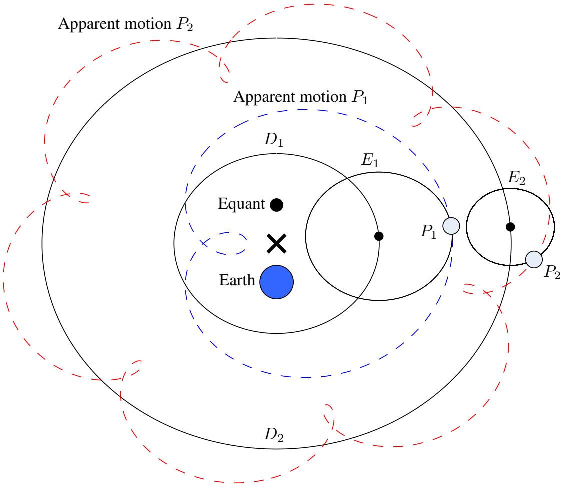


Figure 1.1: An example of a model based on an erroneous first principle: The Ptolemaic cosmological model in which planets move on epicycles which move on deferents. The deferents are centered on the center of the cosmos which lies exactly in between the Earth and an equant.

an inherent pitfall, however, and that is the presumed absoluteness of its first principles[1] [79]. When a new observation on some physical phenomenon indicates that a first principle is incomplete, or even erroneous, then all models built upon that principle are invalidated. In the best case the first principles are modified or extended to fit the new observations. In the worst it can lead to the outright rejection of observational disproof of the principle, potentially leading to centuries of scientific stagnation.

An illustrative example of how an erroneous first principle can lead to such stagnation is the Ptolemaic model of the Solar System (Figure 1.1). The Ptolemaic model of the solar system was based on first principles postulated by the influential Greek philosopher Aristotle (384 BC322 BC) in his cosmology. Aristotle's first principles on cosmology assumed that all celestial bodies moved on perfect circles around the Earth, and that the Earth was the center of the cosmos [120]. Created by the Greek astronomer and mathematician Ptolemy (ca.90-ca.168), the Ptolemaic model of the solar system was based on the astronomical and mathematical knowledge accumulated over 1000 years by the Sumerians of Mesopotamia and Ptolemy's predecessor Hipparchos [170]. Hipparchos(ca.190 BC−ca.120 BC) had earlier discovered that the orbit of the moon was eccentric, which was in contradiction with Aristotle's first principles on cosmology [206].

---

[1] Edmund Husserl, founder of phenomenology, states in his Amsterdam Lectures "*The sense of this method [phenomenology] in men like Mach and Hering lay in a reaction against the threatening groundlessness of theorizing in the exact natural sciences.*"[79]

Hipparchos had solved the problem by assuming that the moon moved on an *epicycle*, which was a secondary smaller circular orbit superimposed on its main orbit, called the *deferent* (Figure 1.1). Over the centuries astronomers had discovered that not only the orbit of the Moon, but also the orbits of the planets were eccentric. Ptolemy found that these eccentricities could not be modeled with any accuracy using Hipparchos' model. Rather than questioning the truth of Aristotle's first principles, Ptolemy solved the problem by adding one new entity to Hipparchos' model structure in the form of the *equant*. By adding additional equants to the epicycles of the planets, Ptolemy could increase the accuracy of his model such that the error between the predicted and observed angle of longitude of a planet was never larger than 10 minutes of arc[2] [116].

It would be another fifteen centuries before Aristotle's first principles were finally falsified by observational evidence. In January 1610, Galileo Galilei (1564-1642) had pointed his self-made telescope at the planet Jupiter and discovered its four moons [62, 48]. These four moons did not orbit the central sphere of the cosmos as Aristotelian cosmology dictated, but another planet. Galileo continued his observations and using telescopes of ever increasing power he discovered the four phases of the planet Venus, which proved that the Sun was the actual center of the Solar System [48].

Ptolemy's model of the cosmos would be superseded by Johannes Kepler's $(1571-1630)$ laws of planetary motion [212]. Kepler's laws of planetary motion were phenomenological in nature as they were not based on any first principle. Kepler's second law[3] (the Area Law) was exact, however, and would be proved by Isaac Newton 70 years later using his laws of gravity and motion.

In this thesis a case is made for a balanced approach towards the physical and the phenomenological modeling philosophies. While the physical approach can result in models with a higher power of prediction than phenomenological models, it also assumes that the underlying laws of physics are known, complete, and correct. In situations where the laws of physics are unknown or incomplete, or the involved mathematics too complex to be solved analytically, accurate phenomenological models can still be produced. Furthermore, the phenomenological approach tends to be more open to first-principle defying observational evidence than the physical approach, simply because the implications of such evidence are limited only to those models that are directly dependent on these observations. For the physical approach, an observational disproof of a first principle can have far-reaching consequences for the whole of science.

As the knowledge of the workings of nature and the science of mathematics advances, valid phenomenological models can eventually be translated into a form based on first principles. Examples of phenomenological models translated into physical models are Kepler's second law of planetary motion, which required Newton's differential calculus, and

---

[2] As a comparison, the angular resolution of the unaided human eye is 1 to 2 arcminutes.

[3] Kepler's second law of planetary motion states that the velocity of a planet changes at each moment such that the time between two positions is always proportional to the area swept out on the orbit between these positions.

the second law of thermodynamics which required the development of statistical mechanics by Ludwig Boltzmann (1844-1906). Modern examples of pure phenomenological model are Whitcombs ‘area rule’ for reducing wave drag during supersonic flight [215], and the Modified Newtonian Dynamics [133], which is an alternative non-Einsteinian universal theory of gravity which can explain the rotational velocities of galaxies [221] without having to assume the existence of dark matter [133, 129]. A final example of phenomenological models are the aerodynamic models created in this thesis.

1.2 Models, Identification, and Control

The primary purpose of a model of a physical system is the prediction of system dynamics based on current and past system states. When used inside a model based controller, these predictions are used in the derivation of laws for controlling the system. Currently, model based controllers play an essential role in many aspects of modern society. Applications of model based controllers range from food processing to engine controllers in cars, and from process control in oil refineries to the flight control systems of modern aircraft [180, 173]. At present, the most widely used model types at the heart of such control systems are polynomial models, spectral models, B-splines, and neural networks. In this thesis, a new type of model is proposed in the form of the multivariate simplex spline model. It will be shown that this new model type has a number of compelling advantages over current model types in terms of computational efficiency, approximation power, and flexibility. As such, the multivariate simplex spline models have the potential to increase controller performance in many industrial applications.

In Figure 1.2, an overview of a model based control system is shown. In this case the controlled system is an aircraft, but any other system could be controlled with this control scheme simply by changing the specific model implementation. The control loop in Figure 1.2 actually consists of two loops. The first loop is a static control loop. In the static control loop, the various aircraft states like angle of attack and Mach number are fed through a state estimation routine which removes sensor biases and noise. These *reconstructed states* then form the input for the model based flight control system. At the heart of this flight control system is the system model, in this case an aerodynamic model. The system model predicts the system output based on the current system states, and these predictions are then used to generate control laws. In the case of the flight control system in Figure 1.2 the control laws lead to commands for the control surface actuators of the aircraft. The second control loop shown in Figure 1.2 is the adaptive loop. Like the static loop, the adaptive loop also uses the state estimation routine to remove sensor biases and noise, but this time the reconstructed states form the input of a model estimation block. This block uses the incoming observations on the aircraft states to reconfigure the system model if its predictions differ significantly from the observations. This can for example be the case if the

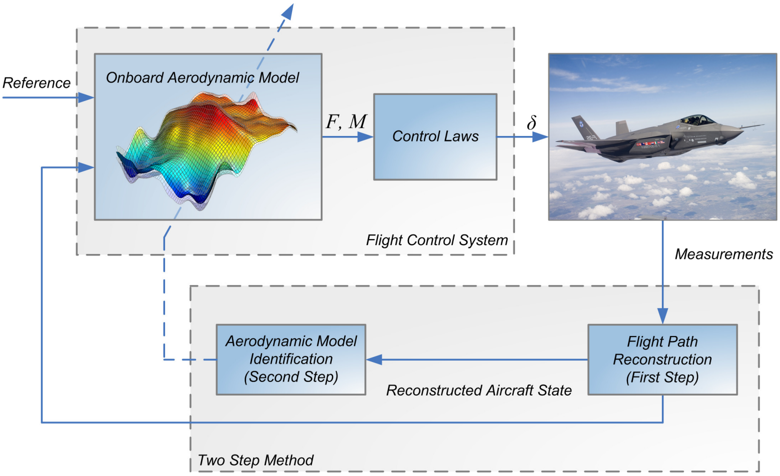

Reference
Onboard Aerodynamic Model
F, M
Control Laws
δ
Flight Control System
Measurements
Aerodynamic Model Identification (Second Step)
Reconstructed Aircraft State
Flight Path Reconstruction (First Step)
Two Step Method

Figure 1.2: A high level overview of a model-based flight control system.

system has been damaged. After reconfiguration, the system model should produce higher quality predictions, allowing the control system to adapt to the changed system dynamics.

Creating models of dynamic systems is a challenging task, especially if the dynamics of the system are nonlinear. It is the research field of *system identification* that is concerned with the *identification* of models of dynamic systems based on their inputs and outputs. System identification allows the creation of accurate models of systems even if the underlying *first principles* of the system are unknown. The field of system identification was founded in the 1970's and has since then grown into an active research area [65, 119]. In its most high-level form, system identification consists of three phases; a model structure selection phase, a model parameter estimation phase, and a model validation phase. Model structure selection is concerned with the definition of the abstract functional form of the model. During the second phase, the model parameter estimation phase, the model parameters are estimated using a parameter estimator and a set of input-output data. The created model is then put through the third phase, the model validation phase. Based on the outcome from the validation, a decision is made whether to start a new identification cycle and again go through these three phases, or to terminate the process and output the model.

When the dynamics of a system change, for example after a structural failure, then its onboard model may no longer be accurate, possibly leading to loss of control. In that case a recursive system identification method must be used to adapt the system model based on incoming observations. Recursive system identification methods operate in-the-loop, and as such must be optimized for real-time functionality. Recursive system identification and its algorithmic implementations have seen many innovations in the past, see e.g. [199, 118, 117, 12]. Current trends in recursive system identification point towards

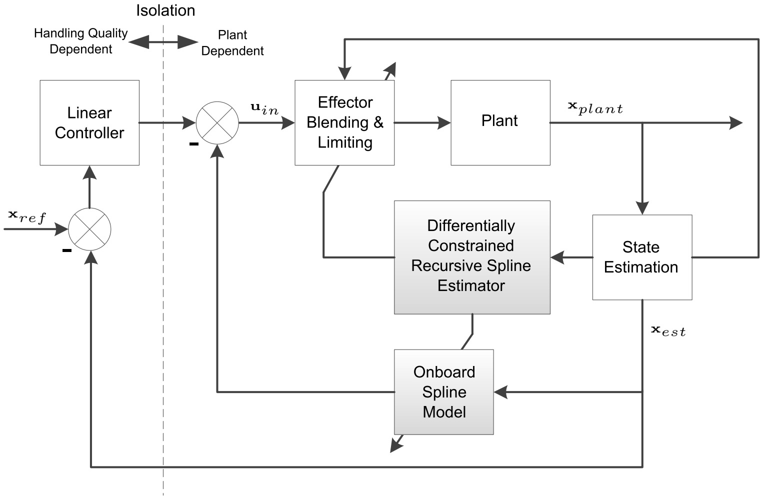


Figure 1.3: Control loop for an adaptive multivariate simplex spline model based controller.

nonlinear recursive identification methods like recursive neural networks [134], direct weighting methods [182], and modified stepwise regression methods [125]. A new method for recursive system identification with multivariate simplex splines was first introduced in [41]. This method uses a constrained recursive least squares estimator for the coefficients of the simplex splines. The recursive estimator for simplex splines would be valuable in an adaptive model based controller for complex nonlinear systems with changing dynamics (Figure 1.3).

The models identified with the identification methods in this thesis are *global models*, which in the context of this thesis are defined as follows:

Definition 1. *A global model is a spatial structure that can instantaneously predict system output at any location in the operating domain of the system.*

A global model differs from a local model in the sense that it has an a-priori 'presence' at every location inside the operating domain of a system. As such, global models can be used to predict system output independently from the current state of the system. In contrast, local models are valid only within a limited radius[4] around the current system state. Local models can therefore not be guaranteed to produce accurate predictions at locations beyond a limited radius about the current system state. The advantage of global models over local models is most apparent during recursive model identification. While a part of the model is being reconfigured using incoming observations, the remainder of the model can be used as

---

[4] This radius should be seen as the geometric radius vector in the state dimensions of the system, which is centered at the current system state.

an a-priori source of information. This can prevent the occurrence of sudden fundamental changes in the basic model structure for which there can be no physical explanation.

1.3 Introduction to Aerodynamic Model Identification

Detailed aerodynamic models play a crucial role in the design and operation of flight simulators and flight control systems. The function of aerodynamic models in flight simulators is the prediction of the required movements of the motion base in order to produce an experience that is as close as possible to real flight. For flight control systems, aerodynamic models are used to predict the real aerodynamic forces and moments acting on an aircraft as a function of its states. These predictions are then used to generate control laws resulting in control actions like the deflection of control surfaces and the changing of engine power output.

Identifying aerodynamic models involves the synthesis of computer fluid dynamics (CFD) results, wind tunnel measurements, and flight test data. In most cases an initial aerodynamic model of an aircraft is created using wind tunnel and CFD results. This initial aerodynamic model in most cases does not cover the complete flight envelope because some dynamic states cannot be reached in wind tunnels, while CFD results may be inaccurate in nonlinear regions of the flight envelope . The goal of flight testing is to validate and expand an initial aerodynamic model such that it covers the complete flight envelope with sufficient accuracy.

One of the first documented systematic flight test campaigns was conducted by F.H. Norton at the beginning of the 1920's. Norton used a converted Curtiss Jenny biplane (Figure 1.4) to determine models for the static and dynamic stability of aircraft [156, 157]. In 1951, Shinbrot was the first to use parameter estimation techniques to identify aerodynamic models from flight data. Using a least squares parameter estimator, Shinbrot determined the coefficients of differential equations describing aircraft responses to transient inputs.

The development of the field of system identification in the beginning of the 1970's led to a much more rigorous approach to aerodynamic model identification. In the following years, experiments with different system identification and parameter estimation techniques applied to aerodynamic model identification were performed, most notably by Taylor [205], Gerlach [64], and Iliff [82].

From the middle of the 1970's, aircraft system identification had become an essential element in the design of aircraft. More advanced methods for aerodynamic model identification based on Kalman filters [91], maximum likelihood estimation [81, 88], stepwise regression [90, 89], and more advanced frequency domain identification methods [88] were developed. While digital computers were getting ever more powerful, in particular the output error method with its maximum likelihood estimator was difficult to

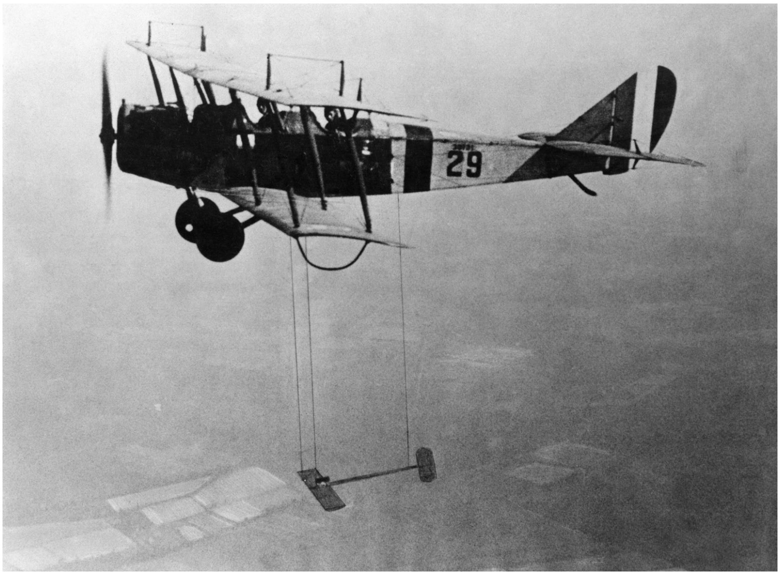


Figure 1.4: Curtiss JN-4H Jenny conducting a flight test in the early 1920's.

use in practice because of the computationally expensive nonlinear optimization problem it presented.

In order to reduce the complexity of the output error method, the Two-Step method was developed at the Delft University of Technology in the 1970's and 1980's [64, 146, 147]. The Two-Step method decoupled the joined nonlinear state-estimation and model parameter estimation problem into two separate optimization problems. In the first step of the Two-Step method, a Kalman filter is used to estimate the true states of an aircraft from biased and noise contaminated sensor measurements [149]. The second step of the Two-Step method is the aerodynamic model parameter identification step, which can now be posed as an equation error problem in terms of the estimated aircraft states.

As aircraft performance increased, so did the complexity of their flight control systems. Simple constant gain linear feedback control systems could be used to control aircraft with limited flight envelopes, but were inadequate for controlling high performance aircraft with large flight envelopes and significant nonlinear aerodynamics. A solution to this problem was to partition the flight envelope of a high performance aircraft into regions with different feedback gains. This, however, created problems on its own as it could not be guaranteed that there were no discontinuities in the resulting control structure. A more elegant solution to the problem would come in the form of feedback linearization (FBL) control [63, 15]. The form of FBL best suited for use in flight control systems was the nonlinear dynamic inversion (NDI) control [111, 178, 179]. NDI uses an accurate nonlinear global model of aircraft aerodynamics to cancel out the real nonlinear aerodynamics of the aircraft,

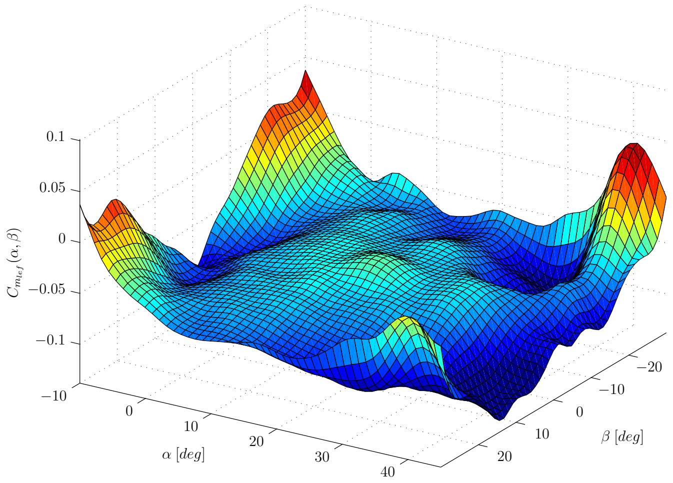


Figure 1.5: The (nonlinear) effect of the leading edge flap on the aerodynamic pitching moment coefficient of the F-16.

resulting in a linearized system. The resulting linearized system can then be controlled using classic linear control techniques. The challenge of using NDI in a real life flight control system is that a sufficiently accurate aerodynamic model is required, because an incomplete cancellation of the nonlinear aerodynamics will result in a system that cannot be adequately controlled using a linear controller [179].

The requirement for more accurate aerodynamic models led to many advances in the field of aerodynamic model identification in the middle of the 1990's. These advances were related not only to the development of new identification techniques, but also to the development of optimal input sequences to increase the efficiency of flight test maneuvers. Mulder had earlier used orthogonal input functions, which were optimized based on norms of the Fisher information matrix of an a-priori aircraft model [147, 148, 68]. Morelli used a different method for input optimization which was based on dynamic programming [143, 139, 140, 141, 85]. More advanced methods for aerodynamic model identification were to follow these developments. Linse for the first time used artificial neural networks to model aircraft aerodynamics [115], Laban created an online aerodynamic model identification method [94], while Morelli used an orthogonal least squares estimator [138, 142] for aerodynamic model identification.

It was soon found that models based on ordinary polynomials had insufficient approximation power to model the highly nonlinear aerodynamics of the new generation of fighter aircraft (Figure 1.5). These limitations could be avoided by using a non-global online identification method, like the modified stepwise regression method introduced by Lombaerts [123, 124, 125, 121] and advanced model-free control methodologies like adaptive backstepping [200, 201, 202, 211].

While polynomials lack the approximation power to produce global models of sufficient accuracy, they can still be used in the estimation of local models that are valid only on partitions of the flight envelope. The resulting set of polynomials can then be blended together using a smoothing technique like fuzzy blending [1, 122] or neural networks [44]. A set of numerical tables can subsequently be derived from the blended polynomial models. This approach was used for the creation of the aerodynamic model of Eurofighter Typhoon [162, 160, 158, 161], the F16-XL [110], and the X-31[181]. The disadvantage of this approach is that the validation and refinement of data table based models is cumbersome at best [162].

Until recently, the only alternative global aerodynamic modeling method of sufficient approximation power were methods based on neural networks[13, 72, 44, 84, 2] and derivative techniques like polynomial neural networks [168, 165, 164, 169]. While neural networks are powerful function approximators they suffer from numerical instabilities [45] and are inherently intransparent as a result of their black-box nature [8].

In this thesis, a new method for aerodynamic model identification is presented which combines the approximation power of neural networks and data table based models with the transparency and numerical stability of polynomial models. This method is based on a recent type of multivariate spline, the multivariate simplex spline [31, 7, 104, 40]. Multivariate simplex splines consist of polynomials defined on simplices which are geometric structures that minimally span a set of dimensions. The basis functions of the simplex splines are Bernstein basis polynomials, which guarantee a stable local basis. In this thesis, the theory of multivariate simplex splines is introduced in Chpt. 2, extended in Chpt. 3, and integrated with a framework for system identification in Chpt. 4. Finally, the new methodology for aerodynamic model identification is demonstrated with the identification of a complete set of aerodynamic models for the Cessna Citation II in Chpt. 5 using flight data obtained during seven test flights.

1.4 A Case for Multivariate Splines

Creating accurate global models of nonlinear systems with large operating domains requires the use of modeling methods of sufficient flexibility and approximation power. Currently, there are only a handful of methods available for modeling such systems. These methods are neural networks, kernel methods, polynomial models, and spline models.

Neural networks are powerful function approximators that have been used successfully in many fields of science and engineering, see e.g. [13, 72, 168, 44, 54, 84, 11, 165, 164, 169, 134]. Neural networks are black box models, and therefore suffer from an inherent intransparency [8]. This has as result that their performance can only be guaranteed under special circumstances [11]. Additionally, the radial basis functions in a neural network have a global influence on neural network output, with the result that all basis functions and coefficients need to be considered during evaluation and estimation, resulting in non-sparse and therefore inefficient evaluation and solution systems. It can therefore be argued that neural networks based on radial basis functions are not well suited for large scale identification problems because of computational limitations.

Kernel methods like the support vector machines can also be used for scattered nonlinear data modeling [50, 189, 51]. While very powerful in their role as categorizers, kernel methods are non-parametric in nature [17]. This produces computationally expensive optimization problems, especially when the modeled datasets are large. In general, kernel methods require that every significant data point is associated with a single kernel function, with the result that the size of the optimization problem is proportional with the total number of data points [17].

Polynomial modeling methods are perhaps the simplest and the most widely used of all modeling methods. While a single polynomial has only a limited approximation power (Figure 1.6), accurate models can still be created by estimating local models on subregions of the system operating domain [122]. This, however, introduces discontinuities in the model which can form a problem if it is to be used in some model based controller. Continuity can be restored by employing a blending scheme like Takagi-Sugeno fuzzy blending [1, 8, 122]. While fuzzy blending techniques are powerful, their construction and tuning is done based on expert knowledge which means that they will probably never result in a fully automated identification technique [8].

Finally, polynomial spline methods have been used in the past for the modeling of nonlinear systems, see e.g. [198, 89, 10]. These spline methods employed multivariate tensor product B-splines which are incapable of fitting scattered data and can only be used on rectangular domains [3].

In this thesis a case is made for using multivariate simplex splines in global models of nonlinear systems with large operating domains. Multivariate simplex splines combine most of the advantages of the above mentioned methods, without suffering from their disadvantages. The advantages of the multivariate simplex splines can be summarized as follows. Firstly, the simplex splines have an arbitrarily high approximation power on a global model scale. Secondly, simplex spline models are parametric models, which allows for efficient approximation of very large datasets. Thirdly, the simplex splines are linear in the parameters, which means that linear regression methods can be used for their estimation [40]. Finally, the simplex splines have a local polynomial basis, which implies that only small subsets of parameters and basis polynomials need to be considered during estimation and evaluation, resulting in efficient computational schemes.

In Figure 1.6 the modeling performance of the multivariate simplex splines is compared with a radial basis function neural network and a high degree polynomial model. The spline function and the neural network both have 750 optimization parameters. The figure clearly shows that the 30^{th} degree polynomial model can only produce a rough approximation to the original dataset. Using polynomials of even higher order did not result in higher quality models due to the numerical instability of the optimization problem. The neural network, in this case a simple single layer network consisting of 750 neurons[5], could not accurately approximate local details in the target dataset. Additionally, the solution system for the neural network was essentially non-sparse (100% non-zeros) resulting in a computationally inefficient optimization problem. From Figure 1.6, it can be seen that the simplex spline model closely resembles the modeled dataset, both on a local scale as well as on a global scale. Furthermore, the solution system for the multivariate simplex splines was highly efficient because of its 99% sparseness factor (more than 99% zeros). While the scope of the illustration in Figure 1.6 is limited, it nevertheless highlights some of the advantages of the multivariate simplex splines over other methods, and helps make the case for using the simplex splines as the modeling tool of choice in this thesis.

1.5 Thesis Goals and Research Approach

1.5.1 Goals of this thesis

The goal of this thesis is the development of a new time domain methodology for global model identification based on multivariate simplex splines. This new methodology should be able to identify global models of higher quality than existing methods. Additionally, the new methodology offers a number of unique model quality assessment methods that enable a thorough analysis of parameter estimator performance, model coherence and stability, and model residue.

The goal of this thesis is reflected by the main research statement:

"*A methodology for global nonlinear model identification based on multivariate simplex splines can outperform current global model identification methods on aspects of model quality and model quality assessment.*"

The main research statement can be split into three research questions. The first question is related to the mathematical theory of the multivariate simplex splines:

"*What theory should be added to current multivariate simplex spline theory to enable its integration in a framework for system identification?*"

---

[5] Neuron placement was done using the clustering algorithm from Park et al. [169]

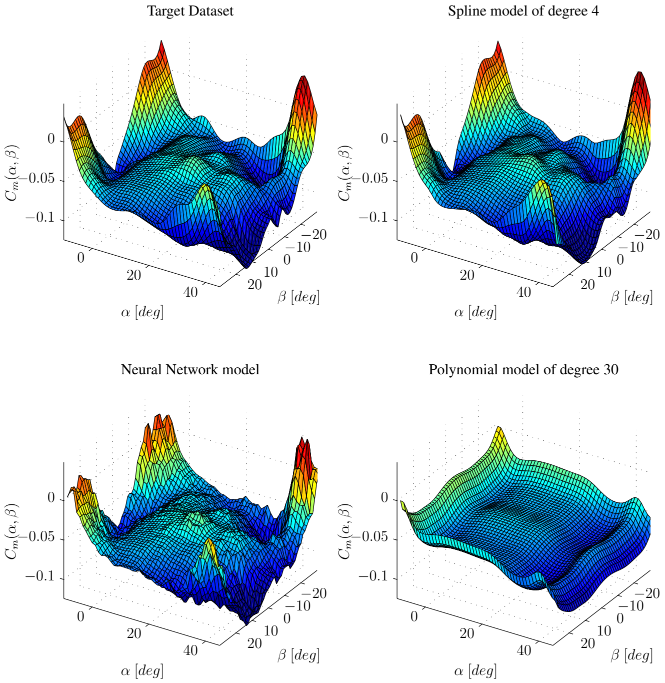

Target Dataset
Spline model of degree 4
Neural Network model
Polynomial model of degree 30

Figure 1.6: Comparison of three models of a nonlinear aerodynamic dataset (top left) created with three different modeling methods; a multivariate simplex spline model (top right), a 750-neuron neural network (bottom left), and a high order polynomial model (bottom right).

The second question is related with the framework for system identification with multivariate simplex splines:

"*What form should a framework for system identification based on multivariate simplex splines take?*"

The third question concerns the demonstration of the new methodology for global nonlinear model identification using a real life dataset:

"*Which advantages of the new methodology for global model identification can be demonstrated in a real-life aerodynamic model identification experiment?*"

The main research statement, together with the three questions, determined the research approach taken in this thesis.

1.5.2 Research Approach

The research approach taken in this thesis follows from the research questions and consists of three main facets. The first facet is the extension of multivariate simplex spline theory to allow its use in a framework for system identification. The second facet is the development of this framework, while the third facet is the demonstration of the new method with a real-life aerodynamic model identification experiment.

The following developments in multivariate simplex spline theory were anticipated:

1. A matrix formulation for the basis polynomials of the simplex splines is required.
2. A method for creating triangulations suitable for use with simplex splines should be developed.
3. A general method for formulating continuity conditions should be developed.
4. The global effects of continuity should be investigated.
5. A new type of constraint for limiting polynomial divergence should be developed.
6. A formulation of the basis polynomials in terms of global Cartesian coordinates should be developed.

Using these developments, a framework for system identification with simplex splines can be defined. This definition requires the following developments:

1. A technique for the structure selection of multivariate simplex spline models should be developed.
2. One or more parameter estimators for the simplex splines need to be developed.
3. The parameter covariance matrix should be estimated.
4. Methods for creating statistical and empirical confidence bounds for the spline model should be developed.
5. A method for assessing the coherence and stability of the simplex spline models should be developed.

Finally, the new method for system identification is demonstrated by identifying a complete set of aerodynamic models for the Cessna Citation II laboratory aircraft based on flight test data. The following steps are required for a successful demonstration:

1. Flight test data needs to be obtained.
2. The flight test data should be pre-processed to remove sensor glitches, measurement rate discrepancies, and engine induced noise sources.

3. An engine model should be used to estimate engine thrust during flight test maneuvers.
4. A mass model should be used to estimate mass and moments of inertia.
5. A flight path reconstruction method must be applied to estimate the true aircraft states.
6. A set of candidate spline dimensions should be defined.
7. A set of candidate polynomial spaces should be defined.
8. A set of prototype aerodynamic models should be created to allow the determining of the dimension and polynomial space of the definite simplex spline model.
9. The final simplex spline based aerodynamic models should be estimated and validated.

1.5.3 Scope and limitations

The new methodology for system identification presented in this thesis is focused on time domain model identification.

As mentioned in Sec. 1.5.1 and Sec. 1.5.2 the research in this thesis is focused on developing a new method for identifying models of nonlinear systems. Therefore, no dynamic simulations are performed using the identified aerodynamic models.

While an in-depth validation of the identified aerodynamic models is performed, no dynamic simulations are realized using the identified aerodynamic models. Additionally, the aerodynamic models identified in this thesis are not used inside any control systems, although doing so would be relatively straightforward, using for example a nonlinear dynamic inversion control design.

Finally, the performance of the new methodology for global model identification is compared with standard polynomial model identification methods, and not with techniques like for example (polynomial) neural networks. Such a comparison was not made in this thesis because to the knowledge of the author none of the more advanced techniques were, in their current forms, suitable for large scale, high-data volume identification experiments.

1.6 Contributions

The research performed in the framework of this thesis has lead to a number of publications in peer reviewed scientific journals and conference proceedings. A summary of these publications is provided below:

- C. C. de Visser, Q. P. Chu, and J. A. Mulder. A new approach to linear regression with multivariate splines. *Automatica*, 45(12):2903–2909, 2009.

- C.C. de Visser, J.A. Mulder, and Q.P. Chu. Global nonlinear aerodynamic model identification with multivariate splines. In *AIAA Atmospheric Flight Mechanics Conference*, 2009.

- E. de Weerdt, C. C. de Visser, Q. P. Chu, and J. A. Mulder. Fuzzy simplex splines. In *IFAC SYSID 2009*, 2009.
- C.C. de Visser, J. A. Mulder, and Q. P. Chu. A multidimensional spline based global nonlinear aerodynamic model for the Cessna Citation II. In *AIAA Atmospheric Flight Mechanics Conference*, 2010.
- C. C. de Visser, Q. P. Chu, and J. A. Mulder. Differential constraints for bounded recursive identification with multivariate splines. *Automatica*, 2011. article in press.
- C. C. de Visser, Q. P. Chu, and J. A. Mulder. A global trim manifold based on multivariate splines for the Cessna Citation II. In *The 1^{st} European Aerospace GNC Conference*, 2011.

1.7 Outline of the Thesis

This thesis is roughly divided into four parts. The first part of the thesis is concerned with an introduction into multivariate simplex spline theory. Because the multivariate simplex splines are not (yet) widely known, an comprehensive introduction to the theory is provided in Chpt. 2. Additionally, a number of innovations are presented in Chpt. 2 such as the vector formulation of the B-form polynomials, and a novel algorithm for triangulation optimization.

The second part of this thesis is focused on introducing a number of new developments in the mathematical theory of the multivariate simplex splines. These developments are presented in Chpt. 3, and are aimed at enabling the use of simplex splines for system identification applications. In Chpt. 3 a new vector formulation of the B-form on a complete triangulation is given, and a new general formulation of the smoothness conditions that govern the continuity between the polynomial pieces of the spline function is introduced. A new effect called *B-net propagation* was observed in a specific class of triangulations. B-net propagation was found to occur for certain combinations of spline degree and continuity order, and can potentially negate the local model property of the simplex splines. Additionally, a new one-step matrix formulation of the de Casteljau algorithm is presented. This matrix formulation is essential in the definition of the differential constraints for simplex spline functions. The differential constraints allow the constraining of the directional derivatives of the simplex splines at specific locations within the spline domain. The differential constraints can be used to bound the directional derivatives of spline polynomials on the edges of the spline domain, thereby reducing polynomial divergence resulting in a bounded model extrapolation capability for the simplex splines. Finally, a new formulation of the B-form polynomials in terms of global Cartesian coordinates was derived. This formulation has many tentative applications, such as global triangulation optimization, and the physical interpretation of B-form polynomials.

The third part of this thesis is focused on the integration of the multivariate simplex splines in a framework for system identification. This part is the main topic of Chpt. 4. In this chapter the process of model structure selection for simplex splines is discussed. A constrained generalized least squares estimator and a constrained recursive least squares estimator are presented and a number of techniques for validating the estimated simplex spline models are introduced. Some of these techniques are unique to simplex spline models, such as the variance surface estimation and the model stability analysis through B-coefficient value assessment. Finally, the new methodology for model identification with simplex splines is demonstrated with a numerical experiment.

The fourth and final part of this thesis is the application of the new model identification method in a real-world aerodynamic model identification experiment. This experiment is described in detail in Chpt. 5. In this chapter the complete procedure, from flight test design to model validation is discussed.

The research performed in this thesis is divided into three main facets; multivariate simplex spline theory, system identification, and aerodynamic model identification. In Chpt. 6 a number of conclusions are provided on each of these three research facets. Finally, in Chpt. 7 recommendations arising from the research are presented.

Chapter 2

Introduction to Multivariate Simplex Splines

Splines have been an important tool in design and construction for many centuries. Splines, in the form of slender flexible beams, were first used by draftsmen in the middle-ages for the design of smooth curves for the bows and hulls of ships. In modern science and engineering, there are many uses for splines. Splines are used in fields as diverse as car design, medical imaging, signal analysis, and system identification. In essence, a spline is a piecewise defined function with a predefined continuity between the pieces. This piecewise nature allows splines to be used to fit data that is too complex to be approximated with any single functional piece.

The mathematical theory of one-dimensional, or univariate, splines is considered to be complete in the form of the extremely popular ordinary B-spline. In contrast, the theory of multi-dimensional, or multivariate, splines is still an active research area. At current, there are many different multivariate spline types, all with their respective strengths and weaknesses. There is one type of multivariate spline, however, that can be considered to be the true generalization of the univariate B-spline, inheriting most of its strengths, and almost none of its weaknesses. This multivariate spline type is the multivariate simplex spline. Multivariate simplex splines consist of ordinary polynomials which are defined on geometric structures called simplices. The approximation power of the multivariate simplex splines can then be increased by stitching together any number of simplices into a larger geometric structure called a triangulation. The most important advantage of the multivariate simplex spline over other multivariate approximation methods is that it can be used to approximate scattered multi-dimensional data on non-rectangular domains using only polynomials. These polynomials have a number of unique properties which makes them extremely useful when used inside a framework for system identification.

The objective of this chapter is to introduce the reader to the basic mathematical theory of the multivariate simplex spline. The chapter contains three main sections. In Sec. 2.1 an in-depth discussion on the basic principle of splines and the history of multivariate spline theory is presented. This section additionally serves as a motivation for the choice in this thesis for the multivariate simplex spline as the primary tool for global non-linear aerodynamic model identification. Then, in Sec. 2.2 the basic mathematical properties of the simplex splines are introduced. It will be explained that these mathematical properties are at the very heart of many of the unique features of the simplex splines. Finally, in Sec. 2.3 an in-depth analysis on the various aspects of triangulations is offered. The triangulation is the source of the approximation power of the multivariate simplex spline but at the same time presents a number of important challenges that, until now, have prevented the simplex splines from being used as a tool for system identification. The solutions to these challenges will be presented in this final section, setting the stage for the next chapter, which will introduce a number of new mathematical theories aimed at enabling system identification with simplex splines.

2.1 Introduction to Multivariate Simplex Splines

In the mathematical theory of splines, there are two main directions. The first is univariate spline theory which is the most reminiscent of the draftsman's spline and deals with smooth 1-dimensional curves. The second direction is multivariate spline theory, which deals with n-dimensional (hyper) surfaces. While univariate spline theory has been well developed in the past, multivariate spline theory is a much less finalized theory. In the literature, there exists a multitude of different types of multivariate spline, each with their particular strengths and weaknesses. Important properties for the various multivariate spline types are their scattered data fitting capabilities, their local basis properties and their generality with respect to the univariate B-spline. Starting at the simple end of the complexity spectrum there are the multivariate tensor product splines, which are considered somewhat of a 'poor man's multivariate spline' in the mathematical community[1]. Despite this, tensor product splines are quite efficient in terms of computing resources, and simple to understand and implement algorithmically. The great downside of tensor product splines is that they are, for fundamental reasons, incapable of fitting scattered data. Therefore, they cannot be considered a true generalization of univariate spline theory. On the extreme end of the complexity spectrum are the polyhedral splines. Polyhedral splines are very general and

---

[1] Carl de Boor, one of the founders of multivariate spline theory, summarizes his thoughts on tensor product splines as follows: "*I want to point out that there is available one way of generalization that is specifically designed to require no thought, no new idea (if this construction is satisfactory for you, I have nothing further to tell you). This is the tensor product construct.*"[32]

are, according to some influential authors, true generalizations of the univariate B-spline. The central idea of polyhedral splines is that the location of the control points in a space $\mathbb{R}^n$ are determined by the projection of a polyhedron in $\mathbb{R}^{n+s}$ with $m > 0$ on $\mathbb{R}^s$. While general and mathematically elegant, there exists no general scattered data approximation scheme using polyhedral splines in the literature.

The purpose of the multivariate splines in this thesis is the fitting of scattered n-dimensional data within a system identification framework. This implies a multivariate spline type which is general in all dimensions, can handle scattered data and has a simple and efficient implementation. At current, there is only one type of multivariate spline in the literature that has all these properties. This multivariate spline type is the multivariate simplex spline. The generality of the multivariate simplex spline with respect to the univariate B-spline is not absolute, as it completely does away with the concept of *knots*. Its mathematical formulation, however, is invariant of dimension and polynomial degree. This makes it an extremely powerful function approximation and data fitting tool which is perfectly adapted for use within a framework for system identification.

In this section a brief history of the multivariate simplex spline is told, starting with the univariate B-spline Sec. 2.1.1 and the difficulty of finding its n-dimensional generalization Sec. 2.1.2. Then, some of the most widely known multivariate spline types are presented: the tensor product splines in Sec. 2.1.3, thin plate splines in Sec. 2.1.4 and the polyhedral splines in Sec. 2.1.5. The main theoretical research topic in this thesis is an alternative multivariate spline type; the multivariate simplex spline, which is introduced in Sec. 2.1.6.

2.1.1 Univariate B-Splines

Splines have been used for hundreds of years by draftsmen for the design of smooth curved lines for the bows and hulls of ships. A spline in this sense is a thin strip of elastic material which is bended in shape and clamped at a number of locations by metal weights called ‘ducks’. The elastic properties of the material together with the location of the ducks, forces the spline in a curve that minimizes strain energy Figure 2.1.

According to Euler-Bernoulli beam bending theory, the curve that is made by the spline tool will approximate that of a cubic polynomial. This can be seen as follows. Let P_a be an externally applied force on a one-dimensional beam at location $A \leq a \leq B$, and let $w(x)$ be the deflection of the beam at any location $A \leq x \leq B$. The deflection of the beam is then given by the Euler-Bernoulli equation with constant stiffness:

$$P_a = EI\frac{d^4 w(x)}{dx^4}, \tag{2.1}$$

with EI the stiffness of the spline material. The solution of this fourth order differential equation is determined by the boundary conditions imposed on $w(x)$ and its derivatives. If

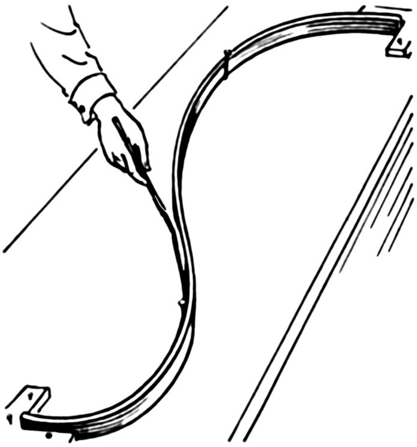


Figure 2.1: The draftsman's spline: a piecewise defined cubic polynomial (©Pearson Scott Foresman)

the spline tool is considered to have two simply supported ends at locations A and B, i.e.

$$w(A) = c_A, \ \frac{d^2w}{dx^2}|_A = 0, \ w(B) = c_B, \ \frac{d^2w}{dx^2}|_B = 0, \tag{2.2}$$

and letting $L = B - A$, then the solution to Eq. 2.1 is found to be:

$$w(x) = \begin{cases} B_1(x) & = & \frac{P_a bx(L^2-b^2-x^2)}{6LEI}, & A \leq x \leq a \\ B_2(x) & = & \frac{P_a bx(L^2-b^2-x^2)}{6LEI} + \frac{P_a(x-a)^3}{6EI}, & a < x \leq B \end{cases} \tag{2.3}$$

Interestingly, the analytical solution to the Euler-Bernoulli equation with the boundary conditions Eq. 2.2 is itself a piecewise polynomial, or a spline function. In Figure 2.2 the polynomial $w(x)$ is drawn, together with the two polynomial pieces $B_1(x)$ and $B_2(x)$ that define it. The polynomial spline function $w(x)$ from Eq. 2.3 is formulated in the so-called piecewise polynomial form, or pp-form. The pp-form is relatively easy to interpret as the individual polynomials that constitute the spline can readily be identified. Each of the polynomial pieces is valid only on a part of the total spline domain, e.g. in Eq. 2.3 we have that $B_1(x)$ is valid only on $A \leq x \leq a$, while $B_2(x)$ is valid only on $a < x \leq B$. The boundaries of the domains of the individual polynomials are called 'breaks'. As it turns out, however, the pp-form is difficult to work with for a number of different reasons. First of all, polynomials defined in the pp-form lack a stable local basis, which not only greatly increases the complexity of schemes for fitting data, but which can also lead to numerical instabilities. Then there is the problem of the definition of continuity between spline pieces for which the pp-form does not provide any obvious functionality. Finally, the formulation

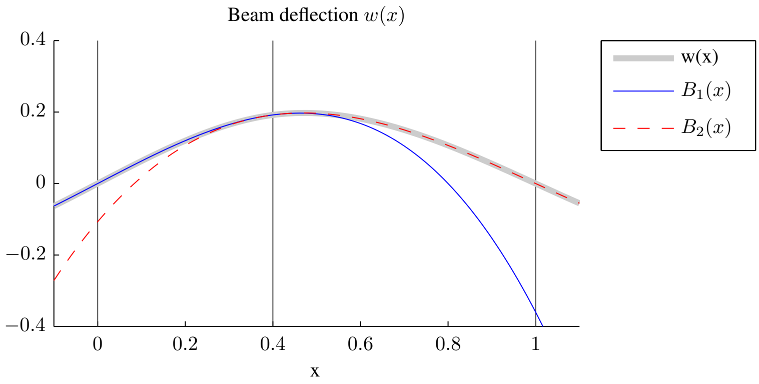


Figure 2.2: Deflection of a natural (cubic) spline consisting of two polynomial pieces due to a force acting at $a = 0.4$.

of the pp-form itself is rather cumbersome:

$$p(x) = \sum_{i=1}^{k} (x - \xi_j)^{k-i} c_{ji}, \quad j = 1, 2, \ldots, L, \tag{2.4}$$

that is, ξ_j is the x-coordinate of the j^{th} break and c_{ji} is the coefficient corresponding with the j^{th} break and the i^{th} polynomial term. Therefore, every break supports a complete polynomial of degree $k-1$ with k terms. Continuity is somehow coded in the matrix of coefficients c_{ji}. In short, the pp-form forces one to store a set of L breaks as well as a matrix of $k \cdot L$ coefficients. Luckily, a much more elegant alternative to the pp-form is available. In 1946 Schoenberg introduced[2] an alternative spline representation to the pp-form he called the 'basic k^{th} order spline curves' [187]. In 1967, based on work by Schoenberg's colleague Curry [20], Schoenberg named the 'basic spline' or B-spline [188]. The beauty of the B-spline is that it is completely determined by a sequence of knots, and a matching sequence of coefficients. Knots differ from the breaks of the pp-form in the sense that they can have a user-definable multiplicity, that is, knots are allowed to overlap while breaks are not. Knot multiplicity controls the continuity of the spline function at the location of the overlapping knots. The univariate B-spline consisting of a total of d basis functions has the following definition:

$$p(x) = \sum_{i=1}^{d} B_{i,k}(x) c_i \tag{2.5}$$

with d a positive integer and with *basis function* $B_{i,k}(x)$ a polynomial of degree $k-1$,

---

[2] Schoenberg himself states that spline curves where already known to Laplace in the 17^{th} century.

defined implicitly in terms of the knot sequence $(t_i, t_{i+1}, \ldots, t_{i+k})$ as follows:

$$B_{i,k}(x) = B(x|t_i, t_{i+1}, \ldots, t_{i+k}) \tag{2.6}$$

The set of basis functions forms a partition of unity as follows:

$$\sum_{i=1}^{d} B_{i-d+1,k}(x) = 1, \;\; x \in [t_i, t_{i+1}], \tag{2.7}$$

which means that at any point x on the interval $[t_i, t_{i+1}]$ there are d non-zero basis functions $B_{i-d+1,k}(x)$ to $B_{i,k}(x)$, which on the given interval sum up to a value of 1 [28, 29].

The basis functions of the univariate B-spline from Eq. 2.6 can be constructed recursively using the Cox-de Boor recursion, named after the inventors Cox [19] and de Boor [28] who independently discovered it. The Cox-de Boor recursion relationship is the following:

$$B_{i,1}(x) = \begin{cases} 1, & t_i \leq x < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{2.8}$$

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(x) + \frac{t_{i+k} - x}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(x) \tag{2.9}$$

In Figure 2.3 a set of six knots supports three third degree basis functions. On the knot interval $[t_i, t_{i+1}]$ all basis function sum op to one, demonstrating the partition of unity property of the B-spline. The graphical depiction of the B-spline in Figure 2.3 helps one to appreciate some of the powerful features of the B-spline. For example, adding a knot to the end of the knot sequence adds a complete, locally acting basis function to the B-spline function. New knots can be added on the fly without sacrificing continuity or disrupting the global model structure. This in turn allows B-splines to be used in for example real-time signal filters, like those presented by Unser [207, 208] and image compression filters, like that presented by Panda [167].

The continuity order of the B-spline function on the knots is determined by the so-called ‘knot-multiplicity’. Knot multiplicity allows for multiple knots to be placed at the same location, thereby influencing the structure of the basis polynomial. The relationship between the multiplicity of the knot t_i, the continuity order C^{r_i} at t_i and the degree d of the B-spline is given by:

$$r_i + \#t_i = d \tag{2.10}$$

which was taken from [33]. With Eq. 2.10 the continuity order at a specific knot location can be defined. For example, if we have a B-spline of degree 4, and we want C^2 continuity at t_i we need knot multiplicity $\#t_i = 4 - 2 = 2$.

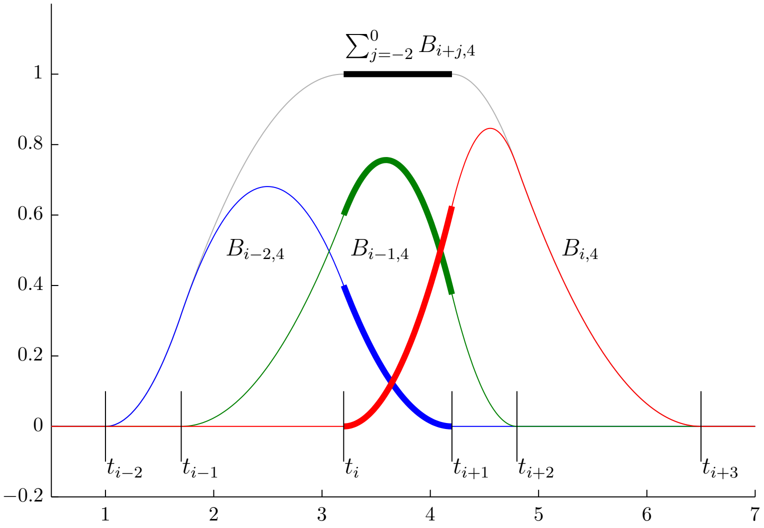


Figure 2.3: Three third degree basis functions defined on six knots.

At this point, the basic theory of the univariate B-spline has been introduced. This introduction, however, contains only a fraction of the complete mathematical theory on univariate B-splines. The interested reader is therefore referred to [29] which provides an excellent overview of the mathematical theory of univariate B-splines, and to [36] which can be considered the standard textbook on univariate B-splines.

2.1.2 The unclear path from univariate to multivariate splines

The elegance and power of the mathematical theory of univariate B-splines motivated the search for a multivariate generalization of the theory, which proved to be much more difficult than initially expected. Since the end of the 1950's many authors have contributed to various types of multivariate splines, each with their advantages and disadvantages. As of this writing, however, the debate about what should be considered the true multivariate generalization of the univariate B-spline is ongoing [32, 38, 34, 153]. In his 1990 paper, de Boor formulated the problem as follows. "*The generalization of univariate polynomial interpolation to the multivariate context is made difficult by the fact that one has to decide just which of the many of its nice properties to preserve, as it is impossible to preserve them all*"[38]. The following list of properties taken from [34] describes a true generalization of the univariate B-spline:

Table 2.1: Comparison of different multivariate spline types

Type	year	main innovators	local polynomial basis	general domains	scattered data fitting	efficient approximation algorithms
Tensor	1971	de Casteljau, Bézier	yes	no	no	yes $(O(N))$
Thin plate	1980	Meinguet, Franke	no	yes	yes	no $(O(N^3))$
Polyhedral	1986	de Boor, Dahmen, Micchelli, Neamtu	yes	yes	yes	no
Simplex	1986	de Casteljau, Farin, Lai, de Boor, Schumaker	yes	yes	yes	yes $(O(N))$

P1 Explicit formulae for the dimension of spline spaces.

P2 Explicit bases consisting of locally supported elements.

P3 Convenient algorithms for storing and evaluating the splines, their derivatives and integrals.

P4 Estimates of the approximation power of spline spaces.

P5 Conditions under which interpolation is well defined.

P6 Algorithms for interpolating and approximation.

As of current, there are four multivariate spline types which possess at least some of the above mentioned properties. They are the tensor product spline, the thin plate spline, the polyhedral spline and the simplex spline. In Table 2.1 the main properties of these four spline types are compared. The four multivariate spline types are discussed in detail in the next four paragraphs.

2.1.3 Tensor product splines

While mathematicians where struggling with the derivation of a true multivariate generalization of the univariate B-spline, an immediate need arose in the car and aviation industry for computer algorithms that could be used to model smooth two- and three-dimensional surfaces. In the car industry, the 1950's and 1960's saw a great rise in the use of curved lines and surfaces in the body works of cars, which were very hard to engineer for manufacturing. In 1959 Paul de Casteljau, a mathematician working at car manufacturer Citroën, invented a method for rendering smooth two dimensional surfaces using Bernstein polynomials in barycentric coordinates defined on rectangular and triangular patches [53]. This method, when defined on triangular patches, would later form the basis of the multivariate simplex spline. The work of the de Casteljau was considered so advanced that it was kept secret by Citroën until 1971, by which time Pierre Étienne Bézier (1910-1999), then head of

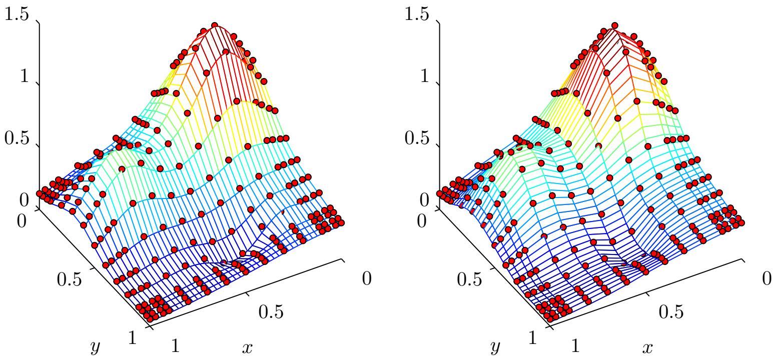


Figure 2.4: Approximation of Franke's function [61] along the x_0-axis (left) and x_1-axis (right) using sets of univariate B-splines.

the design department of car manufacturer Renault, had developed and widely published an almost similar method based on what were afterwards known as Bézier curves and Bézier patches[3]. Bézier used the exact same curves and patches as de Casteljau, but his mathematics were different as he did not use Bernstein polynomials nor barycentric coordinates in his definitions.

In 1972 Forrest for the first time applied Bernstein polynomials in Bézier curves and patches, thereby exactly reproducing de Casteljau's results on rectangular patches from more than a decade earlier [56]. The rectangular Bézier patches immediately led to one of the most widely used types of multivariate spline; the multivariate tensor product spline (TPS), which are essentially Bézier patches stitched together so that they form a continuous surface. Tensor product splines are constructed by taking the tensor product of any number of univariate spline functions, like for example the univariate B-spline. Tensor product splines had a simple and efficient algorithmic implementation, which led to their wide acceptance in CAD application used in industry [30, 145]. By their very nature, however, tensor product splines are incapable of fitting scattered data [16, 3]. This is not such an important issue when the tensor product splines are used for CAD purposes, because the data is artificial and can be placed exactly on a rectangular grid. However, data obtained by taking measurements of the state of some continuous physical system is inherently scattered, which means that tensor product splines cannot be used to model this data directly.

---

[3] Bézier curves and Bézier patches were used in the Renault CAD/CAM system UNISURF which was entirely based upon them. UNISURF, in turn, influenced developments at the French aircraft manufacturer Dassault who built a system called EVE which later evolved into the well-known and widely used CAD/CAM system CATIA (Computer Aided Three-dimensional Interactive Application).

The principle behind the tensor product spline is very simple. If one has two univariate functions $f(x)$ and $g(y)$ than one can construct a bivariate function by taking their tensor product: $p(x, y) = f(x) \otimes g(y)$. The bivariate tensor product spline can thus be constructed by taking the tensor product of two univariate B-splines:

$$f(x, y) = \sum_{i=1}^{r} \sum_{j=1}^{s} B(x|t_i, \ldots, t_{i+v}) B(y|t_j, \ldots, t_{j+w}) c_{ij}, \tag{2.11}$$

with $B(x|t_i, \ldots, t_{i+v})$ a set of basis functions of degree $v - 1$ in terms of x and with $B(y|t_j, \ldots, t_{j+w})$ a set of basis functions of degree $w - 1$ in y as in Eq. 2.6 and with $c_{ij} \in \mathbb{R}^{r \times s}$ a matrix of coefficients. Using the shorthand notation from Eq. 2.6 this can be simplified into the following:

$$f(x, y) = \sum_{i=1}^{r} \sum_{j=1}^{s} B_{i,v}(x) B_{j,w}(y) c_{ij}. \tag{2.12}$$

The working of the tensor product spline can be seen as follows. By first fitting a set of univariate B-spline curves in the y-direction, the expression Eq. 2.12 is reduced into an explicit function of x and an implicit function of y as follows:

$$f(x, y) = \sum_{i=1}^{r} B_{i,v}(x) c_{ij}(y). \tag{2.13}$$

This situation is visualized in the right hand plot of Figure 2.4. The set of B-spline curves in the y direction clearly uses the same set of knots. The complete bivariate tensor product surface is then created by using the knots of the univariate B-splines in the y-direction as knots in the x-direction, thereby producing a smooth surface as shown in Figure 2.5. The process can also be reversed with the exact same results by first fitting univariate B-spline curves in the x-direction as shown in the left hand plot of Figure 2.4, and then interpolating these curves in the y-direction. The total number of knots is thus simply the number of knots in the x-direction multiplied with the number of knots in the y-direction. In order to produce the value of $f(x, y)$ at the knot location (i, j), the value of the basis polynomial at that location is multiplied with the corresponding coefficient value c_{ij}. Because the B-spline curves in the x-direction are independent from the B-spline curves in the y-direction, basis polynomials of different degrees can be used along the different axis. For example, it is possible to produce a tensor product spline with 4^{th} degree polynomials in the x-direction and 2^{nd} degree polynomials in the y-direction. It is now easy to see that data points must be located exactly on the knot locations, because only then can they be approximated by *both* the B-spline curves in the x-direction and the B-spline curves in the y-direction.

The general tensor product B-spline of dimension n is created by taking the tensor

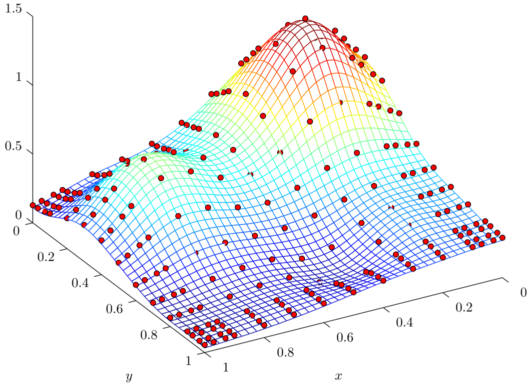


Figure 2.5: Bivariate tensor product spline approximation to Franke's function using a set of gridded data points.

product of n univariate B-splines as follows:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i_1=1}^{r_0} \sum_{i_2=1}^{r_2} \cdots \sum_{i_n=1}^{r_n} B_{i_1,v_1}(x_1) B_{i_2,v_2}(x_2) \cdots B_{i_n,v_n}(x_n) c_{i_1 i_2 \cdots i_n} \\ &= \prod_{j=1}^{n} \sum_{i_j=1}^{r_j} B_{i_j,v_j}(x_j) c_{i_1 i_2 \cdots i_n}, \end{aligned} \tag{2.14}$$

with the coefficient matrix $c_{i_1 i_2 \cdots i_n} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_n}$ an n-dimensional array of size $\prod_{j=1}^{n} r_j$.

Tensor product splines were the first true multivariate spline type. Their development was driven by the industry, and as such their mathematical generality was of secondary importance. Tensor product splines are very efficient computationally and have a simple software implementation. Their greatest limitation, however, is their inability to fit scattered data as well as their dependence on rectangular domains. As most real-life physical datasets are scattered and defined on non-rectangular domains, their use as a general modeling tool is very limited.

2.1.4 Thin plate splines

The fundamental inability to approximate or interpolate scattered data with tensor product splines led to the development of a number of alternative methods for modeling scattered

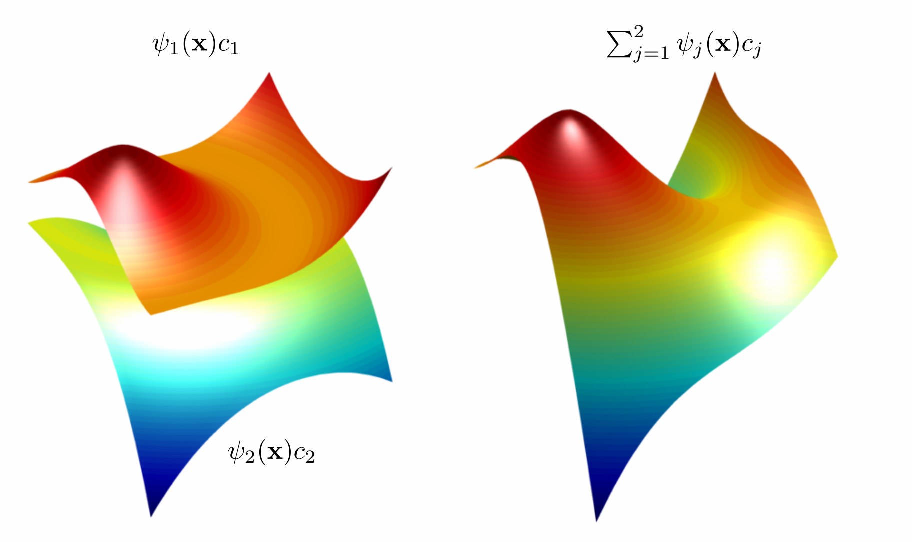


Figure 2.6: Principle of the thin plate spline: two radial basis functions (left) on a global domain combine into a single smooth thin plate spline surface on the same domain (right).

data. One of the most popular of these methods was the 'thin plate spline', named so because of the analogy with the bending of a plate due to some external force [130]. Such a plate will deform such that its bending energy is minimized, and as such a thin plate spline interpolation to a set of scattered data is determined by minimizing an energy integral. The basis functions of thin plate splines are radial basis functions (RBF) which are centered on the data points. As such, a thin plate spline has as much basis functions as there are data points. The basis functions are non-local, that is, they have a global influence on the thin plate spline function, see Figure 2.6. This means that all basis functions, and all coefficients must be used to evaluate the thin plate spline at a single location. Additionally, the global influence of the basis functions leads to very non-sparse solution systems for the coefficients of the thin plate splines.

Thin plate spline have seen many uses in industry, for example to model deflections of aircraft wings due to aeroelastic loads [71]. Because the thin plate spline has a non-local basis, however, solving the optimization problem involves all data points, and all basis functions which takes an $O(N^3)$ effort [32]. Franke demonstrated that this led to very inefficient interpolation algorithms, effectively limiting the dimension of the thin plate spline to just two [59]. Franke developed a local version of the thin plate spline which significantly reduced computational load, but which required the data to be placed in a rectangular grid of rectangles, and was limited to first order continuity [60]. Micchelli

Table 2.2: Different radial basis functions.

RBF type	definition
Gaussian	$\psi(\mathbf{x}) = \exp\left(-c\|\mathbf{x}\|^2\right), \ \ c > 0$
thin plate spline	$\psi(\mathbf{x}) = \|\mathbf{x}\|^\lambda \log\left(\|\mathbf{x}\|\right), \ \ \lambda \in 2\mathbb{N}: \text{ even}$
multiquadric	$\psi(\mathbf{x}) = \left(\|\mathbf{x}\|^2 + c^2\right)^{\lambda/2}, \ \ \lambda \in \mathbb{N}: \text{ odd}, \ c > 0$

proved a conjecture by Franke that any set of scattered data can be uniquely approximated with multiquadric surfaces [132].

In more recent years there have been some advancements in the area of thin plate splines. Important theoretical proofs on aspects of scattered data interpolation with radial basis functions were presented by Powell [172], Dyn and Ron [49], Schaback [185, 186] and de Boor [37]. Yoon presented an advanced type of thin plate spline called the shifted thin plate spline [216, 217, 218]. Shifted thin plate splines have a semi-local basis, resulting in more efficient approximation schemes than ordinary thin plate splines. However, as Yoon himself observes, the method requires special 'tricks' to ensure reasonable interpolation behavior near the boundaries of the spline domain [217]. These tricks involve the creation of virtual data points beyond the spline domain bounds through the extrapolation of existing data.

The thin plate spline is defined as follows:

$$f(\mathbf{x}) = \sum_{j=1}^{N} \psi_j(\mathbf{x}) c_j \tag{2.15}$$

with $\psi_j(\mathbf{x})$ a kernel function, with c_j a vector of coefficients and with N the total number of data points. The kernel function $\psi_j(\mathbf{x})$ can be any type of RBF, see Table 2.2.

Thin plate splines were an early solution to the scattered data interpolation and approximation problem. In contrast with the tensor product splines, thin plate splines had a scattered data modeling capability. The basis functions of the thin plate splines are non-local which directly leads to computationally inefficient algorithms for their creation and evaluation. Both its creation and its evaluation requires the evaluation of all basis functions as well as all coefficients. Thin plate splines can therefore not be considered a truly general scattered data modeling tool, as it is limited to only small datasets and low dimensions.

2.1.5 Polyhedral splines

The middle of the 1980's was a time of great innovation in the field of multivariate splines. Much of the mathematics used in current multivariate spline theory was developed by mathematicians in these days. What was considered to be the 'true' multivariate generalization of univariate B-spline theory was developed in the form of the polyhedral

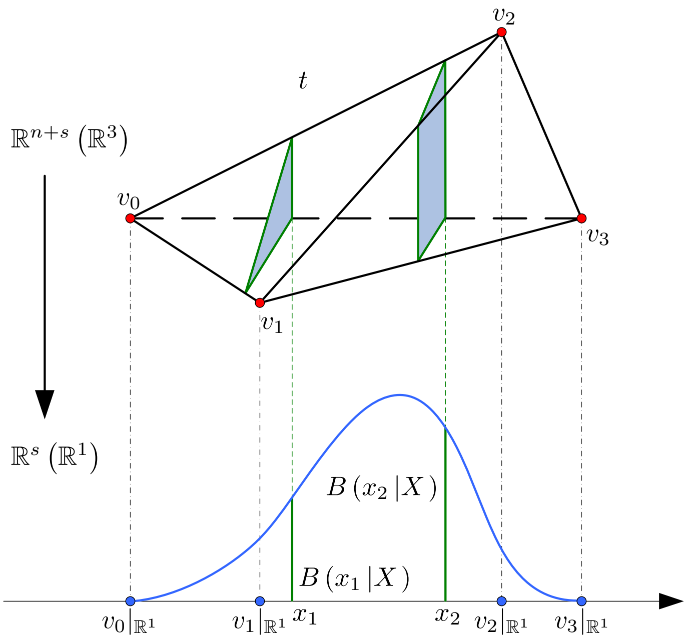


Figure 2.7: Principle of the polyhedral spline: a univariate B-spline as the shadow of a tetrahedron.

spline [32, 34]. The mathematical theory of the polyhedral spline combined most properties of the univariate B-spline into a framework for multivariate approximation. In essence, a polyhedral spline is formed by projecting a multidimensional polyhedron in $\mathbb{R}^{n+s}$ like a hypercube or simplex onto a lower dimensional plane in $\mathbb{R}^s$. The value of the multivariate polynomial at a specific location in $\mathbb{R}^s$ would be the volume of a slice of the polyhedron 'floating above' the plane in $\mathbb{R}^s$, see Figure 2.7. A polyhedral spline would be formed by combining polyhedrons such that the resulting projections would overlap [152]. In this way, polyhedrons could be added to polyhedral splines on the fly, like knots could be added to univariate B-splines. Two of the most used types of polyhedral splines are the Box spline and the polyhedral simplex spline. It must be noted at this point that in this thesis the term 'simplex spline' refers exclusively to a different multivariate spline concept, that is, Bernstein basis polynomials in barycentric coordinates defined on simplices.

The polyhedral splines were a popular research topic from the middle of the 1980's till the middle of the 1990's. Important contributions were made by Micchelli [131], Hakopian [67], Dahmen [21, 22, 23, 24, 25], de Boor [32, 39, 34, 35], Lai [109] and later Neamtu [151, 152, 153, 154]. The 1992 paper by Dahmen, Micchelli, and Seidel [25] was considered by de Boor [34] to be the 'right' approximation scheme in hindsight. In their paper, Dahmen *et al.* apply the concept of blossoming [175, 176], which uses the Polarizing Principle to convert polynomial behavior into multiaffine behavior, to obtain the control points of Bézier patches. This particular type of polyhedral spline would later be called the Dahmen-Micchelli-Seidel spline or DMS spline. DMS splines use as basis functions Bernstein polynomials in polar form obtained through the blossoming principle.

The polyhedral spline is defined as follows. First let $h \in \mathbb{R}^{n+s}$ be a polyhedron of dimension $n+s$, and let $X \subset \mathbb{R}^s$ be the canonical projection of the vertices of h onto $\mathbb{R}^s$ acting as the spline knots in $\mathbb{R}^s$. Now let the canonical projection of a point $p = (p_1, p_2, \ldots, p_s, \ldots, p_{s+n}) \in \mathbb{R}^{n+s}$ onto $\mathbb{R}^s$ be defined as $p|_{\mathbb{R}^s} := (p_1, p_2, \ldots, p_s)$. The polyhedral spline $B_h(x|X)$ is then defined as the volumetric projection of h onto $\mathbb{R}^s$ as follows:

$$B_h(x|X) := \frac{\mathrm{vol}_n\left(p \in h \;:\; p|_{\mathbb{R}^s} = x\right)}{\mathrm{vol}_{n+s}\left(h\right)}, \tag{2.16}$$

with $\mathrm{vol}_k(A)$ the k-dimensional volume of a set A. In the case that h is a simplex, then the polyhedral spline can be evaluated by Micchelli recurrence [131]. Micchelli recurrence allows a polyhedral simplex spline to be expressed in terms of its lower degree versions as follows:

$$B_h(x|X) := \frac{n+s}{n} \sum_{y \in X} \lambda_y B_h\left(x|X \setminus \{y\}\right), \;\; x \in \mathbb{R}^s, \tag{2.17}$$

with the number λ_y and y chosen such that

$$\sum_{y \in X} \lambda_y = 1, \tag{2.18}$$

$$\sum_{y \in X} \lambda_y y = x. \tag{2.19}$$

As an example of Eq. 2.16 consider the case pictured in Figure 2.7. Here h is a tetrahedron in $\mathbb{R}^3$ and the projection is onto $\mathbb{R}^1$. This means that $s = 1$ and thus that the spline function is univariate. The degree of the spline function is $n = 2$, that is, a quadratic spline function. The knotset X is given by the projection of the vertices of h onto $\mathbb{R}^1$: $X = v_0|_{\mathbb{R}^1}, v_2|_{\mathbb{R}^1}, v_3|_{\mathbb{R}^1}, v_4|_{\mathbb{R}^1}$. In this case Eq. 2.16 can be written as

$$B_h\left(x\left|v_0|_{\mathbb{R}^1}, v_2|_{\mathbb{R}^1}, v_3|_{\mathbb{R}^1}, v_4|_{\mathbb{R}^1}\right.\right) := \frac{\mathrm{vol}_2\left(p \in h \;:\; p|_{\mathbb{R}^1} = x\right)}{\mathrm{vol}_3\left(h\right)} \tag{2.20}$$

It is now clear that the only way to increase the degree of the spline function is by increasing the dimension of the polyhedrons that form it.

Polyhedral splines have lost much of their popularity over the last decade. This is reflected by the fact that at the time of this writing, there have only been a handful of practical applications of polyhedral splines in the literature. Fong and Seidel compared DMS splines with triangular Bézier patches and showed that the DMS splines preserved

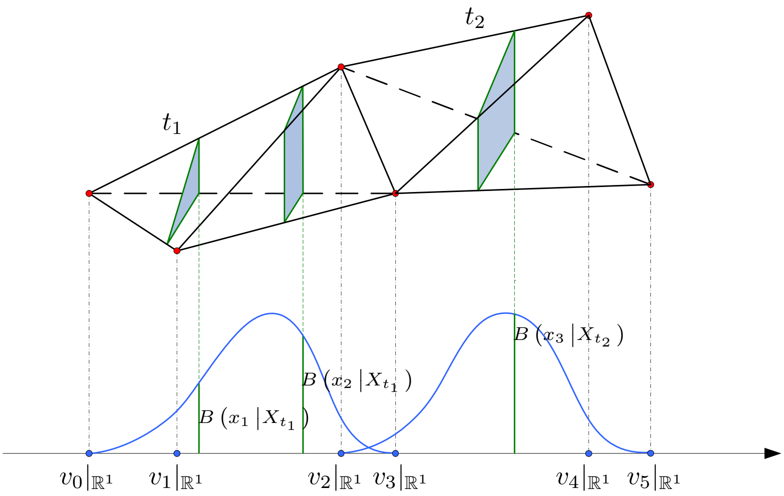


Figure 2.8: A problem with the polyhedral spline: where should the next tetrahedron be located, and how should it be oriented?

continuity when moving a single control point, while the Bézier patches did not [55]. Seidel elegantly demonstrated the basis function property of DMS splines in [191]. In 1996, Pfeifle and Seidel used DMS splines in 2-D to create a geographical model of a large, scattered, set of elevation data [171]. More recently Hua, He and Qin used DMS splines to model complex 3-dimensional datasets [76]. Hansen uses a scheme developed by Neamtu to parameterize 2-dimensional image data [70].

As it turns out, the problem with polyhedral splines is that they simply are too general. Fong and Seidel had observed in 1991 that polyhedral splines "*exhibit the problem that it is a priori unclear how to form linear combinations and how to use them for constructing piecewise polynomial surfaces over arbitrary triangulations*" [55]. That is, they found that it was unclear how to combine multiple polyhedrons of higher dimension into a single geometry of which the projection would result in a specific spline function, see Figure 2.8. This argument was emphasized in a more recent paper by Neamtu as follows: "*Since there is no natural ordering in the Euclidean plane, it should not come as a surprise that the problem of choosing an appropriate collection of knots is far from trivial*" [152]. This is in stark contrast with the univariate B-spline in which the addition of knots is quite trivial. Finally, Grandine had already proved in 1987 that construction and evaluation of polyhedral splines would always remain very inefficient computationally [66]. It is therefore safe to conclude that polyhedral splines, while mathematically elegant, are not a suitable tool for practical scattered data approximation.

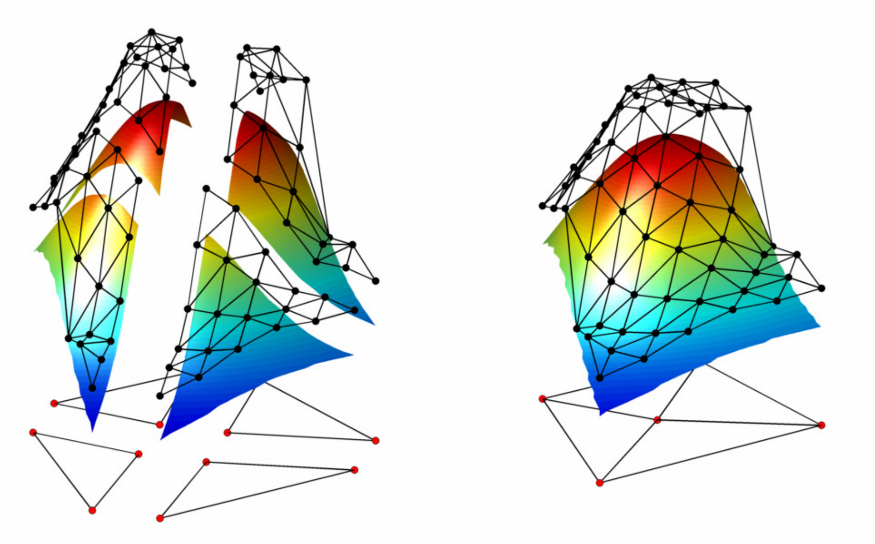


Figure 2.9: The principle of the simplex spline: (left) four individual 2-simplices (triangles) each support a bivariate B-form polynomial and its control coefficient net. By joining the four simplices together into a triangulation, a simplex spline surface is created (right).

2.1.6 Simplex splines

In 1986, while most spline theoreticians were focused on the development of the polyhedral spline, a small group of mathematicians and computer graphics specialists concentrated their efforts on another type of multivariate spline. This multivariate spline type was the multivariate simplex spline, and it was based on the triangular Bernstein-Bézier patches invented by de Casteljau in 1971 [53]. In 1986, Farin published a paper in which he presented, in exquisite detail, the theory of the 2-dimensional simplex spline [52]. A year later, de Boor published an influential paper in which he presented a general multivariate formulation of the B-form [31] using Bernstein polynomials in barycentric coordinates. Initial methods for creating spline interpolations were based on the multivariate B-form and used simplex subdivision schemes like the Clough-Tocher and Powell-Sabin subdivision schemes [34].

The principles of the simplex spline are relatively simple, see Figure 2.9. Geometric n-dimensional structures called simplices each support a single B-form polynomial. The B-form polynomial is a linear combination of Bernstein basis polynomials which are individually scaled by control coefficients called B-coefficients. Many simplices can be joined together into a special configuration called a *triangulation*[4], see Figure 2.10.

---

[4] While the triangle is the 2-simplex, the word triangulation would seem to be specific for the 2-dimensional case. However, in the literature, the word triangulation is used for all simplex spline dimensions.

The simplex spline is defined as follows. First, let $\mathbf{b} \in \mathbb{R}^{n+1}$ be the barycentric coordinate of the point $\mathbf{x} \in \mathbb{R}^n$ with respect to the n-dimensional simplex t. The Bernstein basis polynomial $B^d_\kappa(\mathbf{b})$ of degree d is then given by:

$$B^d_\kappa(\mathbf{b}) := \frac{d!}{\kappa!}\mathbf{b}^\kappa, \tag{2.21}$$

with κ a multi-index. Using Eq. 2.21, the B-form of the multivariate simplex spline of degree d and dimension n for a single simplex t is then defined as follows:

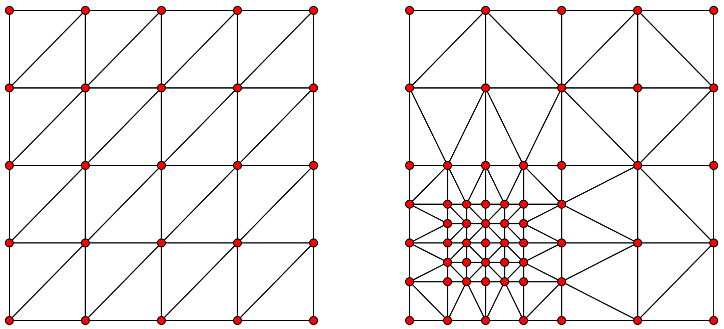
$$p(\mathbf{b}) = \sum_{|\kappa|=d} c_\kappa B^d_\kappa(\mathbf{b}), \tag{2.22}$$

with c_κ the B-coefficients and with $B^d_\kappa(b(\mathbf{x}))$ the Bernstein basis polynomials. The control coefficients completely determine the shape of the spline function. This means that finding a solution to a scattered data approximation problem on a given triangulation reduces to finding an optimal set of B-coefficients for every B-form polynomial. The B-coefficients of the multivariate simplex spline have a spatial location within a simplex. Together they form a structure called the B-coefficient net, see Figure 2.9.

The true power of the simplex spline comes from the versatility of the triangulation. Triangulations can come in many sorts and sizes, from two simplices to millions of simplices in any number of dimensions. Triangulations can be refined locally in areas of higher data complexity by increasing the density of simplices, or be simplified locally to increase computational efficiency. There is one strict rule, however, as to what is a valid triangulation and what is not. In a valid triangulation, simplices are only allowed to meet at their vertices and no two simplices are allowed to overlap, see Figure 2.11. This is one of the properties that sets the simplex spline apart from the polyhedral spline because the polyhedral spline imposes no strict rules on the placing of polyhedrons. One of the most used triangulation types is the Delaunay triangulation, invented by the Russian mathematician Boris Delaunay in 1934 [113].

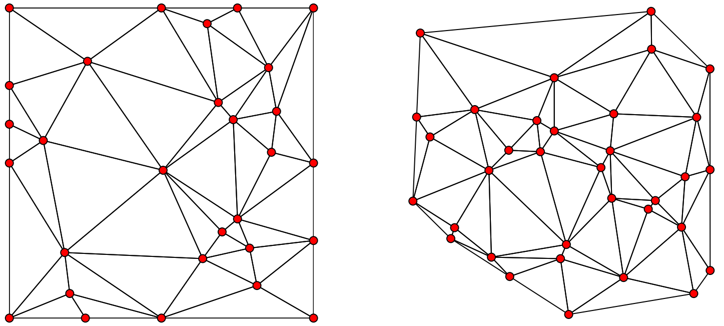
Because the simplex spline completely does away with the concept of knots, mathematicians did not consider it to be the true multivariate generalization of the univariate B-spline. This lack of generality perhaps prevented the simplex splines from becoming a popular research topic in the mathematical community. While at first it seemed that the dependence on triangulations rather than knots would make simplex splines a static and inflexible tool, the concept of the triangulation also prevented the arbitrariness from which the polyhedral spline was suffering. As it turned out, in fact, concepts like triangulation refinement and variable inter-simplex smoothness provided simplex splines with most of the flexibility of polyhedral splines.

From the middle of the 1990's, Lai and Schumaker laid down most of the theoretical foundation of the multivariate simplex spline. In 1996 Lai presented an important result



Uniform rectangular (32 triangles)

Non-uniform rectangular (92 triangles)



Scattered rectangular (43 triangles)

Scattered non-rectangular (55 triangles)

Figure 2.10: 4 different types of triangulations in the 2-plane.

on scattered data approximation with piecewise cubic polynomials on a triangulation [95]. Then, in 1997 Lai reproduced Farin's geometric interpretation [52] of the smoothness conditions and generalized them to higher degrees and dimensions [96]. Together with Schumaker, Lai presented a number of important theories on the approximation power of various spline spaces [100, 101, 102, 26]. Lai, Wenston and Awanou demonstrated that simplex splines are powerful function approximators by using them to approximate the Navier-Stokes equations for fluid flows [106, 107, 4, 5]. More recently, the same authors presented a highly efficient and powerful scheme for scattered data approximation using simplex splines [97, 213, 103, 108, 7, 99, 98]. This scheme allows the formulation of any scattered data approximation problem into a Karush-Kuhn-Tucker (KKT) solution

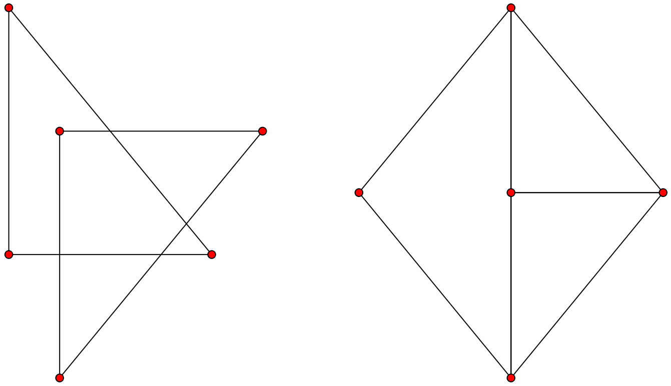


Figure 2.11: Two invalid triangulations: overlapping simplices (left) and asymmetric edge cut (right).

system using Lagrange multipliers to incorporate the smoothness conditions [7]. Awanou presented an iterative method for efficiently solving the KKT system with a guaranteed convergence [6]. Because the size of the KKT system grows proportionally with the total number of simplices and exponentially with the degree and dimension of the splines, it can become large enough to exceed computer memory limits. However, because the simplex splines are local approximators, the triangulation can be decomposed into smaller sets in a process called domain decomposition. Domain decomposition splits up the complete triangulation into a number of sub-triangulations leading to much smaller, partial solution systems [105]. Most theoretical developments from the 1980's and 1990's on simplex splines were consolidated in the excellent textbook by Lai and Schumaker in 2007 [104]. Although the book concentrates on bivariate splines, it contains many theories and insights applicable to simplex splines of any dimension.

Multivariate simplex splines are a truly general interpolation and approximation tool. They can be used to model scattered data on non-rectangular domains in any number of dimensions. Their approximation power is limited only by the complexity and density of the triangulation on which they are defined. Because the multivariate simplex spline has existed in the shadow of the other multivariate spline types, its applications have been relatively limited. Some recent applications of simplex splines are the solving of partial differential equations [75], aerodynamic model identification [43, 42] and volumetric data modeling [86]. Another recent development has been the integration of multivariate simplex splines in a framework for system identification [40, 41].

2.2 The Basis of the Multivariate Simplex Spline

In this section the basic mathematical theory of the multivariate simplex splines is presented. This basic mathematical theory is centered on the polynomial basis functions of the simplex spline, and the coefficients called B-coefficients that are used to scale these basis functions. The theory in this section is limited to polynomials on a single simplex, where a simplex is a special geometric entity that minimally spans a set of dimensions. It will be shown in this section that the basis functions are Bernstein basis polynomials, which form a stable local basis for approximating any continuous function as closely as desired. The stable local basis property will be proved to be a direct result of expressing the basis polynomials in terms of the barycentric coordinate system, a special coordinate system that is native to the simplex.

This section has the following structure. First, in Sec. 2.2.1 the simplex is introduced. It will then be shown in Sec. 2.2.2 that the simplex brings rise to a special local coordinate system called the barycentric coordinate system. In Sec. 2.2.3 the Bernstein basis functions of the multivariate simplex spline are presented. Then, in Sec. 2.2.4 it is proven that Bernstein basis polynomials, when expressed in barycentric coordinates, allow every polynomial of given degree and dimensionality to be written in the B-form. The B-form has an alternative vector representation introduced in Sec. 2.2.5 which will prove to be invaluable in later sections. In Sec. 2.2.6 it will be then shown that the control coefficients of B-form polynomials have a spacial location within a simplex, giving rise to one of the most powerful features unique to the multivariate simplex spline: the B-net. Finally, Sec. 2.2.7 introduces the concept of spline spaces.

2.2.1 The Simplex

A key concept within the mathematical theory of multivariate simplex splines is a general geometric entity called the *simplex*. The simplex is a special type of n-dimensional polytopes that minimally spans a given set of n dimensions. More precisely, the simplex of n-dimensional space, or n-simplex, is the convex hull of a set of $n+1$ non-degenerate points embedded in an $n+k$ dimensional space with $k \geq 0$. As an example of the above, the 0-simplex is the vertex, the 1-simplex is the line, the 2-simplex is the triangle and the 3-simplex is the tetrahedron, see Figure 2.12. The n-simplex can be considered a platonic entity because it resembles a class of n-dimensional objects rather than any particular instance of that class. It is easy to see that there are infinitely many unique instances of any n-simplex; for example, there are infinitely many possible triangles that are all instances of the 2-simplex.

The n-simplex has the following mathematical definition. First, define a singular point, or *vertex*, in an $(n+k)$ dimensional space as follows:

$$\mathbf{v}_{p_i} := (x_1, x_2, \ldots, x_{n+k}) \in \mathbb{R}^{n+k},\ k \geq 0, \tag{2.23}$$

with $p_i \in \mathbb{N}$ the *vertex index*, and with x_j single coordinate components in $(n+k)$-space. The vertex indexer is simply a non-negative integer that uniquely identifies a vertex within the global set of vertices[5].

Now, using Eq. 2.23, let $\mathcal{V}_t$ be a tuple[6] consisting of $n+1$ vertices:

$$\mathcal{V}_t := (\mathbf{v}_{p_0}, \mathbf{v}_{p_1}, \ldots, \mathbf{v}_{p_n}) \in \mathbb{R}^{n+k},\ k \geq 0, \tag{2.24}$$

with

$$p_0 > p_1 > \cdots > p_n. \tag{2.25}$$

The vertices in the tuple $\mathcal{V}_t$ are ordered based on the vertex index, which will prove to be essential when working with triangulations consisting of more than a single simplex as will be explained in Sec. 2.3.5.

The n-simplex t is defined as the *convex hull* of $\mathcal{V}_t$ from Eq. 2.24 as follows:

$$t := \langle \mathcal{V}_t \rangle \in \mathbb{R}^n. \tag{2.26}$$

A key requirement for the vertices of the n-simplex t is that they must be non-degenerate, where non-degeneracy is defined as follows:

Definition 2. *A set of $n+1$ vertices $\mathcal{V}_t$ is considered non-degenerate if, and only if, the convex hull of $\mathcal{V}_t$ is of dimension n.*

That is, if a vertex $\mathbf{v}_{p_i}$ within the vertex set $\mathcal{V}_t$ of the simplex t can be removed without changing the dimension of t, then $\mathbf{v}_{p_i}$ is a degenerate vertex.

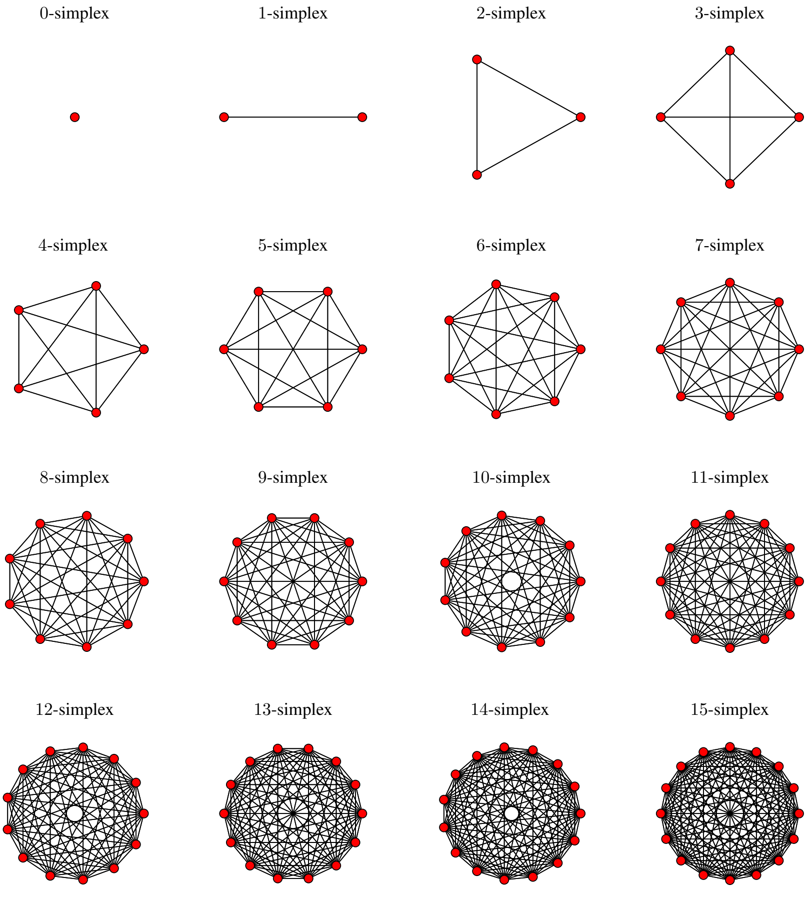
In Figure 2.12 the simplices of dimension 0 to 15 are drawn by projecting their vertices on the 2-plane using a skew orthogonal projection. Visualizing and interpreting simplices of dimension 4 and higher is not an easy task, as we humans are used to thinking in 3-dimensions. For example, it may not seem apparent that the n-simplices in Figure 2.12 are all regular, which means that the Euclidean distance between any two of their vertices is equal in n-space. The n-simplex has a fundamental property, however, that aids its visualization and interpretation. This fundamental property of the n-simplex is that its geometric elements, also called *faces*, are themselves simplices of all dimensions lower than n. Simplices have a recursive structure! For example, the triangle (2-simplex) contains three lines (1-simplex) and three vertices (0-simplex). In the same way, the tetrahedron (3-simplex) contains four triangles, six lines and four vertices.

In Table 2.3 the total number of faces of increasing dimension of the simplices of dimension 0 to dimension 8 are listed. Observe that the elements in Table 2.3 are, in fact,

---

[5] While not explained at this point, the global set of vertices are all vertices for all simplices that form the complete spline function. A detailed discussion of the need for global indexing as well as its uses takes place in Sec. 2.3 and Sec. 3.3.

[6] A tuple is a special set in which the elements are ordered.



0-simplex

1-simplex

2-simplex

3-simplex

4-simplex

5-simplex

6-simplex

7-simplex

8-simplex

9-simplex

10-simplex

11-simplex

12-simplex

13-simplex

14-simplex

15-simplex

Figure 2.12: Projections on the 2-plane of the regular simplices of dimension 0 to 15.

Table 2.3: Elements of simplices

Dimension n	0-faces	1-faces	2-faces	3-faces	4-faces	5-faces	6-faces	7-faces	8-faces
0	1	0	0	0	0	0	0	0	0
1	2	1	0	0	0	0	0	0	0
2	3	3	1	0	0	0	0	0	0
3	4	6	4	1	0	0	0	0	0
4	5	10	10	5	1	0	0	0	0
5	6	15	20	15	6	1	0	0	0
6	7	21	35	35	21	7	1	0	0
7	8	28	56	70	56	28	8	1	0
8	9	36	84	126	126	84	36	9	1

the elements of the first eight rows of Pascal's Triangle[7]. In general, it is found that the total number of m-faces of an n-simplex is:

$$N_m = \left(\begin{array}{c} n+1 \\ m+1 \end{array} \right) \tag{2.27}$$

which is the equation for the $(n+1)^{th}$ row and $(m+1)^{th}$ column in Pascal's triangle.

One important type of face of an n-simplex t is the $n-1$ face, called the *edge facet* and denoted $\tilde{t}$. An edge facet shares n vertices with t, and therefore is itself a simplex of dimension $n-1$. It is easy to check that any n-simplex has $n+1$ edge facets. Therefore, the following indirect definition of the i^{th} edge facet can be given:

$$\tilde{t}_i := \langle \mathcal{V}_t \setminus \mathbf{v}_{p_i} \rangle \in \mathbb{R}^n, \; i = 0, 1, \ldots, n. \tag{2.28}$$

In words, Eq. 2.28 states that any edge facet $\tilde{t}_i$ can be defined implicitly in terms of the single vertex $\mathbf{v}_{p_i}$ in the n-simplex t which is not in $\tilde{t}_i$. This observation, made by de Boor in [31], will prove to be instrumental in the definition of the continuity conditions that govern the continuity between simplex polynomials on neighboring simplices.

Using the recursive simplex structure, the visualization and interpretation of higher dimensional simplices can be simplified. Consider for example a 4-simplex, or *pentachoron*, which has five tetrahedrons as edges. Figure 2.13 shows a pentachoron with four vertices in the 3-plane, and one vertex in the 4-plane. The 'base plane' of this pentachoron thus is an ordinary tetrahedron; it exists entirely in 3-space. The other four tetrahedron edges exist partly in 4-space. In Figure 2.14 the five edge tetrahedrons of a pentachoron are highlighted.

---

[7] Pascal's triangle makes a number of unexpected entries throughout multivariate simplex spline theory, perhaps suggesting some mysterious connection. While attributed posthumously to the French mathematician Blaise Pascal (1623-1662), Pascal's triangle has been known to many mathematicians throughout written history, with the first surviving reference dating back to the Persian mathematician al Karajï sometime around 1100 AD [57]. The popularity of Pascal's triangle is probably due to its close links with geometry, which also explains its (less than mysterious) connection with simplex spline theory.

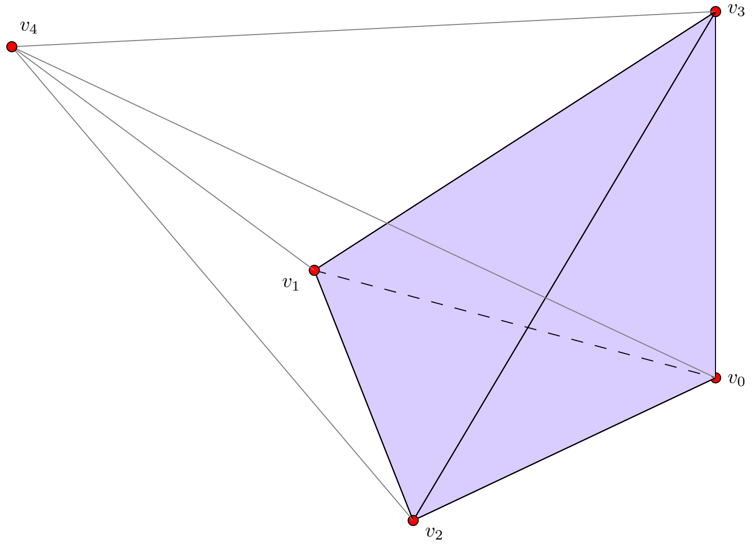


Figure 2.13: The pentachoron, or 4-simplex, with its tetrahedral base plane highlighted.

It is clear from the figure that each set of four vertices forms an edge tetrahedron. Now consider a 5-simplex such as that pictured in Figure 2.12. What the tetrahedron was to the pentachoron, the pentachoron is to the 5-simplex, that is, a 5-simplex has six pentachorons as edges. In the same line of reasoning, a line is to a tetrahedron what a tetrahedron is to a 5-simplex. Conversely, a 5-simplex is to a 7-simplex what a line is to a tetrahedron; a rather insignificant part in terms of complexity as compared to the whole. This reasoning not only aids the interpretation, but also helps one to appreciate the true complexity of these higher dimensional structures.

A useful entity related to any n-simplex is the normalized simplex vertex matrix $\mathbf{A}_t$. This matrix contains the vertex coordinates of t normalized with the coordinates of the first vertex as follows:

$$\mathbf{A}_t = \left[\begin{array}{cccc} \mathbf{v}_{p_1} - \mathbf{v}_{p_0} & \mathbf{v}_{p_2} - \mathbf{v}_{p_0} & \cdots & \mathbf{v}_{p_n} - \mathbf{v}_{p_0} \end{array} \right] \in \mathbb{R}^{n \times n} \tag{2.29}$$

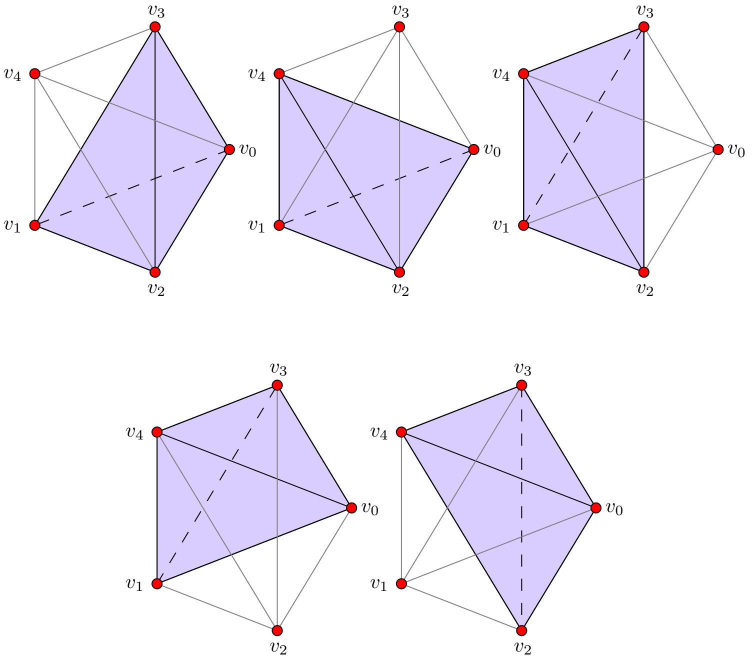


Figure 2.14: Highlighting the five tetrahedrons that form the edges of a pentachoron, or 4-simplex.

The normalized simplex vertex matrix from Eq. 2.29 can be used to calculate the Euclidean volume of an n-simplex t as follows:

$$Vol(t) = \frac{1}{n!} det\ \mathbf{A}_t \tag{2.30}$$

2.2.2 Barycentric Coordinates

In 1827, the famous German mathematician August Ferdinand Möbius[8] (1790-1868) introduced a new coordinate system he named the *barycentric coordinate system* [136]. Möbius used the term barycentric to indicate the relationship of the new coordinate system with the center of mass, or barycenter, of a set of point masses located at the vertices of (fixed) triangles and tetrahedrons. Möbius found the new coordinate system to be

---

[8] Möbius is perhaps best known for his invention of the Möbius strip, which is a geometry embedded in Euclidean 3-space, and which only has one boundary component.

homogeneous [136, 197], that is, invariant of scalar multiplication. This can be explained by considering the barycenter of the set of point masses; multiplication of these masses with a common multiplication factor will not change the location of the barycenter. Möbius proved that any coordinate in a 2-dimensional or 3-dimensional Euclidean plane can be expressed as a weighted vector sum of the vertices of respectively a triangle or tetrahedron. While Möbius restricted his discussion of barycentric coordinates to the 2-dimensional and 3-dimensional Euclidean planes, the concept of the barycentric coordinates is easily generalized to the n-dimensional Euclidean plane.

The derivation of the barycentric coordinates starts with a point $\mathbf{x}$ in the n-dimensional Euclidean plane:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \ \in \mathbb{R}^n. \tag{2.31}$$

The point $\mathbf{x}$ can be expressed in terms of the vertices of an n-simplex as follows:

$$\mathbf{x} = \sum_{i=0}^{n} b_i \mathbf{v}_{p_i}, \tag{2.32}$$

with $\mathbf{v}_{p_i} \in \mathbb{R}^n$ from Eq. 2.23 and with $b_i \in \mathbb{R}$ scalar weights. The barycentric coordinate of $\mathbf{x}$ is the vector of vertex weights $\mathbf{b}$ given by:

$$b = (b_0, b_1, \ldots, b_n) \ \in \mathbb{R}^{n+1}, \tag{2.33}$$

In the following we define the implicit transformation of the Cartesian coordinate $\mathbf{x}$ from Eq. 2.31 into the barycentric coordinate $\mathbf{b}$ from Eq. 2.33 as follows:

$$b(\mathbf{x}) := \mathbf{b} \ \in \mathbb{R}^{n+1}, \tag{2.34}$$

While the definition of the barycentric coordinates from Eq. 2.32 is complete, an additional expression in the form of the normalization property is required to explicitly calculate $b(\mathbf{x})$ given a set of simplex vertices:

$$\sum_{i=0}^{n} b_i = 1. \tag{2.35}$$

The normalization property of the barycentric coordinates effectively reduces the number of degrees of freedom of the barycentric coordinate transformation by one, since any single component of $\mathbf{b}$ can now be expressed in terms of the remaining components:

$$b_0 = 1 - \sum_{i=1}^{n} b_i. \tag{2.36}$$

An implicit expression for the Cartesian-to-barycentric coordinate transformation was already given in Eq. 2.34. At this point, all the theory is in place to allow the derivation of an explicit form of this transformation. Starting with Eq. 2.32 together with the normalization property Eq. 2.36 we get:

$$\begin{aligned}
\mathbf{x} &= \sum_{i=0}^{n} b_i \mathbf{v}_{p_i} \\
&= b_0 \mathbf{v}_{p_0} + \sum_{i=1}^{n} b_i \mathbf{v}_{p_i} \\
&= \left(1 - \sum_{i=1}^{n} b_i\right) \mathbf{v}_{p_0} + \sum_{i=1}^{n} b_i \mathbf{v}_{p_i} \\
&= \mathbf{v}_{p_0} + \sum_{i=1}^{n} \left(b_i \mathbf{v}_{p_i} - b_i \mathbf{v}_{p_0}\right) \\
&= \mathbf{v}_{p_0} + \sum_{i=1}^{n} b_i \left(\mathbf{v}_{p_i} - \mathbf{v}_{p_0}\right).
\end{aligned} \tag{2.37}$$

So, finally:

$$\mathbf{x} - \mathbf{v}_{p_0} = \sum_{i=1}^{n} b_i \left(\mathbf{v}_{p_i} - \mathbf{v}_{p_0}\right). \tag{2.38}$$

Notice that the right hand side of this equation can be written in the following vector form:

$$\mathbf{x} - \mathbf{v}_{p_0} = \begin{bmatrix} \mathbf{v}_{p_1} - \mathbf{v}_{p_0} & \mathbf{v}_{p_2} - \mathbf{v}_{p_0} & \cdots & \mathbf{v}_{p_n} - \mathbf{v}_{p_0} \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{2.39}$$

This expression can be simplified by substitution of the normalized simplex vertex matrix from Eq. 2.29:

$$\mathbf{x} - \mathbf{v}_{p_0} = \mathbf{A}_t \cdot \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{2.40}$$

The vector of barycentric coordinates is obtained by multiplying the left and right hand side with $\mathbf{A}_t^{-1}$:

$$\mathbf{A}_t^{-1} \cdot (\mathbf{x} - \mathbf{v}_{p_0}) = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{2.41}$$

Note that $\mathbf{A}_t$ is invertible if the vectors $v_{p_i} - v_{p_0}$ are linearly independent, which is the case when the vertices v_{p_i} are non-degenerate according to **Definition** 2. If $\mathbf{A}_t$ is invertible then the Cartesian-to-barycentric coordinate system transformation is a linear, one-to-one transformation. This in turn implies that any Cartesian coordinate in the Euclidean n-plane has a unique barycentric coordinate with respect to a *given* n-simplex. This also holds for points that are located outside the simplex with respect to which the barycentric coordinates are calculated. In fact, the barycentric coordinates provide a simple method for determining whether a point is inside or outside an n-simplex t:

$$b_i > 0 \quad \longrightarrow \quad \mathbf{x} \in t,\ \forall i \tag{2.42}$$

$$b_i < 0 \quad \longrightarrow \quad \mathbf{x} \notin t,\ 0 \leq i \leq n \tag{2.43}$$

That is, if a point is located inside a simplex, all its barycentric coordinate components with respect to that simplex are larger than or equal to zero. If a point is located outside a simplex at least one, and possibly all, barycentric coordinate components are smaller than zero.

In Figure 2.15 the principle of barycentric coordinates is demonstrated on a single triangle. There are some important things to note from the figure. First, all barycentric coordinate components of points inside the triangle are positive, which is according to Eq. 2.43. After closer inspection, it can be seen that points located on the triangle edges have 2 nonzero components, while points located at the vertices have only 1 nonzero component. This is not a coincidence; in general, points located on the m-face of an n-simplex face will have $m + 1$ nonzero barycentric components.

A numerical example of a Cartesian-to-barycentric coordinate transformation will now be given.

Example 1 (Cartesian to barycentric transformation)**.** *The Cartesian-to-barycentric coordinate transformation for a point* $\mathbf{x} \in \mathbf{R}^2$ *with respect to a triangle t.*

Let $\mathbf{x} = (1, 0.5)$ *and let t be a triangle with a vertex set* $\mathcal{V}_t = \{\mathbf{v}_{12}, \mathbf{v}_{13}, \mathbf{v}_{34}\}$ *with* $p_0 = 12$*,* $p_1 = 13$*, and* $p_2 = 34$*. The vertices have coordinates* $\mathbf{v}_{12} = (0.5, 0), \mathbf{v}_{13} = (2, 0)$*, and* $\mathbf{v}_{34} = (0, 2)$*. The* 2×2 *normalized simplex vertex matrix from Eq. 2.29 can now be constructed:*

$$\begin{aligned}
\mathbf{A}_t &= \begin{bmatrix} \mathbf{v}_{p_1} - \mathbf{v}_{p_0} & \mathbf{v}_{p_2} - \mathbf{v}_{p_0} \end{bmatrix} \\
&= \begin{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 2 \end{bmatrix} - \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} \end{bmatrix} \\
&= \begin{bmatrix} 1.5 & -0.5 \\ 0 & 2 \end{bmatrix}.
\end{aligned}$$

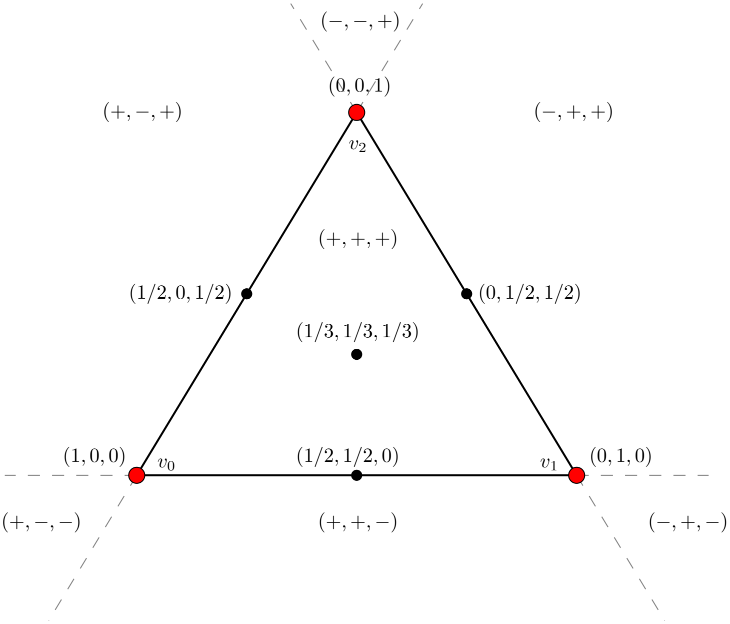


Figure 2.15: Barycentric coordinates of some points with respect to a triangle, together with regions of positivity and negativity of barycentric coordinate components.

Inverting $\mathbf{A}_t$ *and using Eq. 2.41 we get for the last* 2 *elements of* $\mathbf{b}$*:*

$$\begin{aligned}
\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} &= \mathbf{A}_t^{-1} \cdot (\mathbf{x} - \mathbf{v}_{p_0}) \\
&= \begin{bmatrix} 0.6667 & 0.1667 \\ 0 & 0.5 \end{bmatrix} \cdot \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \\
&= \begin{bmatrix} 0.4167 \\ 0.25 \end{bmatrix}.
\end{aligned}$$

Finally, using the normalization property Eq. 2.36 we get:

$$\begin{aligned}\mathbf{b} &= (1-(0.4167+0.25), 0.4167, 0.25)\\ &= (0.3333, 0.4167, 0.25)\,.\end{aligned}$$

All barycentric coordinate components are larger than zero, so according to Eq. 2.43 the point $\mathbf{x}$ *is located inside* t.

■

The barycentric coordinate system may at first seem to be an unnecessary excursion into some obscure corner of mathematics. As it will turn out, however, the barycentric coordinate system is absolutely essential in the definition of the polynomial basis functions that constitute the multivariate simplex spline functions.

2.2.3 Bernstein basis polynomials

A simplex spline function consists of a set of neighboring simplices on each of which is defined a single *simplex polynomial*. A simplex polynomial is defined locally on a single simplex, and can be considered a *Bernstein polynomial*[9]. Being a Bernstein polynomial, a simplex polynomial consists of a linear combination of *Bernstein basis polynomials*. In this section, the Bernstein basis polynomials of the multivariate simplex spline will be derived from the well-known multinomial theorem.

First, let $\mathbf{b} = (b_0, b_1, \ldots, b_n)$ be a coordinate in $\mathbb{R}^{n+1}$. The multinomial theorem then states the following for polynomials of degree d in $\mathbf{b}$:

$$(b_0 + b_1 + \cdots + b_n)^d = \sum_{\kappa_0+\kappa_1+\cdots+\kappa_n=d} \frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} \tag{2.44}$$

with the exponential $d \in \mathbb{N}$ the polynomial degree. This expression can be simplified by the introduction of a new entity called the *multi-index*. The multi-index κ of dimension n is a tuple defined as follows:

$$\kappa := (\kappa_0, \kappa_1, \ldots, \kappa_n) \in \mathbb{N}^{n+1}. \tag{2.45}$$

The 1-norm of the multi-index κ is given by:

$$|\kappa| = \kappa_0 + \kappa_1 + \cdots + \kappa_n = d, \;\; d \geq 0, \tag{2.46}$$

---

[9] Bernstein polynomials are named after Russian mathematician Sergei Natanovich Bernstein (1880-1968) who used them to prove the Weierstrass approximation theorem. The Weierstrass approximation theorem states that every continuous function can be approximated as closely as desired by a polynomial function on a closed interval, and as such has many important theoretical and practical applications.

The factorial of the multi-index κ is:

$$\kappa! = \kappa_0!\kappa_1!\cdots\kappa_n!. \tag{2.47}$$

Using Eq. 2.47, the multinomial coefficient from Eq. 2.44 can be simplified as follows:

$$\frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} = \frac{d!}{\kappa!}. \tag{2.48}$$

The polynomial part from Eq. 2.44 can also be simplified using multi-index notation:

$$b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} = \mathbf{b}^{\kappa} \tag{2.49}$$

Combining Eq. 2.48 with Eq. 2.49 leads to the simplified expression for the multinomial theorem using multi-index notation:

$$(b_0 + b_1 + \cdots + b_n)^d = \sum_{|\kappa|=d} \frac{d!}{\kappa!}\mathbf{b}^{\kappa}, \tag{2.50}$$

Note that the total number of polynomial terms in Eq. 2.50 is determined by the length of the sum, which in turn is determined by the total number of valid permutations of κ. The total number of valid permutations of κ for a given dimension n and polynomial degree d is $\hat{d}$, which is defined as follows:

$$\hat{d} = \left(\begin{array}{c} d+n \\ d \end{array} \right) = \frac{(d+n)!}{n!d!}. \tag{2.51}$$

This expression has been taken from literature, see [31] and [7]. Interestingly, the total number of valid permutations of κ for a given dimension n and polynomial degree d is equal to the total number of elements on the d^{th} level of Pascal's $(n+1)$-simplex . This is no coincidence, as the values on the d^{th} level of Pascal's n-simplex are the values of the multinomial coefficient for a given n and d. Logically, there must be as many multinomial coefficient values as there are polynomial terms and thus total number of valid permutations of κ. For example, for $n = 2$ the total number of permutations of κ for $d = 1, 2, 3, 4$ is equal to the total number of elements on respectively the first, second, third and fourth levels of Pascal's tetrahedron: $3, 6, 10$ and 15.

The Bernstein basis polynomial can now be defined as follows:

$$B_{\kappa}^{d}(\mathbf{b}) := \frac{d!}{\kappa!}\mathbf{b}^{\kappa}. \tag{2.52}$$

The Bernstein basis polynomial of the multivariate simplex spline is itself a polynomial of degree d, scaled by the constant multinomial coefficient. Using Eq. 2.52 the multinomial expression from Eq. 2.50 can be simplified as follows:

$$(b_0 + b_1 + \cdots + b_n)^d = \sum_{|\kappa|=d} B^d_\kappa(\mathbf{b}), \tag{2.53}$$

where the expression on the right hand side is a Bernstein polynomial. The multinomial theorem from Eq. 2.50 can be expressed in terms of the barycentric coordinates, leading to a powerful new concept called the *partition of unity* of the Bernstein basis polynomials. If $\mathbf{b} = (b_0, b_1, \ldots, b_n)$ is a barycentric coordinate, then according to the normalization property from Eq. 2.35 we have $b_0 + b_1 + \cdots + b_n = 1$. The normalization property thus allows a slight, but very significant, reformulation of Eq. 2.50 into:

$$1 = \sum_{|\kappa|=d} B^d_\kappa(\mathbf{b}). \tag{2.54}$$

The partition of unity quality simply states that every Bernstein basis polynomial contributes to the total value of 1, at *every* location within an n-simplex, see e.g. Figure 2.17. In the following the Bernstein polynomial will sometimes be expressed in terms of Cartesian coordinates, rather than barycentric coordinates. In that case, the implicit Cartesian to barycentric coordinate transformation operator $b(\mathbf{x})$ is used to reformulate Eq. 2.54 into:

$$1 = \sum_{|\kappa|=d} B^d_\kappa(b(\mathbf{x})). \tag{2.55}$$

Every Bernstein basis polynomial has a unique maximum at a specific location inside the n-simplex. The location of this maximum $\mathbf{b}_{max_\kappa}$ is:

$$\mathbf{b}_{max_\kappa} = \frac{\kappa}{d}, \ \ |\kappa| = d. \tag{2.56}$$

Before continuing with the definition of the B-form in the next section, one additional entity needs to be introduced in the form of the tuple of valid permutations of the multi-index κ. Let $\mathcal{K}^d_n$ be this tuple for polynomial degree d and dimension n as follows:

$$\mathcal{K}^d_n = (\kappa)_{|\kappa|=d}\,. \tag{2.57}$$

The tuple $\mathcal{K}^d_n$ thus contains all permutations of κ that have all non-negative components and that sum up to d. The cardinality of $\mathcal{K}^d_n$ follows immediately from Eq. 2.51:

$$\left|\mathcal{K}^d_n\right| = \hat{d}. \tag{2.58}$$

A complete example of the construction of a Bernstein polynomial from its basis functions is now given.

Example 2 (Using the Multinomial Theorem)**.** *Expansion of the multinomial theorem for a third degree trivariate basis polynomial.*

The procedure for the expansion starts by constructing the tuple $\mathcal{K}_n^d$ *with* $n = 2$ *and* $d = 3$ *using Eq. 2.57:*

$$\mathcal{K}_2^3 = \{(3,0,0),(2,1,0),(2,0,1),(1,2,0),(1,1,1),(1,0,2),(0,3,0),(0,2,1),(0,1,2),(0,0,3)\}$$

This tuple can now be used to expand the multinomial equation from Eq. 2.54 as follows:

$$\begin{aligned}
(b_0 + b_1 + b_2)^3 &= \sum_{|\kappa|=d} B_\kappa^3(\mathbf{b}) \\
&= B_{300}^3(\mathbf{b}) + B_{210}^3(\mathbf{b}) + B_{201}^3(\mathbf{b}) + B_{120}^3(\mathbf{b}) + B_{111}^3(\mathbf{b}) + \\
&\quad B_{102}^3(\mathbf{b}) + B_{030}^3(\mathbf{b}) + B_{021}^3(\mathbf{b}) + B_{012}^3(\mathbf{b}) + B_{003}^3(\mathbf{b}) \\
&= \frac{3!}{3!0!0!} \cdot b_0^3 b_1^0 b_2^0 + \frac{3!}{2!1!0!} \cdot b_0^2 b_1^1 b_2^0 + \frac{3!}{2!0!1!} \cdot b_0^2 b_1^0 b_2^1 + \frac{3!}{1!2!0!} \cdot b_0^1 b_1^2 b_2^0 + \\
&\quad \frac{3!}{1!0!2!} \cdot b_0^1 b_1^1 b_2^1 + \frac{3!}{0!3!0!} \cdot b_0^1 b_1^0 b_2^2 + \frac{3!}{0!3!0!} \cdot b_0^0 b_1^3 b_2^0 + \frac{3!}{0!2!1!} \cdot b_0^0 b_1^2 b_2^1 + \\
&\quad \frac{3!}{0!1!2!} \cdot b_0^0 b_1^1 b_2^2 + \frac{3!}{0!0!3!} \cdot b_0^0 b_1^0 b_2^3 \\
&= b_0^3 + 3b_0^2 b_1 + 3b_0^2 b_2 + 3b_0 b_1^2 + 6b_0 b_1 b_2 + 3b_0 b_2^2 + b_1^3 + 3b_1^2 + 3b_1 b_2^2 + b_2^3
\end{aligned}$$

■

In Figure 2.16, the 15 individual polynomial basis functions that together form the 4^{th} degree polynomial $\sum_{|\kappa|=4} B_\kappa^4(\mathbf{b})$ on a single 2-simplex are shown. The figure clearly shows that the Bernstein basis polynomials have a localized influence on the complete simplex polynomial. In Figure 2.17 the partition of unity property of the polynomial basis functions is demonstrated. In the figure, individual polynomial basis function terms are added according to the sum in $\sum_{|\kappa|=4} B_\kappa^4(\mathbf{b})$. This sum of terms reaches a constant value 1 (unity) over the entire simplex after the last basis polynomial has been added.

2.2.4 The B-form

In his 1987 theoretical paper, Carl de Boor provided a proof of the stable local basis property of Bernstein basis polynomials, thereby laying down the mathematical foundations of what was to become the multivariate simplex spline [31]. The stable local basis allows every polynomial of degree d to be expressed as a unique linear combination of Bernstein basis polynomials. The B-form[10] in turn provides an elegant notation for the linear combination

---

[10] At current, there seems to be no consensus on the actual meaning of the capital ‘B’ in the B-form. Carl de Boor, who was the first author in the literature to use the term B-form notes that ‘*For the sake of brevity, and since there are several people and ideas responsible, I am proposing here the term B-form [...] for what would, more properly, be called the barycentric-Bernstein-de Casteljau-Bézier-Farin-$\cdots$-form.*’

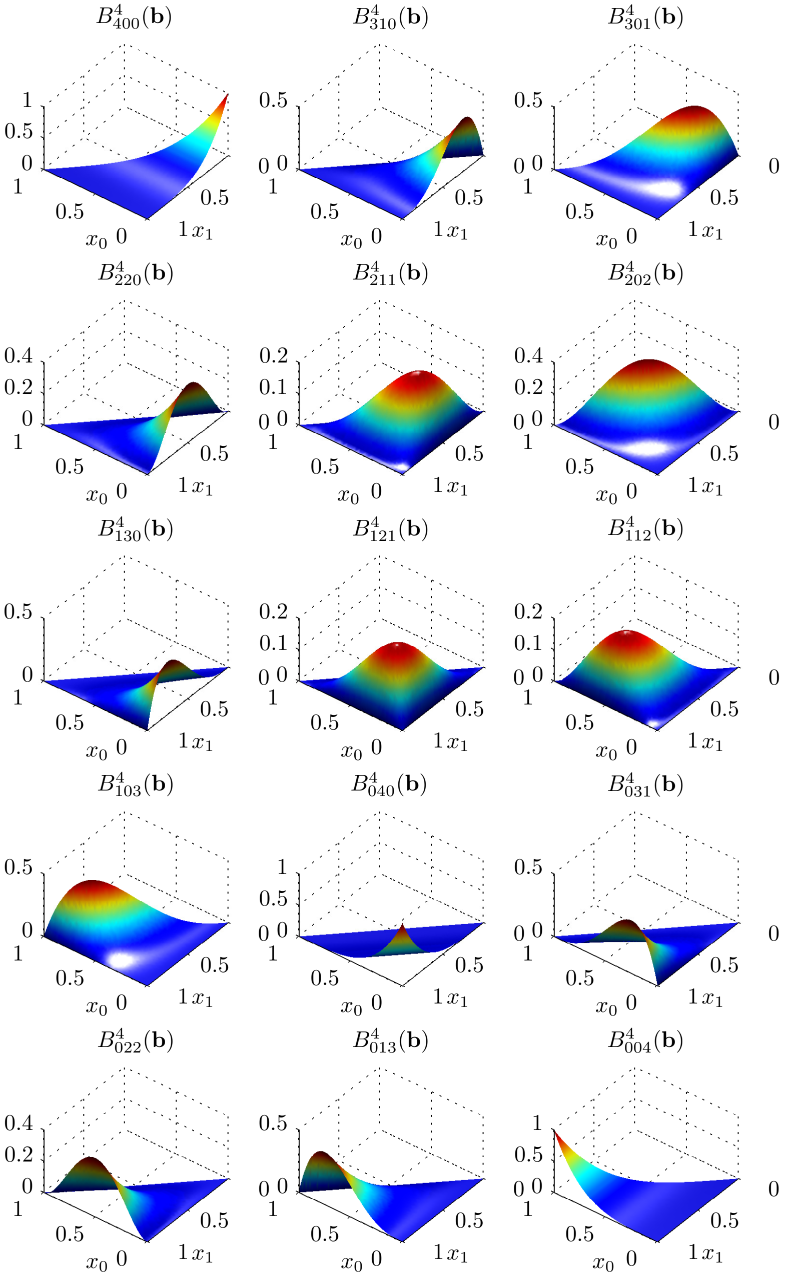


Figure 2.16: The 15 individual basis functions of a 4^{th} degree polynomial on a single 2-simplex.

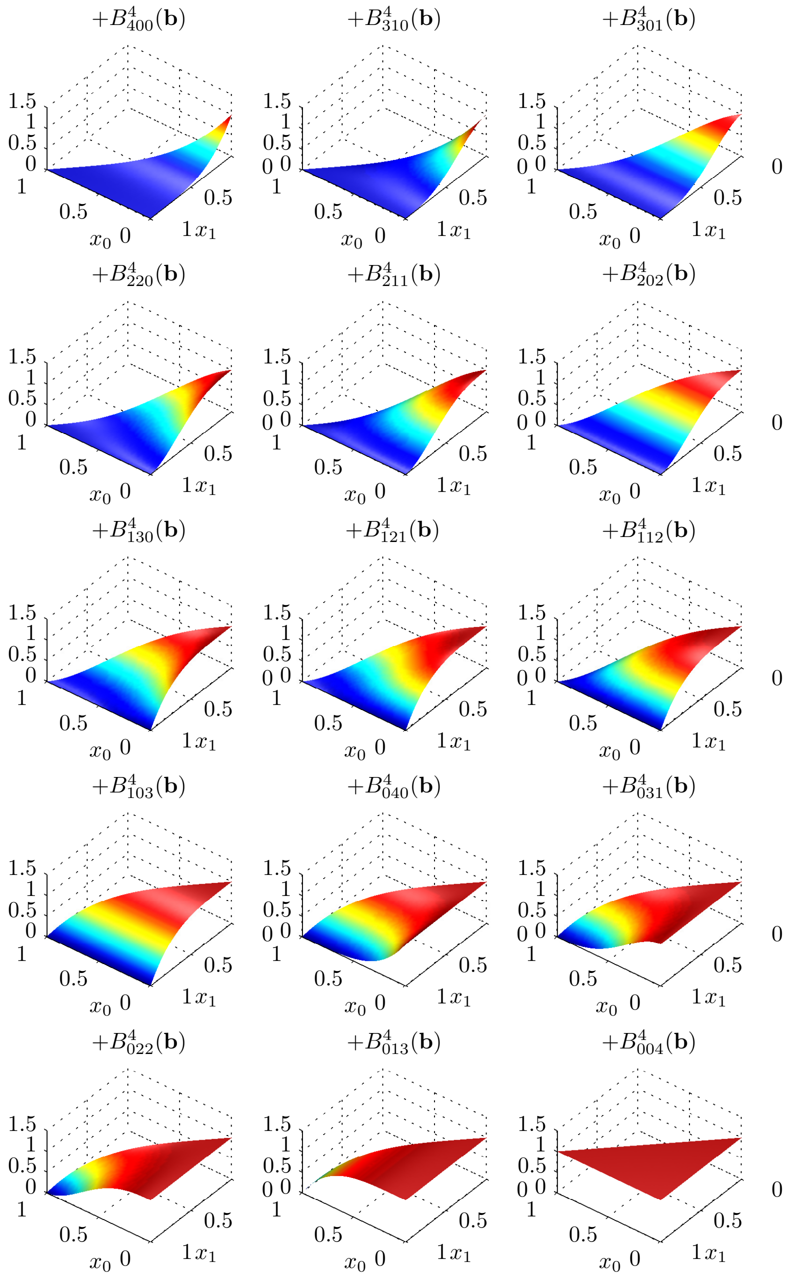


Figure 2.17: Construction of the complete 4^{th} degree polynomial $\sum_{|\kappa|=4} B^4_\kappa(\mathbf{b})$ demonstrates the partition of unity property of the 15 individual basis polynomials.

of Bernstein basis functions on a single n-simplex. De Boor's general proof for n-variate basis functions has been repeated for some special cases in the literature, see e.g. [104] for bivariate basis functions and [4] for trivariate basis functions. Because this proof is fundamental to the validity of the multivariate spline approximation scheme it is repeated here for the n-variate case using the notation from [4] and [104] .

Theorem 1. *Any polynomial $p(\mathbf{x})$ of degree d in the space of polynomials $\mathcal{P}_d$ can be written in the B-form as follows:*

$$p(\mathbf{x}) = \sum_{|\kappa|=d} c_\kappa B^d_\kappa(b(\mathbf{x})). \tag{2.59}$$

Proof. The theorem implies that the set of basis polynomials $\mathcal{B}^d = \left\{B^d_\kappa(b(\mathbf{x}))\right\}_{|\kappa|=d}$ is a basis for $\mathcal{P}_d$. Therefore, we must have that every polynomial $x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n}$ for $0 \leq |\gamma| \leq d$ is in the span of $\mathcal{B}^d$, with $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ a multi-index. The proof is based on the partition of unity property of the Bernstein basis polynomials from Eq. 2.55 and the definition of the barycentric coordinates from Eq. 2.32.

Using Eq. 2.32 we can write for the variable x_i:

$$x_i = b_0 \mathbf{v}_{0_i} + b_1 \mathbf{v}_{1_i} + \cdots + b_n \mathbf{v}_{n_i}. \tag{2.60}$$

The partition of unity property holds for Bernstein polynomials of all degrees, which means that multiplying with $\sum_{|\kappa|=d-1} B^{d-1}_\kappa(b(\mathbf{x}))$ is equal to multiplying by 1:

$$\begin{aligned} x_i &= b_0 \mathbf{v}_{0_i} + b_1 \mathbf{v}_{1_i} + \cdots + b_n \mathbf{v}_{n_i} \\ &= \left(b_0 \mathbf{v}_{0_i} + b_1 \mathbf{v}_{1_i} + \cdots + b_n \mathbf{v}_{n_i}\right) \left(\sum_{|\kappa|=d-1} B^{d-1}_\kappa(b(\mathbf{x}))\right) \\ &= \sum_{|\kappa|=d} \left(\mathbf{v}_{0_i} + \mathbf{v}_{1_i} + \cdots + \mathbf{v}_{n_i}\right) B^d_\kappa(b(\mathbf{x})) \\ &= \sum_{|\kappa|=d} a_\kappa B^d_\kappa(b(\mathbf{x})), \end{aligned} \tag{2.61}$$

with a_κ a constant independent of $\mathbf{b}$. We have now proved that all individual components, or variables x_i are in span $\left(\mathcal{B}^d\right)$. Using the same line of reasoning we can prove that $x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n} \in$ span $\left(mathcalB^d\right)$ by again using the partition of unity property to raise the total degree of the right hand side to d as follows:

$$\begin{aligned} x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n} &= \left(b_0 \mathbf{v}_{0_1} + b_1 \mathbf{v}_{1_1} + \cdots + b_n \mathbf{v}_{n_1}\right)^{\gamma_1} \left(b_0 \mathbf{v}_{0_2} + b_1 \mathbf{v}_{1_2} + \cdots + b_n \mathbf{v}_{n_2}\right)^{\gamma_2} \cdots \\ & \left(b_0 \mathbf{v}_{0_n} + b_1 \mathbf{v}_{1_n} + \cdots + b_n \mathbf{v}_{n_n}\right)^{\gamma_n} \cdot \sum_{|\kappa|=d-|\gamma|} B^{d-|\gamma|}_\kappa(b(\mathbf{x})) \end{aligned}$$

We will now proceed with moving the first polynomial $x_1^{\gamma_1}$ inside the sum:

$$\begin{aligned} x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n} &= (b_0 \mathbf{v}_{0_2} + b_1 \mathbf{v}_{1_2} + \cdots + b_n \mathbf{v}_{n_2})^{\gamma_2} \cdots (b_0 \mathbf{v}_{0_n} + b_1 \mathbf{v}_{1_n} + \cdots + b_n \mathbf{v}_{n_n})^{\gamma_n} \cdot \\ &\quad \sum_{|\kappa|=d-|\gamma|+\gamma_1} (\mathbf{v}_{0_1} + \mathbf{v}_{1_1} + \cdots + \mathbf{v}_{n_1})^{\gamma_1} B_\kappa^{d-|\gamma|+\gamma_1}(b(\mathbf{x})) \\ &= (b_0 \mathbf{v}_{0_2} + b_1 \mathbf{v}_{1_2} + \cdots + b_n \mathbf{v}_{n_2})^{\gamma_2} \cdots (b_0 \mathbf{v}_{0_n} + b_1 \mathbf{v}_{1_n} + \cdots + b_n \mathbf{v}_{n_n})^{\gamma_n} \cdot \\ &\quad \sum_{|\kappa|=d-|\gamma|+\gamma_1} a_\kappa^1 B_\kappa^{d-|\gamma|+\gamma_1}(b(\mathbf{x})) \end{aligned} \tag{2.62}$$

with a_κ^1 a constant independent of $\mathbf{b}$. Applying this procedure to all polynomials $x_i^{\gamma_i}$ we finally end up with:

$$\begin{aligned} x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n} &= (b_0 \mathbf{v}_{0_n} + b_1 \mathbf{v}_{1_n} + \cdots + b_n \mathbf{v}_{n_n})^{\gamma_n} \cdot \sum_{|\kappa|=d-\gamma_n} a_\kappa^1 a_\kappa^2 \cdots a_\kappa^{n-1} B_\kappa^{d-\gamma_n}(b(\mathbf{x})) \\ &= \sum_{|\kappa|=d} a_\kappa^1 a_\kappa^2 \cdots a_\kappa^n B_\kappa^d(b(\mathbf{x})) \\ &= \sum_{|\kappa|=d} c_\kappa B_\kappa^d(b(\mathbf{x})) \end{aligned} \tag{2.63}$$

which proves that $x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n} \in \operatorname{span}\left(\mathcal{B}^d\right)$ for $0 \leq |\gamma| \leq d$ thereby proving the theorem.

□

The coefficients c_κ in Eq. 2.59 are called a control coefficients, or more commonly, *B-coefficients*[11]. The B-coefficients fully control the particular shape of a B-form polynomial by scaling the individual basis functions. In total, there are as much B-coefficients for a given B-form polynomial as there are basis function terms, with the total number given by Eq. 2.51. The resulting complete B-form polynomial is constructed by multiplying every individual basis function with its corresponding B-coefficient. In the following, the term *simplex polynomial* shall be used intermittently with the term 'B-form polynomial', with both terms pointing to the same entity: a polynomial of any degree, and dimension in barycentric coordinates written in the B-form on a single simplex.

The following example demonstrates the scaling of individual basis function terms with the B-coefficients. Note that the commas in the multi-index values are dropped for readability!

Example 3 (Using B-coefficients)**.** *B-coefficients in action: scaling the basis functions of a third degree trivariate polynomial.*

---

[11] In the past, B-coefficients have also been called Bézier ordinates after Pierre Étienne Bézier (1910-1999), an engineer at car maker Renault who patented them in 1962, but not invented them. The inventor of the Bézier ordinates is actually Paul de Casteljau, another important innovator of multivariate spline theory.

The third degree trivariate B-form polynomial from Eq. 2.54 is expanded as follows:

$$\begin{aligned} p(\mathbf{x}) &= \sum_{|\kappa|=3} c_\kappa B^3_\kappa(b(\mathbf{x})) \\ &= c_{300}B^3_{300}(\mathbf{b}) + \\ &\quad c_{210}B^3_{210}(\mathbf{b}) + c_{201}B^3_{201}(\mathbf{b}) + \\ &\quad c_{120}B^3_{120}(\mathbf{b}) + c_{111}B^3_{111}(\mathbf{b}) + c_{102}B^3_{102}(\mathbf{b}) + \\ &\quad c_{030}B^3_{030}(\mathbf{b}) + c_{021}B^3_{021}(\mathbf{b}) + c_{012}B^3_{012}(\mathbf{b}) + c_{003}B^3_{003}(\mathbf{b}) \\ &= c_{300}b_0^3 + c_{210}3b_0^2b_1 + c_{201}3b_0^2b_2 + c_{120}3b_0b_1^2 + c_{111}6b_0b_1b_2 + \\ &\quad c_{102}3b_0b_2^2 + c_{030}b_1^3 + c_{021}3b_1^2 + c_{012}3b_1b_2^2 + c_{003}b_2^3 \end{aligned}$$

∎

2.2.5 A Vector form of the B-form

The B-form can be expressed in a vector form, which shall prove to be essential when simplex splines are to be used within a parameter estimator. Vector forms of the B-form are well known in the literature, e.g. [104] and [40]. The vector form from [40] is used in this thesis. This particular formulation differs from others present in the literature in its explicit matrix formulation, which simplifies its algorithmic implementation in programming languages like C++ and Matlab.

First, define the vector of B-coefficients for a single n-simplex t_j as follows:

$$\mathbf{c}^{t_j} := [c^{t_j}_\kappa]_{|\kappa|=d} \in \mathbb{R}^{\hat{d}\times 1}. \tag{2.64}$$

In this definition the rows of $\mathbf{c}^{t_j}$ contain individual B-coefficients. In contrast with the ordinary B-form, the vector B-form requires an explicit sorting of the B-coefficients in the vector $\mathbf{c}^{t_j}$. [75] as well [104] introduced a very useful *lexicographical* sorting order on the values of the multi-index κ:

$$\kappa_{d,0,0\cdots0} > \kappa_{d-1,1,0\cdots0} > \kappa_{d-1,0,1,0\cdots0} > \cdots > \kappa_{0\cdots0,1,d-1} > \kappa_{0\cdots0,0,d}. \tag{2.65}$$

The lexicographical sorting rule is used to sort the individual B-coefficients in $\mathbf{c}^{t_j}$ and basis functions. This effectively means that κ is mapped to a linear index i in the vector of B-coefficients or basis functions.

Now let $\mathbf{B}^d_{t_j}$ be defined as the vector of basis polynomials for the simplex t_j.

$$\mathbf{B}^d_{t_j}(\mathbf{b}) := [B^d_\kappa(\mathbf{b})]_{|\kappa|=d} \in \mathbf{R}^{1\times\hat{d}}, \tag{2.66}$$

in which the polynomial members of $\mathbf{B}^d_{t_j}$ are sorted with the lexicographical sorting rule from Eq. 2.65. With Eq. 2.64 and Eq. 2.66 the per-simplex B-form in vector formulation is:

$$p(\mathbf{b}) = \mathbf{B}^d_{t_j}(\mathbf{b}) \cdot \mathbf{c}^{t_j}. \tag{2.67}$$

Example 4 (The vector B-form)**.** *The vector form of the B-form of a first degree quadrivariate polynomial on a single tetrahedron.*

The vector of B-coefficients is constructed using Eq. 2.64 and the lexicographical sorting rule from Eq. 2.65:

$$\mathbf{c}^{t_j} = \begin{bmatrix} c_{1000} & c_{0100} & c_{0010} & c_{0001} \end{bmatrix}^\top \in \mathbb{R}^{4\times 1}.$$

The vector of basis function is constructed using Eq. 2.66 and Eq. 2.65:

$$\mathbf{B}^1_{t_j}(\mathbf{b}) = \begin{bmatrix} B^1_{1000}(\mathbf{b}) & B^1_{0100}(\mathbf{b}) & B^1_{0010}(\mathbf{b}) & B^1_{0001}(\mathbf{b}) \end{bmatrix} \in \mathbb{R}^{1\times 4}.$$

This leads to the vector form of the B-form:

$$p(\mathbf{b}) = \begin{bmatrix} B^1_{1000}(\mathbf{b}) & B^1_{0100}(\mathbf{b}) & B^1_{0010}(\mathbf{b}) & B^1_{0001}(\mathbf{b}) \end{bmatrix} \cdot \begin{bmatrix} c_{1000} \\ c_{0100} \\ c_{0010} \\ c_{0001} \end{bmatrix} \in \mathbb{R} \tag{2.68}$$

■

2.2.6 The B-coefficient net

The B-coefficients are ordered in a unique spatial structure called the B-coefficient net, or *B-net* for short. This spatial structure is well known in the literature, see e.g. [52, 109, 191, 96] and [104]. The B-net enables a number of features that are unique to multivariate simplex splines:

- Simplification of the formulation of the continuity equations that govern continuity between polynomials on neighboring simplices.
- The ability to perform local model modification without disrupting the global model structure by modifying B-coefficients close to a specific region of interest.
- Estimated variances in the B-coefficients also have a spatial location, which allows for a new kind of statistical model quality assessment.

The spatial location in barycentric coordinates of a B-coefficient c_κ inside an n-simplex is given by:

$$b(c_\kappa) = \frac{\kappa}{d}, \ \ |\kappa| = d, \tag{2.69}$$

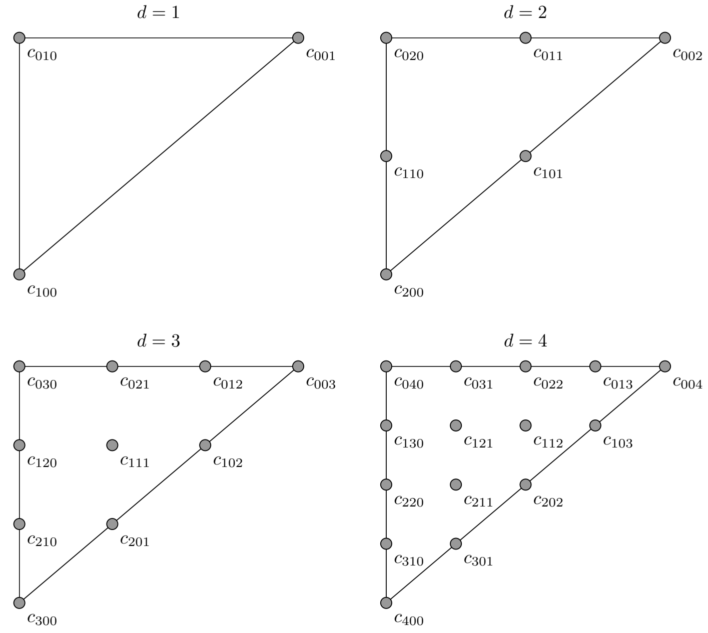


Figure 2.18: Spatial location of B-coefficients for 2-dimensional simplex polynomials of degree 1 to 4.

which is equal to the location of the unique maximum of the Bernstein basis polynomial $B_\kappa^d(\mathbf{b})$ from Eq. 2.56. The uniqueness of the B-coefficients' spatial location property may be easy to underestimate. Consider for example a bivariate polynomial in Cartesian coordinates $z(x,y) = a_0x^2 + a_1xy + a_2y^2$. The question 'what is the location of the coefficients a_0, a_1 and a_2?' has no answer because the question itself is meaningless; the coefficients have a global influence on $z(x,y)$. On the other hand, if one considers the bivariate polynomial in barycentric coordinates $p(b_0,b_1) = c_{20}b_0^2 + c_{11}b_0b_1 + c_{02}b_1^2$ the above question has a perfectly reasonable answer: in barycentric coordinates we would find $b(c_{20}) = (1,0)$, $b(c_{11}) = (0.5,0.5)$ and $b(c_{02}) = (0,1)$.

Using Eq. 2.69, the B-net of simplex polynomials of any degree and dimension can be constructed. In Figure 2.18 four B-nets for trivariate (in barycentric coordinates) simplex polynomials of degree 1 to 4 are shown on a single triangle. The density of the B-net clearly increases with increasing polynomial degree. In Figure 2.19 the B-nets for quadrivariate simplex polynomials of degree 1 and 2 are shown. The total number of B-coefficients for the quadrivariate case is higher for a given polynomial degree, but the overall structure of the B-net is similar.

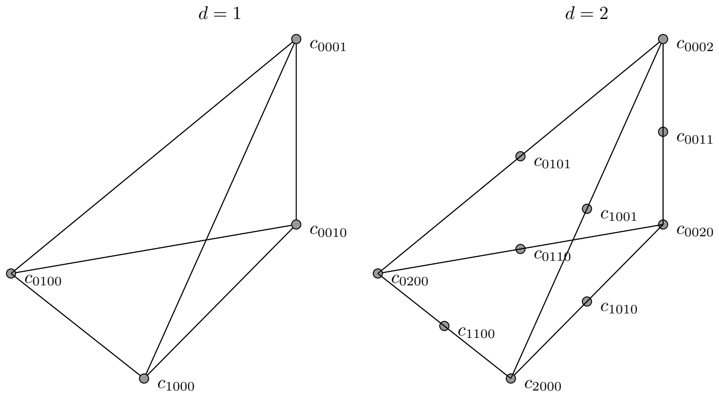


Figure 2.19: Spatial location of B-coefficients for 3-dimensional simplex polynomials of degree 1 and 2.

Finally, in Figure 2.20 the functioning of the B-net is demonstrated for a bivariate B-form polynomial $p(\mathbf{x})$ of degree 12 on a single simplex. There are a few interesting observations that can be made from the figure. First, the figure shows that a polynomial does not have to go through every single B-coefficient. This means that the value of the polynomial in general differs from the value of the B-coefficient at the same location. The only exceptions are the vertices, where the B-coefficients have the exact same value as the polynomial. The barycentric coordinate at a vertex contains only one nonzero component which is equal to 1. The partition of unity property then dictates that there can be only one 'active' basis function at the vertex which also has the value 1. Therefore, the value of the simplex polynomial at a vertex must be equal to the B-coefficient at that vertex. A second observation is that the B-form polynomial is bounded by the B-coefficients. This is a direct result of the stable local basis property of the Bernstein basis polynomials.

2.2.7 Spline Spaces

A spline space is the space of all spline functions s of a given degree d and continuity order C^r on a given triangulation $\mathcal{T}$. Such spline spaces have been studied extensively, see e.g. [109] [101] [104]. In this thesis the definition of the spline space from [104] is used:

$$\mathcal{S}_d^r(\mathcal{T}) := \{s \in C^r(\mathcal{T}) : s|_t \in \mathcal{P}_d, \ \forall t \in \mathcal{T}\} \tag{2.70}$$

with s the n-variate simplex spline function of degree d and continuity order r on the triangulation $\mathcal{T}$ and with $\mathcal{P}_d$ the space of all polynomials of total degree d. The definition of the spline space in Eq. 2.70 provides a convenient notation for stating the degree, continuity

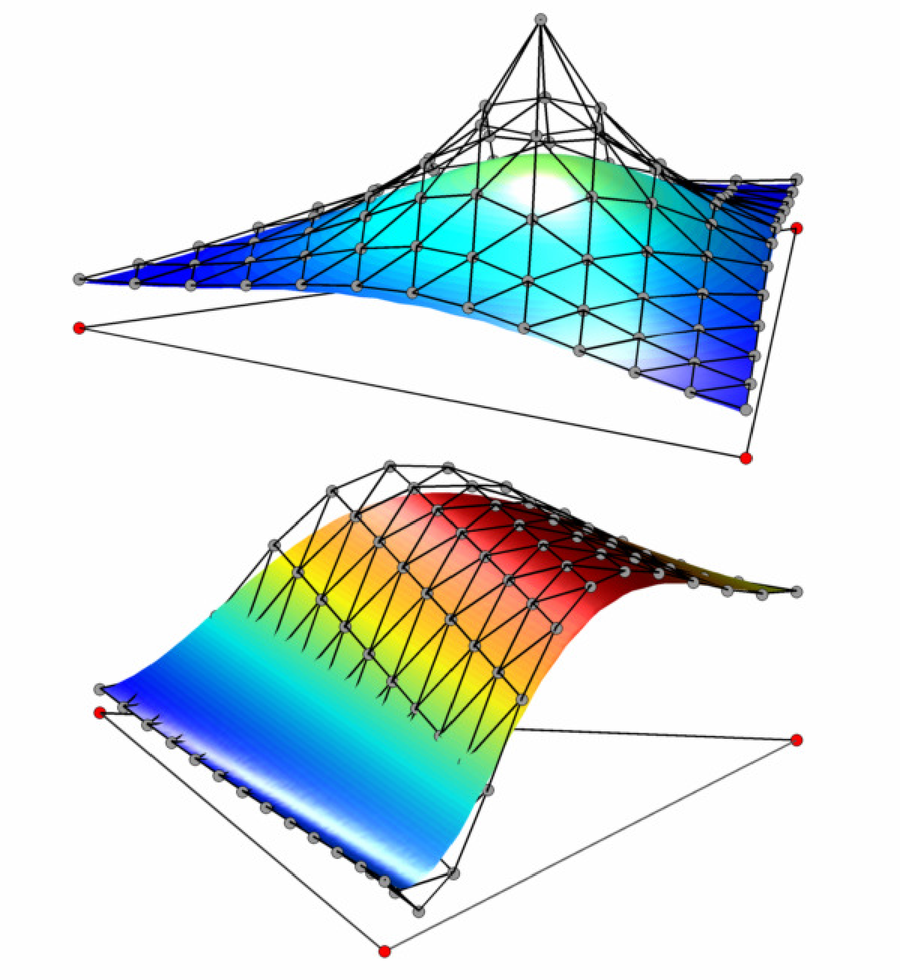


Figure 2.20: B-nets of a trivariate 12^{th} degree simplex polynomial approximating a Gaussian radial basis function (top), and a periodic function (bottom).

and triangulation of a spline solution without having to specify individual spline functions. For example, $\mathcal{S}_3^1(\mathcal{T})$ is the space of all cubic spline functions with continuity order C^1 defined on the triangulation $\mathcal{T}$.

2.3 Triangulations of Simplices

The true approximation power of a simplex spline function is only attained when many simplex polynomials are combined. Because each of the simplex polynomials is defined on its own respective simplex, combining simplex polynomials means combining simplices. While in theory any configuration of simplices would suffice in the construction of a global spline function, there are some stringent constraints on which simplex configurations are allowed and which are not. These constraints are based on some important numerical arguments, but most importantly, on the requirement for continuity of simplex polynomials between neighboring simplices. When these constraints are met, the resulting set of simplices is called a *triangulation*.

In this section the various aspects of triangulations for simplex splines are discussed. First, a number of simplex metrics important for triangulation purposes are introduced in Sec. 2.3.1. Then, in Sec. 2.3.2 a general definition of a triangulation is given. In the following sections a number of the most used triangulation types are presented. In Sec. 2.3.3 the Type I/II triangulation is introduced, followed by the well-known Delaunay triangulation in Sec. 2.3.4. In Sec. 2.3.5 an important rule for the orientation of B-nets in general triangulations is introduced. As it turns out, the sharing of information between simplices is determined by the Star structure of a simplex, which is discussed in Sec. 2.3.6. Finally, the section is concluded with the presentation of a geometric triangulation optimization method which aims to produce high quality triangulations.

2.3.1 Simplex metrics

Before starting the discussion on the definition of triangulations, a number of metrics for simplices need to be defined. These metrics not only play an important role in assessing the quality of a triangulation, but also act as optimization criteria when performing geometric triangulation optimization. In this thesis four different metrics are used, see Figure 2.21; the relative location and radius of the circum (hyper) sphere of a simplex (SRLC), the ratio between the radius of the circumsphere and the shortest simplex ridge (SRSC), the minimum in-plane angle between two simplex ridges (SMA) and the total number of data points contained by a simplex (SDP). All metrics except for the simplex data volume metric SDP are taken from the literature, see [193].

The SDP is the single simplex metric that does not only depend on the vertex configuration of a simplex, but also on the spatial distribution of the dataset. The SDP was found to be the most important simplex metric because an insufficient value will in

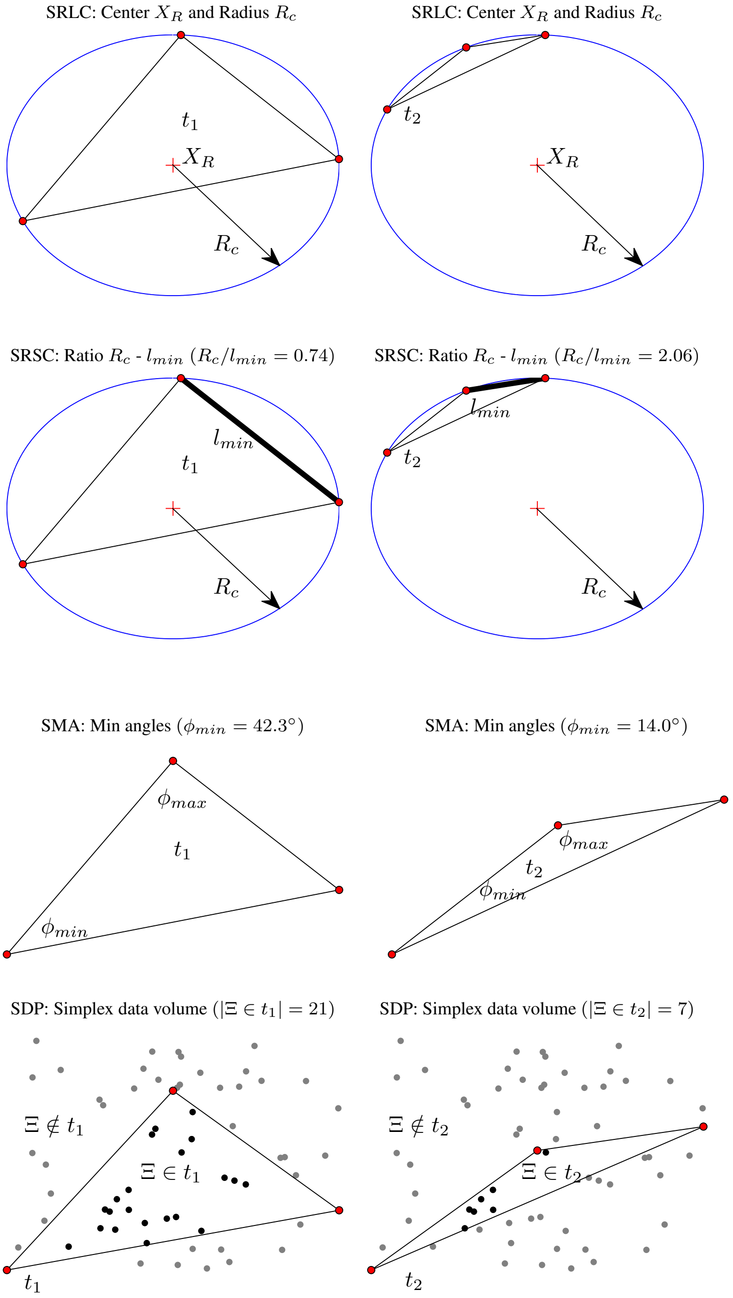

SRLC: Center X_R and Radius R_c
t_1
X_R
R_c
SRLC: Center X_R and Radius R_c
t_2
X_R
R_c
SRSC: Ratio R_c - l_{min} ($R_c/l_{min} = 0.74$)
l_{min}
t_1
R_c
SRSC: Ratio R_c - l_{min} ($R_c/l_{min} = 2.06$)
l_{min}
t_2
R_c
SMA: Min angles ($\phi_{min} = 42.3°$)
ϕ_{max}
t_1
ϕ_{min}
SMA: Min angles ($\phi_{min} = 14.0°$)
ϕ_{max}
t_2
ϕ_{min}
SDP: Simplex data volume ($|\Xi \in t_1| = 21$)
$\Xi \notin t_1$
$\Xi \in t_1$
t_1
SDP: Simplex data volume ($|\Xi \in t_2| = 7$)
$\Xi \notin t_2$
$\Xi \in t_2$
t_2

Figure 2.21: Four different simplex metrics for a well-defined simplex (left) and a sliver simplex (right).

Table 2.4: Guidelines for the metrics of well defined ('good') and sliver ('bad') simplices.

Simplex metric	'Good'	'Bad'
Location and radius of circumsphere center (SRLC)	SRLC $\in t$, $R_c \leq K$	SRLC $\notin t$, $R_c > K$
Ratio radius circumsphere and shortest ridge (SRSC)	SRSC < 2	SRSC > 2
Minimum angle (SMA)	SMA $\geq 14^\circ$	SMA $< 14^\circ$
Minimum data points (SDP)	SDP $\geq \hat{d}$	SDP $< \hat{d}$

general prevent an optimization problem from having a solution. The SRLC, SRSC, and SMA metrics on the other hand, will not affect the direct solvability of an optimization problem, but can cause numerical problems in the solver algorithm. Additionally, these metrics will also give an indication of the distribution of approximation power of a spline function. The SRLC, SRSC and SMA are purely geometric metrics because they depend solely on the geometric configuration of the vertices of the simplex. The values of the SRSC and SMA in particular help to determine whether a simplex is a so called sliver-simplex, see the right hand plots in Figure 2.21. A sliver simplex is a simplex with a large circumsphere radius with respect to its shortest edge (large SRSC), and a small minimum angle (small SMA). Sliver simplices are undesirable because they tend to produce non-uniformities in the distribution of the approximation power of a spline function. This can be seen as follows. The difference in length between the shortest and longest ridge of a sliver simplex will be large. This causes the B-net of the simplex to be compacted in the direction of the shortest ridge and expanded in the other directions. The simplex polynomial will therefore have a greater approximation power in the direction of the shortest ridge as compared to the other directions. In general, non-uniformities in the approximation power of a spline function will lead to inadequate spline models.

In Table 2.4 the four simplex metrics for a well-defined simplex and a sliver simplex are compared. The SDP depends on the total number of per-simplex basis polynomials $\hat{d}$ from Eq. 2.51. The numerical values for the SRLC, SRSC and SMA in the table should be taken as guidelines and not as absolute measures. It may for instance prove to be impossible to create a triangulation for a given dataset and problem domain that does not contain any sliver simplices. Only the data volume metric SDP provides a hard limit on the configuration of a simplex. An optimization problem for the B-coefficient of a simplex spline function defined on a triangulation containing a single simplex with an insufficient SDP will in general not be solvable.

2.3.2 General definition

A triangulation is a special ordering of n-simplices in which two neighboring simplices are only allowed to overlap on at most $n-1$ of their vertices. The formal definition of a

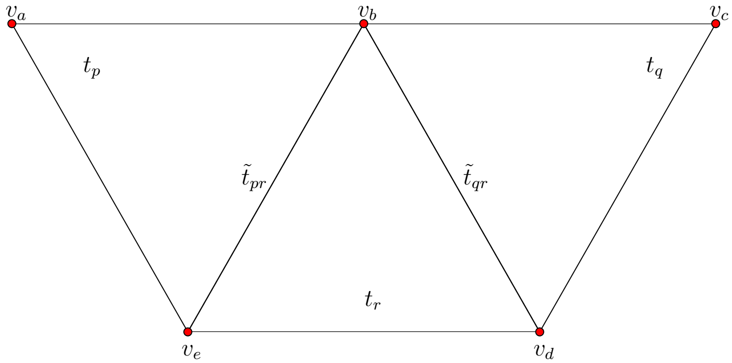


Figure 2.22: A triangulation consisting of the three simplices t_p, t_q and t_r together with the two edge facets $\tilde{t}_{pr}$ and $\tilde{t}_{qr}$.

triangulation $\mathcal{T}$ consisting of J simplices is:

$$\mathcal{T} := \bigcup_{i=1}^{J} t_i, \;\; t_i \cap t_j \in \left\{\emptyset, \tilde{t}\right\}, \; \forall (t_i, t_j) \in \mathcal{T}, \; i \neq j \tag{2.71}$$

with $\tilde{t}$ a k-face of the n-simplices t_i and t_j, with $0 \leq k \leq n-1$ as discussed in Sec. 2.2.1. A special case of $\tilde{t}$ would be $k = n-1$, in which case $\tilde{t}$ is called an *edge facet*. In this case two simplices of dimension n share exactly $n-1$ vertices, and each have a single unshared vertex. Another special case is $k = 0$ in which case $\tilde{t}$ is just a single vertex. In Figure 2.10 some examples were shown of valid triangulations, while in Figure 2.11 an example is given of an invalid triangulation.

In Figure 2.22 a simple two-dimensional triangulation is shown consisting of the three simplices t_p, t_q and t_r. This triangulation has two edge facets: $\tilde{t}_{pr} = t_p \cup t_r = \langle v_b, v_e \rangle$ and $\tilde{t}_{qr} = t_q \cup t_r = \langle v_b, v_d \rangle$. Also, the simplices t_p and t_q share a single vertex: $\tilde{t}_{pq} = t_p \cup t_q = v_b$.

In order to assess the quality of a triangulation, an objective performance measure of the entire triangulation is required. In this thesis it was found that the mean simplex radius to shortest edge ratio for the complete triangulation provided a good measure of triangulation quality. This measure is based on the earlier defined SRSC simplex metric and for an n-dimensional triangulation is defined as follows:

$$P_{\mathcal{T}} := \frac{1}{J} \sum_{i=1}^{J} \frac{r_{\Theta_j}}{\min \left| \mathbf{v}_u - \mathbf{v}_w \right|_{\mathbf{v}_u, \mathbf{v}_w \in \tilde{t}_j}}, \tag{2.72}$$

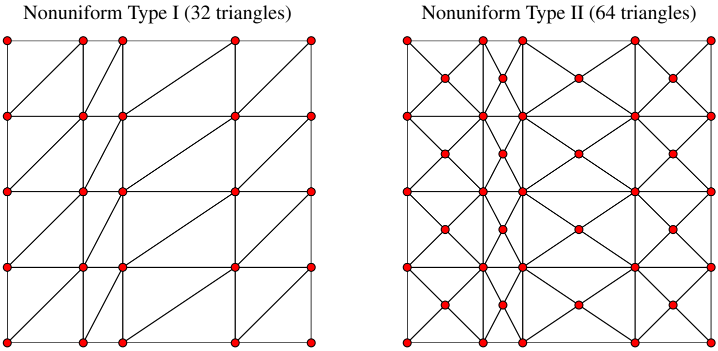

Nonuniform Type I (32 triangles)
Nonuniform Type II (64 triangles)

Figure 2.23: Nonuniform rectangular Type I (left) and Type II (right) triangulations in two dimensions.

with r_{Θ_j} the radius of the circum-hypersphere of the n-simplex t_j, with $\tilde{t}_j$ an edge facet of t_j and with $\mathbf{v}_u$ and $\mathbf{v}_w$ vertices in the edge facet $\tilde{t}_j$.

2.3.3 Type I/II triangulations

The Type I and Type II triangulations are the simplest forms of triangulation. A Type I/II triangulation is constructed by filling in the multi-dimensional cells of a grid with a single *symmetric* prototype triangulation of the hypercube, or n-cube, see Figure 2.23 for a 2-dimensional example and Figure 2.24 for a 3-dimensional example. Mara finds in [128] that a symmetric Type I triangulation contains $n!$ simplices:

$$|\mathcal{T}| = n! \tag{2.73}$$

The difference between the Type I and Type II triangulations is that the Type II uses a single extra vertex at the center of the n-cube. Valid Type I and Type II triangulations of the form Eq. 2.71 can only be constructed if the n-cube triangulation is symmetric along the grid axes. In that case, the grid cells can be filled using scaled and unrotated versions of the n-cube triangulation.

In Figure 2.25 a triangulation of the 3-cube for a Type I triangulation is shown. This triangulation consists of six tetrahedrons and clearly is symmetric along the ridges of the 3-cube. In Figure 2.26 the Type II triangulation of the 3-cube is shown. In this case the cube is split into six pyramids, with each pyramid consisting of two tetrahedrons.

The total number of n-simplices in the symmetric Type I/II triangulation of the n-cube as a function of the dimension is given in Table 2.5. The values for the Type I triangulation were obtained using Eq. 2.73.

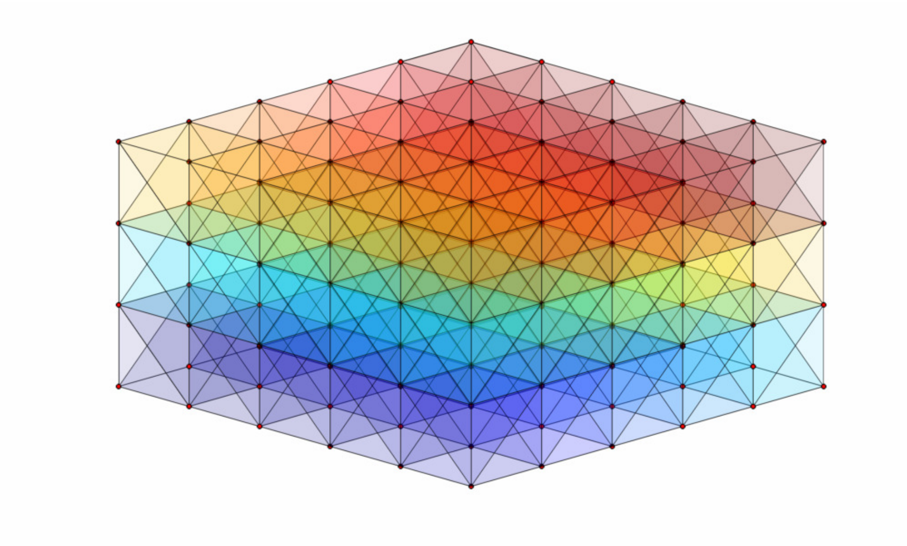


Figure 2.24: Uniform rectangular Type I triangulation in three dimensions consisting of 450 tetrahedrons.

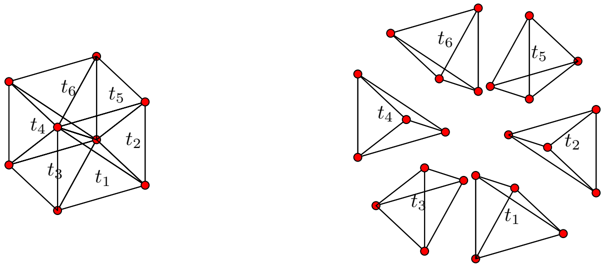


Figure 2.25: Triangulated 3-cube (left) for a Type I triangulation consisting of 6 tetrahedrons and its exploded view (right).

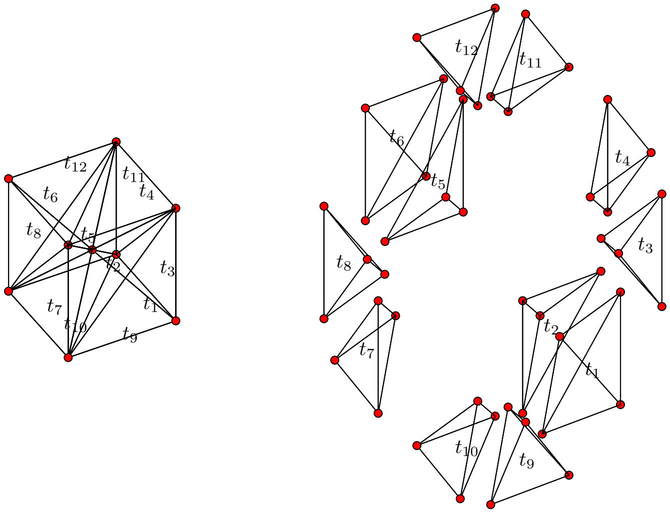


Figure 2.26: Triangulated 3-cube (left) for a Type II triangulation consisting of 12 tetrahedrons and its exploded view (right).

The first of the bracketed numbers in Table 2.5 were obtained with the Qhull triangulation engine [12] to triangulate the n-cube. The second of the bracketed numbers in Table 2.5 is the theoretical minimum number of simplices in a triangulation. These numbers are given by the work by Cottle [18], Sallee [184], Hughes and Anderson [77], and Orden and Santos [166]. Note that the minimum number of simplices given by these authors is significantly less than the numbers obtained by using Qhull. In other words, the n-cube can be triangulated in many different ways, which is again a reminder that higher dimensional spaces are complex and sometimes counterintuitive.

The most important advantages of Type I/II triangulation method over other triangulation methods, like the Delaunay triangulation method discussed in the next section, is that the simplices of the Type I/II are guaranteed to be well defined in terms of the metrics from Sec. 2.3.1; sliver simplices are in general not present in Type I/II triangulations.

While at first it would seem that the Type I and Type II triangulations are purely rectangular triangulation forms, this is actually not the case. Approximations of non-rectangular domains can be made using Type I/II triangulations by removing simplices extending beyond the convex hull of the non-rectangular domain, see Figure 2.27. Additionally, the triangulated n-cubes can be scaled to produce nonuniform Type I and Type II triangulations. Simplex removal together with n-cube scaling is a simple but powerful

---

[12] Qhull (http://www.qhull.org/) is an open source software package for computing Delaunay triangulations and Convex hulls of point sets in any number of dimensions. Matlab uses the Qhull libraries in its *delaunayn* and *convhulln* functions.

Table 2.5: Per-n-cube simplex count for triangulated n-cubes of dimension 2 to 8 as produced by the Qhull triangulation engine. Values between brackets are minimal theoretical triangulations of the n-cube from [166].

n	n-cube vertex count	Type I simplex count	Type II simplex count
2	4	2 (2,2)	4
3	8	6 (6,5)	12
4	16	24 (22,16)	44
5	32	120 (108,67)	210
6	64	720 (618,308)	1236
7	128	5040 (4217,1493)	8870
8	256	40320 (33313,$\leq$ 11944)	65298

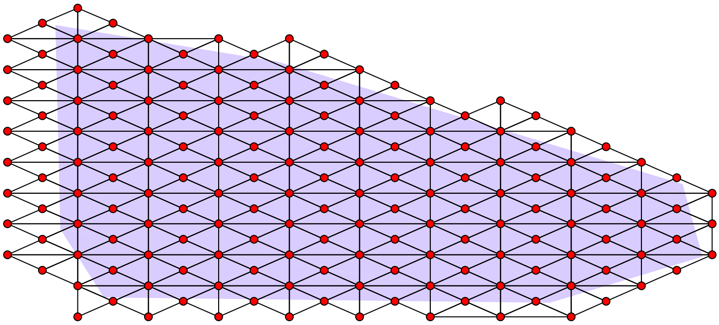


Figure 2.27: Uniform non-rectangular Type II triangulation in two dimensions.

method for optimization of Type I/II triangulations, a topic that will be discussed in more detail in Sec. 2.3.7.

A note must be made at this point on the generality of type I/II triangulations. While the principle of the Type I/II triangulation is general in any number of dimensions, the practical construction of these triangulation types in dimensions higher than 8 becomes problematic. This is because of the computational cost, but more importantly, because of the demands on the size of the required data set.

In order to appreciate the problem, one must consider the sheer size of higher dimensional spaces. Hypercubes of dimension more than five are truly enormous structures. For example, the 8-cube is a structure with $2^8 = 256$ vertices, or corners, see Table 2.5. An 8-dimensional Type I triangulation consisting of triangulated 8-cubes would then contain 33313 eight-dimensional simplices per 8-cube while an 8-dimensional Type II triangulation would contain a staggering 65298 eight-dimensional simplices per 8-cube! In order to find

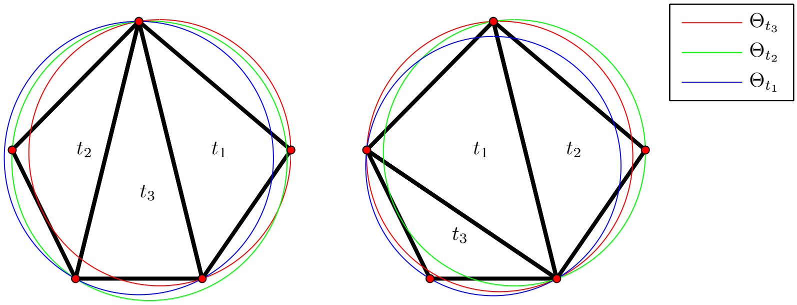


Figure 2.28: Two valid triangulations of two identical vertex sets: a non-Delaunay triangulation (left) and a Delaunay triangulation (right).

a first order (linear) solution to a data approximation problem on a Type II triangulation of this single 8-cube, we would need at least 587682 (scattered) data points, distributed such that every 8-simplex contains a minimum of 9 data points. This requirement on data volume is a direct result of the SDP simplex measure from Table 2.4 that provides a bound on the total number of data points per simplex required for a solution to exist.

In short, the Type I/II triangulation is a very powerful and easy to implement triangulation method. It produces well defined simplices and avoids the creation of sliver simplices by using triangulated n-cubes as basic units. In dimensions higher than 5, however, the n-cubes become so large that it may be hard to meet the simplex data volume metric from Table 2.4.

2.3.4 Delaunay triangulations

The Delaunay triangulation is perhaps the best known triangulation type. Delaunay triangulations are the most flexible type of triangulation to be used with simplex splines. The Delaunay triangulation method was invented by the Russian mathematician Boris Delaunay (1890-1980) in 1934. It has seen many uses over the years in fields as diverse as terrain modeling [190, 27] and wireless network design [114]. The most important advantage of the Delaunay triangulation method over other triangulation methods is that it makes no assumption on the configuration of the set of points to be triangulated, other than that they should be non-degenerate. Furthermore, the Delaunay triangulation method is general in any number of dimensions and easy to implement algorithmically, see [27].

The general definition of the Delaunay triangulation is the following. In a valid Delaunay triangulation, the vertices of the n-simplex t_j must be located exactly on, and only on, the circum-hypersphere of t_j. This is the *Delaunay condition* for simplices. That is, if Θ_{t_k} is the circum-hypersphere of the n-simplex t_k, and $\mathcal{V}_j$ are the vertices of t_j, then the Delaunay condition is the following:

$$\begin{aligned} \mathcal{V}_j &\in \Theta_{t_k} \;\text{ if }\; j = k \\ \mathcal{V}_j &\notin \Theta_{t_k} \;\text{ if }\; j \neq k \end{aligned} \tag{2.74}$$

The principle of the Delaunay triangulation is demonstrated in Figure 2.28. In the left hand plot of this figure a set of 5 vertices is triangulated in a non-Delaunay simplex configuration. Notice how the circumsphere of t_2 contains not only the vertices of t_2, but also a vertex from t_3. The circumsphere of t_3 contains the vertices of t_3, but also a vertex from t_2. In this case, only t_1 is 'Delaunay' as its circumsphere contains only vertices from t_1. In the right hand plot of Figure 2.28 the exact same set of vertices is triangulated using a Delaunay triangulation algorithm. In this case, the circumspheres of the three resulting simplices contain only their respective vertices. This example shows that a set of vertices can be triangulated such that the resulting triangulation is valid in terms of Eq. 2.71, but not a Delaunay triangulation according to Eq. 2.74. This is why, in general, a Type I/II triangulation is not a Delaunay triangulation. When the grid of hypercubes has a lower or higher resolution in one direction, then the simplices resulting from the scaling and translating of the triangulated hypercube are no longer in a Delaunay configuration.

A larger scale example of a valid Delaunay triangulation is shown in Figure 2.29. In the left hand plot of Figure 2.29 the circumspheres of a number of simplices are drawn. This plot again illustrates the general definition of the Delaunay triangulation from Eq. 2.74; the circumsphere of any simplex t_j contains only the vertices of t_j. In the right hand plot of Figure 2.29 the Voronoi diagram of the Delaunay triangulation in the left hand plot is shown. A Voronoi diagram can be considered the dual graph of a Delaunay triangulation. The lines in a Voronoi diagram are the equidistance points of two neighboring vertices. The Voronoi diagram therefore provides a graphical depiction of the closest neighbor set of a set of vertices. This closest neighbor set can be used to create the Delaunay triangulation of the set of vertices by constructing simplices which each contain the most compact set of $n + 1$ vertices.

In general, the Delaunay triangulation of a set of scattered vertices is unique. However, when the vertices are not scattered but in some way coplanar, there will be multiple sets of $n + 1$ vertices with equal compactness. The Voronoi diagram of the set of coplanar vertices will then be symmetric, and there will be many different valid Delaunay triangulations of that same vertex set, see the example in Figure 2.30. The actual triangulation produced by a Delaunay triangulation algorithm may in this case be determined by roundoff error rather than geometry[13].

While the Delaunay triangulation method is very flexible, it tends to produce badly

---

[13] the Qhull package issues a warning when coplanar points are detected. In most cases, however, a valid

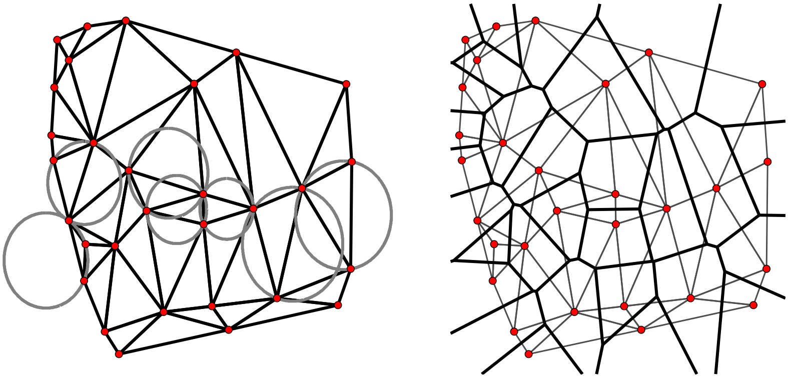


Figure 2.29: Delaunay triangulation of a set of scattered vertices together with some simplex circumspheres (left), and the Voronoi diagram of the same vertex set. (right)

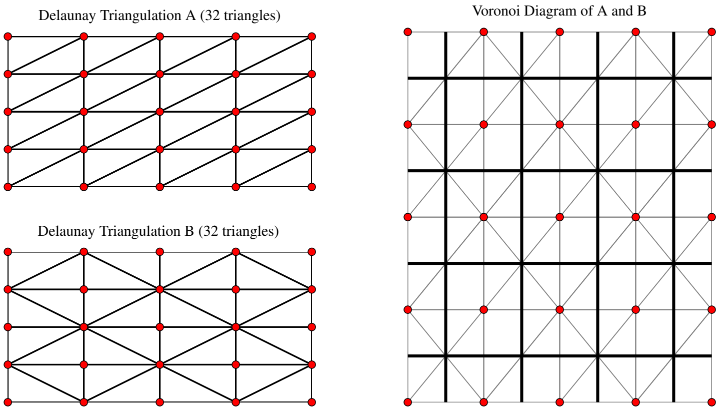

Delaunay Triangulation A (32 triangles)
Delaunay Triangulation B (32 triangles)
Voronoi Diagram of A and B

Figure 2.30: Two valid Delaunay triangulations of a coplanar vertex set and their shared Voronoi diagram.

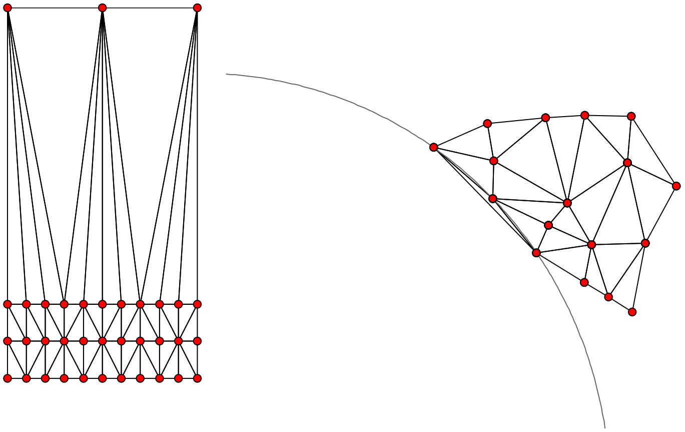


Figure 2.31: Two common problems with Delaunay triangulations: simplex fans (left) and sliver simplices (right).

defined simplices, especially along the boundaries of the triangulation domain. This is a fact that has been recognized by many authors in the past, see e.g. [14, 183] and [192]. In Figure 2.31 the two most common problems with Delaunay triangulations are illustrated. The left hand plot shows a so-called simplex fan. Simplex fans are produced when the vertex density changes abruptly, causing the creation of simplices with badly defined SRLC and SRSC metrics. In the right hand plot of Figure 2.31 an illustration is given of a sliver simplex on the boundary of the spline domain. The sliver simplex, in this case, has a circumsphere (the blue line in Figure 2.31) that has a larger radius than the entire triangulation.

2.3.5 B-nets on triangulations

The relative ordering of B-coefficients within a simplex is determined using the expression from Eq. 2.69. However, no rules are specified in the literature for orienting the B-net as a whole with respect to the simplex vertices. In fact, there are $(n + 1)!$ different ways of orienting the B-net within an n-simplex, and all are valid from the perspective of a single simplex. When working with general triangulations, however, a formal orientation rule is essential, as without it a systematic algorithmic implementations of solution systems and constraint conditions are impossible.

The definition of a rule for the orientation of the B-net requires that every vertex in a triangulation has a globally unique index. Now define the *vertex B-coefficients* as the

---

Delaunay triangulation can still be produced by giving each vertex a very small (i.e. $1e^{-6}$) random translation or 'joggle'. In that case, the resulting Delaunay configuration is effectively determined by the random joggle.

lexicographically sorted subset of B-coefficients located at the simplex vertices as follows:

$$\{\check{c}^t_{(\gamma\cdot d)}\}_{|\gamma|=1} \subset \{c^t_\kappa\}_{|\kappa|=d}, \tag{2.75}$$

with $\{\check{c}_{(\gamma\cdot d)}\}_{|\gamma|=1}$ the vertex B-coefficients for a degree d B-form polynomial on the n-simplex t. The vector of lexicographically sorted vertex B-coefficients then is:

$$\check{\mathbf{c}}^t = [\{\check{c}^t_{(\gamma\cdot d)}\}]_{|\gamma|=1} \in \mathbb{R}^{n\times 1}. \tag{2.76}$$

The B-net should be oriented such that the vertex B-coefficient with the highest lexicographical sorting order is located at the vertex with the highest index. The vertex B-coefficient with the second highest sorting order should be located at the vertex with the second highest index, and so on. This is the B-net Orientation Rule , which formal definition is given below.

Definition 3. ***The B-net Orientation Rule*** *Let every n-simplex t consist of a tuple $\mathcal{V}_t$ of $n+1$ globally indexed vertices from Eq. 2.24. The vertices in the set are ordered using the rule from Eq. 2.25. A new set is now introduced which consists of tuples of vertices and B-coefficients as follows:*

$$\mathcal{V}_{\mathcal{B}} = \{(c_{d,0,0,\ldots,0}, \mathbf{v}_{p_0}), (c_{0,d,0,\ldots,0}, \mathbf{v}_{p_1}), \ldots, (c_{0,0,0,\ldots,d}, \mathbf{v}_{p_n})\}. \tag{2.77}$$

The B-net orientation rule, which must hold for all $n+1$ vertex B-coefficients, then is:

$$\left\{(\check{\mathbf{c}}^t_i, \mathbf{v}_{p_i})\right\}_{i=0}^{n} = \mathcal{V}_{\mathcal{B}}, \ \ \forall t \in \mathcal{T}. \tag{2.78}$$

with $\check{\mathbf{c}}^t_i$ the vector of vertex B-coefficients from Eq. 2.76.
□

Example 5 (The B-net Orientation Rule)**.** *Use the B-net orientation rule to determine the sorting order of the vertices in Figure 2.32.*

When the B-net orientation rule is applied to the B-net in Figure 2.32 we find the following vertex ordering: $c > b > e > a, d$. This can be seen as follows. Using Eq. 2.78 we find the following sets for t_i, t_j and t_k:

$$\begin{aligned}
\left\{(\check{\mathbf{c}}^{t_i}_e, v_{p_e})\right\}_{e=0}^{n} &= \{(c^{t_i}_{4,0,0}, v_b), (c^{t_i}_{0,4,0}, v_e), (c^{t_i}_{0,0,4}, v_a)\}, \\
\left\{(\check{\mathbf{c}}^{t_j}_f, v_{p_f})\right\}_{f=0}^{n} &= \{(c^{t_j}_{4,0,0}, v_b), (c^{t_j}_{0,4,0}, v_e), (c^{t_j}_{0,0,4}, v_d)\}, \\
\left\{(\check{\mathbf{c}}^{t_k}_g, v_{p_g})\right\}_{g=0}^{n} &= \{(c^{t_k}_{4,0,0}, v_c), (c^{t_k}_{0,4,0}, v_b), (c^{t_k}_{0,0,4}, v_e)\},
\end{aligned}$$

from the first set it can be concluded that $b > e > a$, from the second set that $b > e > d$, and from the third set that $c > b > e$. Combining these results leads to the given vertex ordering $c > b > e > a, d$, with the relative order of a and d undefined. ■

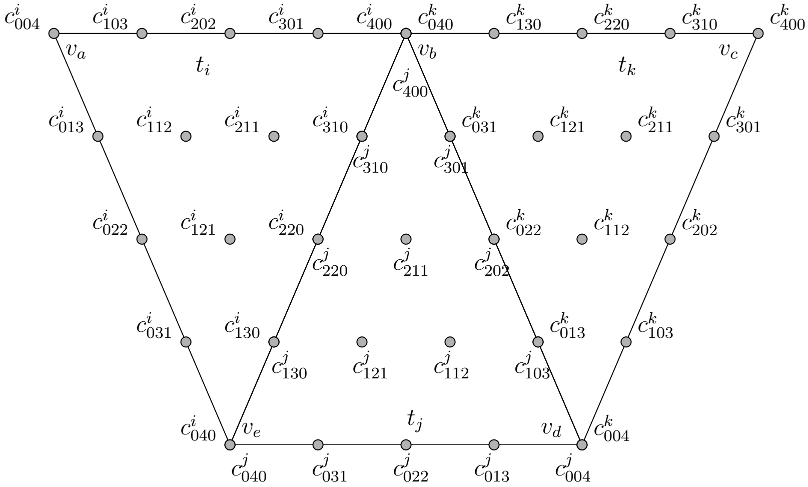


Figure 2.32: B-nets for the three simplices t_i, t_j and t_k defined using the B-net orientation rule. The B-net orientation rule in this case requires that the vertex indices have the ordering: $c > b > e > a, d$.

2.3.6 Simplex Stars

An important sub-structure of a triangulation are *simplex Stars*. The Star of a simplex t_j is the set of all simplices that share at least one vertex with t_j. For example, it will be proved in a later section that the geometry of a simplex Star determines the propagation of disturbances through the global B-net. In this thesis, the definition of the simplex Star from [104] is extended slightly as follows:

$$star_{t_j} = \bigcup \{t_i \cap t_j\}, \ \ \forall t_i \in \mathcal{T}, \tag{2.79}$$

where the subscript is the simplex of which the Star is taken. The Star structure of level-L is formed by taking the Star of all simplices in the Star structure of level-$(L-1)$ as follows:

$$star^L_{t_j} = \bigcup \{t_i \cap t_k\}, \ \ \forall t_k \in star^{L-1}_{t_j}, \ \ \forall t_i \in \mathcal{T}. \tag{2.80}$$

In Figure 2.33 and Figure 2.34 the Star structure of level-0 to level-5 is shown for respectively a Type I and Type II triangulation.

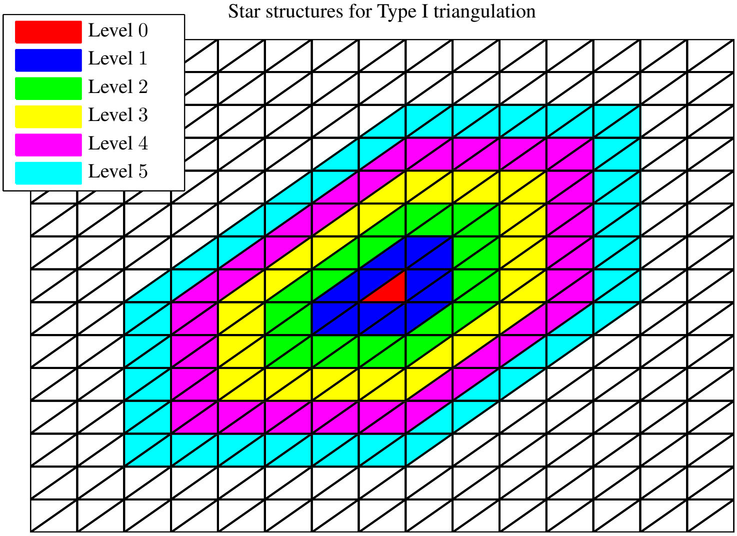


Figure 2.33: Star structures of level-0 to level-5 for a single simplex in a Type I triangulation.

2.3.7 Geometric Triangulation optimization

A triangulation can be considered to be both a curse and a blessing when used in combination with simplex splines. A blessing, because it is very flexible and supports the local basis properties of the simplex polynomials. A curse, because it is not flexible enough and through the Delaunay condition from Eq. 2.74 produces 'bad' simplices. All is not lost, however. Because of their wide use in many fields, the optimization of triangulations is and has been an active research area. In short, triangulation optimization is concerned with the reconfiguring of a triangulation such that no badly defined simplices in terms of the simplex metrics from Table 2.4 are present after the optimization. In the literature, most triangulation optimization methods are constrained Delaunay triangulation (CDT) methods. CDT methods use geometric simplex metrics as user definable constraints in their Delaunay algorithms. The CDT method is therefore considered a *geometric* triangulation optimization method.

Important contributions to triangulation optimization in the form of CDT were from Chew [14], Dey *et al.* [47], Mitchell *et al.* [135], Ruppert [183] and Shewchuk [192]. According to Shewchuk, the most successful triangulation optimization method in two dimensions is Ruppert's method, about which he says the following: "*Jim Rupperts algorithm for two-dimensional quality mesh generation is perhaps the first theoretically*

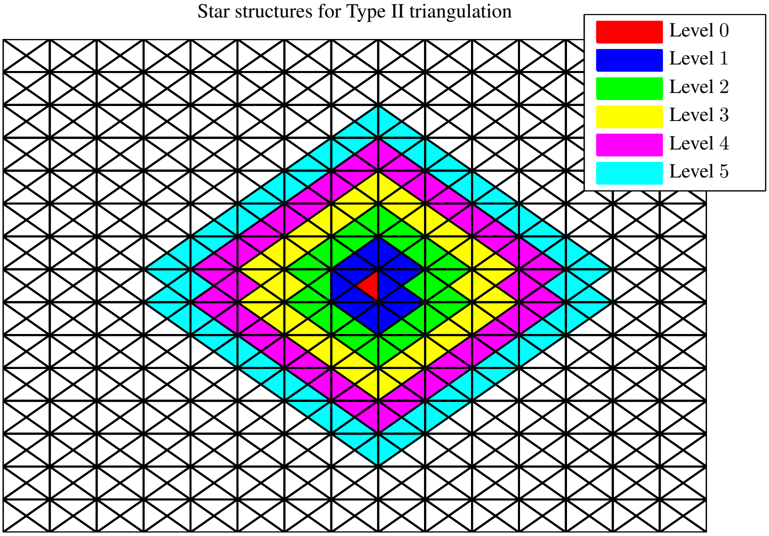


Figure 2.34: Star structures of level-0 to level-5 for a single simplex in a Type II triangulation.

guaranteed meshing algorithm to be truly satisfactory in practice.". In 2001 Shewchuk[14] published a very extensive paper in which he presented Ruppert's algorithm in minute detail [192]. The strength of Ruppert's method is that the quality of the resulting triangulation can be guaranteed, despite its stepwise nature. The central idea of Ruppert's algorithm is the stepwise vertex insertion and removal procedure. For this procedure to work, Ruppert (and Shewchuk) defined a number of different scenarios, each with a particular solution. Shewchuk extended his method to include 3-dimensional constrained Delaunay triangulation which is, for fundamental reasons, more difficult than 2-dimensional CDT. The problem with 3- and higher dimensional CDT is that there are certain polytopes, such as the Schönhardt's polyhedron, that cannot be triangulated at all without adding additional vertices. Shewchuk's work was extended by Si[15] and Gärtner [195, 196] who refined some aspects of Shewchuk's algorithm. The current state of the art in n-dimensional CDT is represented by Shewchuk's 2008 paper [194], which is the first part in a projected three-part series. However, at current there is no implementation of Shewchuk's algorithm. One of the

---

[14] Jonathan Shewchuk is also the author of the open source quality mesh generation program 'Triangle' available for download from http://www.cs.cmu.edu/∼quake/triangle.html.

[15] TetGen, an open source three-dimensional optimal mesh generating application was authored by Si, see http://tetgen.berlios.de/

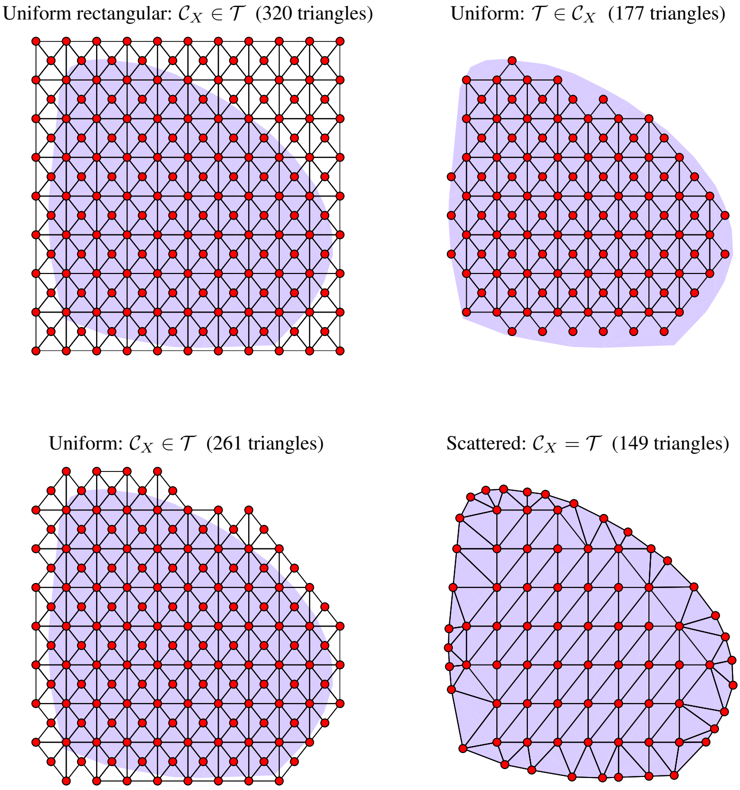
recommendations in this thesis is therefore to create a code implementation of Shewchuk's n-dimensional CDT.

In this thesis a new method for geometric triangulation optimization is used. This new method uses the Type I/II hypercube triangulation method from Sec. 2.3.3 together with a new vertex insertion method based on the intersection points of a grid and the convex hull of the dataset. This method is called the HCI triangulation method which stands for 'Hypercube-Convex hull-Intersection' method. The HCI method differs from CDT methods in the literature because it produces triangulations that are specifically suitable for simplex splines. Such a triangulation should be as close as possible to a Type I/II triangulation because this triangulation type contains simplices of equal dimensions. This in turn creates an homogeneous grid of B-coefficients which provide a homogeneous distribution of approximation power. Additionally, the HCI method allows a post-processing step in which the SDP simplex metric is used as an optimization parameter to produce a triangulation in which every simplex contains a guaranteed minimum of data points.

In Figure 2.35 the four different modes of the HCI triangulation method are demonstrated. The simplest mode, Mode-0, of the HCI method is actually just the Type I/II triangulation method from Sec. 2.3.3. HCI Mode-1 creates a non-rectangular Type I/II triangulation that is guaranteed to minimally fit the convex hull of a non-rectangular dataset. The next mode is HCI Mode-2 which creates a Type I/II triangulation which is guaranteed to overfit the convex hull of a non-rectangular dataset. The most complex mode is HCI Mode-3 which fits a scattered triangulation to the convex hull of a non-rectangular dataset. HCI Mode-3 triangulations exactly fits the convex hull of any dataset, while at the same time guaranteeing adequate values for all simplex metrics. The SDP simplex metric can be used in all modes as a optimization parameter in a post-processing step.

Some numerical results from four different triangulations created with the HCI algorithm are shown in Table 2.6. The dataset in this case consisted of 4551 scattered data points on a non-rectangular domain, see Figure 2.36. The four modes of the HCI algorithm clearly produce triangulations with widely varying simplex metrics. From Table 2.6 it is clear that the only triangulation with a non-zero SDP is the triangulation created with HCI Mode-3 together with an SDP post-processing optimization. The SDP optimization reduces the geometric quality of the triangulation, as the 'worst' simplex can almost be considered a sliver simplex.

A high level listing of the HCI algorithm is shown in **Algorithm** 1. In the following, this algorithm will be discussed step by step, using the plots in Figure 2.37, Figure 2.36, Figure 2.38, Figure 2.39, Figure 2.40, Figure 2.41, and Figure 2.42 as an illustration of the algorithm. The HCI algorithm requires as input a dataset ($X \in \mathbb{R}^{N \times n}$), a specifier for the type of the triangulation (TYPE), the minimum allowed distance between to vertices in the triangulation (R_{min}), the required grid resolution (RES) and the optimization mode (MODE). In the following it is assumed that X is a bivariate non-uniformly scattered dataset consisting of 4551 points on a non-rectangular domain, see Figure 2.36. A direct Delaunay



Uniform rectangular: $\mathcal{C}_X \in \mathcal{T}$ (320 triangles)

Uniform: $\mathcal{T} \in \mathcal{C}_X$ (177 triangles)

Uniform: $\mathcal{C}_X \in \mathcal{T}$ (261 triangles)

Scattered: $\mathcal{C}_X = \mathcal{T}$ (149 triangles)

Figure 2.35: Four triangulation methods for non-rectangular datasets. Top left: a uniform rectangular triangulation completely envelopes the convex hull of the dataset $\mathcal{C}_\Xi$ (Mode-0). Top right: a uniform non-rectangular triangulation completely fits in $\mathcal{C}_\Xi$ (Mode-1). Bottom left: a uniform non-rectangular triangulation minimally envelopes $\mathcal{C}_\Xi$ (Mode-2). Bottom right: a scattered non-rectangular triangulation which exactly fits the convex hull $\mathcal{C}_\Xi$ (Mode-3).

Table 2.6: Numerical quality assessment of the HCI triangulation method.

Method	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{V}\rvert$	Coverage	max SRSC	mean SRSC	min SDP	mean SDP
Delaunay	4551	2300	100.00%	645.04	2.95	0	0.51
HCI Mode-1	79	54	71.39%	0.80	0.80	0	23.10
HCI Mode-2	139	87	100.00%	0.80	0.80	0	16.55
HCI Mode-3	121	78	99.49%	1.45	0.81	0	14.56
HCI Mode-3 + SDP	101	68	99.49%	1.98	0.89	3	22.07

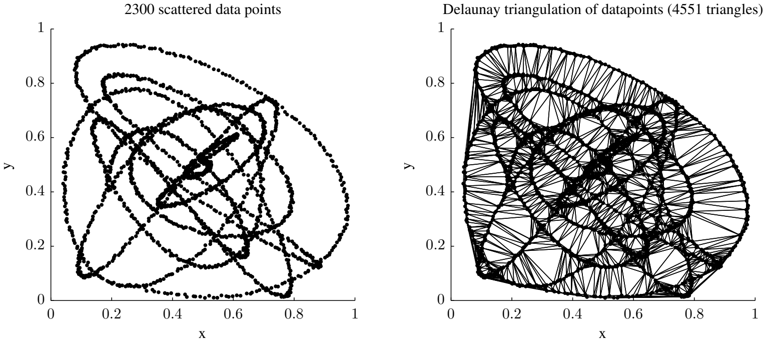


Figure 2.36: The triangulation optimization problem: non-uniformly scattered dataset and its direct Delaunay triangulation.

triangulation of the dataset produces many badly conditioned simplices, see Figure 2.37, validating the use of the HCI algorithm. The HCI algorithm will use a Type I triangulation that exactly fits the convex hull of the dataset (MODE $= 3$) with a symmetric grid (RES $=$ 0.1).

The algorithm starts with the allocation of the global vertex matrix $\mathcal{V}$ and the triangulation index matrix $\mathcal{T}$. The rows of $\mathcal{T}$ contain $n+1$ integers which are indices into the vertex matrix $\mathcal{V}$; each integer indexes a single vertex of the n-simplex. The first functional step is the determination of the convex hull of the dataset $\mathcal{C}_X$ and its corresponding vertex set $\mathcal{V}_\mathcal{C}$, listed on line 1 of **Algorithm** 1, and plotted in Figure 2.37. In most cases, the convex hull of the dataset will be a polygon consisting of many vertices, some located very close together, see for example the top left of Figure 2.37. Triangulating vertices that are too close together will result in a triangulation containing many badly defined simplices. To prevent this, the convex hull of the dataset can be resampled using the R_{min} parameter which is the minimum allowed distance between any two vertices. This step, listed on lines 2-6, is optional. If R_{min} is set to zero, no convex hull points will be removed. In the right hand plot of Figure 2.37, the resampled convex hull of the dataset is shown. Notice in Figure 2.37

that not all data points are contained by the resampled convex hull. If this is unwanted, then the convex hull can be stretched such that every data point is inside it.

```
Require: $X \in \mathbb{R}^{N \times n}$, TYPE $\in \{I, II\}$ , $R_{min} \in \mathbb{R}^n$, RES $\in \mathbb{R}^n$, MODE $\in \{0, 1, 2, 3\}$
 1: $\mathcal{C}_X, \mathcal{V}_\mathcal{C}$ = getConvexHull($X$)
 2: for all vertices $p$ in $\mathcal{V}_\mathcal{C}$ do
 3:     if for any vertex $v \neq p$ in $\mathcal{V}_\mathcal{C}$: $||p - v|| > R_{min}$ then
 4:         remove $p$ from $\mathcal{V}_\mathcal{C}$
 5:     end if
 6: end for
 7: $\mathcal{T}_H$ = triangulateHyperCube(TYPE)
 8: $\mathcal{G}$ = buildGrid(RES)
 9: for all gridcells $g$ in $\mathcal{G}$ do
10:     scale and translate $\mathcal{T}_H$ such that $\mathcal{T}_H \in g$
11:     add $\mathcal{T}_H$ to $\mathcal{T}$, add $V_{\mathcal{T}_H}$ to $\mathcal{V}$
12: end for
13: if MODE = 0 then
14:     return $\mathcal{T}_{rect} = \mathcal{T}, \mathcal{V}$
15: end if
16: if MODE = 1 or MODE = 3 then
17:     for all simplices $t$ in $\mathcal{T}_{nrect} = \mathcal{T}_{rect}$ do
18:         if $t \backslash \mathcal{C}_X \neq \emptyset$ then
19:             remove $t$ from $\mathcal{T}_{nrect}$
20:         end if
21:     end for
22:     if MODE = 3 then
23:         $\mathcal{V}_G$ = getConvexhullGridIntersectionPoints($\mathcal{G}, \mathcal{C}_X$)
24:         for all vertices $p$ in $\mathcal{V}_G$ do
25:             if for all vertices $v$ in $\mathcal{V}_\mathcal{C}$: $||p - v|| < R_{min}$ then
26:                 add $p$ to $\mathcal{V}_\mathcal{C}$
27:             end if
28:         end for
29:         for all vertices $p$ in $\mathcal{V}$ do
30:             if for all vertices $v$ in $\mathcal{V}_\mathcal{C}$: $||p - v|| < R_{min}$ then
31:                 remove $p$ from $\mathcal{V}$
32:             end if
33:         end for
34:         $\mathcal{T}_{scat}, \mathcal{V}_{scat}$ = retriangulate($\mathcal{T}_{nrect}, \mathcal{V} \cup \mathcal{V}_\mathcal{C}$)
35:         return $\mathcal{T}_{scat}, \mathcal{V}_{scat}$
36:     end if
37: else if MODE = 2 then
38:     for all simplices $t$ in $\mathcal{T}_{nrect} = \mathcal{T}$ do
39:         if $t \cap \mathcal{C}_X = \emptyset$ then
40:             remove $t$ from $\mathcal{T}_{nrect}$
41:         end if
42:     end for
43: end if
44: return $\mathcal{T}_{nrect}, \mathcal{V}$
```

Algorithm 1: The Hypercube-Convex hull-Intersection (HCI) triangulation method.

The prototype Type I triangulation $\mathcal{T}_H$ of the 2-cube is then created on line 7. The prototype Type I triangulation of the 2-cube consists of 4 vertices and 2 simplices, according to Table 2.5. Then, on line 8 of **Algorithm** 1, the grid $\mathcal{G}$ is created which spans the complete rectangular domain containing the convex hull of the dataset, see also the left plot in Figure 2.38. The following step, listed on lines 9-12, is the crux of the HCI algorithm. Here the prototype triangulation $\mathcal{T}_H$ is scaled and translated such that the n-cube vertices coincide exactly with the grid $\mathcal{G}$. The result is a rectangular Type I triangulation $\mathcal{T}_{rect}$ and its corresponding vertex set $\mathcal{V}$, see the right hand plot in Figure 2.38. At this point the algorithm can return the Type I triangulation, or go into the next phase which is the convex hull fitting phase.

In the first step of the convex hull fitting phase, all simplices are removed that are outside the (resampled) convex hull $\mathcal{C}_X$, see the left hand plot in Figure 2.39. This step of the HCI algorithm is listed on lines 17-21 of **Algorithm** 1. The resulting triangulation is a non-rectangular Type I triangulation $\mathcal{T}_{nrect}$, that is completely inside the convex hull of the dataset, see the right hand plot in Figure 2.39. The algorithm can now return the non-rectangular Type I triangulation, or continue with the next step, which is the generation of intersection points of the grid with the convex hull. The intersection points can be included with the vertex set from $\mathcal{T}_{nrect}$ to provide a complete covering of the convex hull.

Finding the intersection points of a grid with a convex hull polytope is not a trivial task. In **Algorithm** 1 this step takes place on line 23, where the method *getConvexhullGridIntersectionPoints()* is used to determine the intersection points $\mathcal{V}_G$. In the left hand plot of Figure 2.40 the results of the grid-convex hull intersection method are shown. The solid green dots in this case are the intersection points of the grid $\mathcal{G}$ with the convex hull $\mathcal{C}_X$. The intersection points can now be added to the set of convex hull points $\mathcal{V}_\mathcal{C}$, as listed on lines 24-28, but only if they are not to close to any other point already in $\mathcal{V}_\mathcal{C}$. The resulting set $\mathcal{V}_\mathcal{C}$ is shown in the right hand plot of Figure 2.40. This procedure of adding points to $\mathcal{V}_\mathcal{C}$ guarantees that the convex hull points have priority over any intersection points. In this way it can be guaranteed that the (resampled) convex hull is protected from any further changes.

A final filtering step is the removal of points from $\mathcal{V}$ that are too close to the, now optimal, convex hull point-set $\mathcal{V}_\mathcal{C}$. This step in the algorithm is listed on lines 29-33 of **Algorithm** 1. Any of these points that will be removed from $\mathcal{V}$ can be associated with a simplex in $\mathcal{T}_{nrect}$; these will have to be removed from $\mathcal{T}_{nrect}$. This step is demonstrated in the left hand plot in Figure 2.41, in which points in $\mathcal{V}$ that are too close to $\mathcal{V}_\mathcal{C}$ are shown in bold black, and in which the simplices that will be removed are shown in solid red. The resulting two vertex sets are $\mathcal{V}$ which only contains points which are at a distance of at least R_{min} from any points in $\mathcal{V}_\mathcal{C}$ and the optimal convex hull point-set $\mathcal{V}_\mathcal{C}$, as shown in the right hand plot in Figure 2.41.

The final step in the algorithm is the re-triangulation of the combined vertex set $\mathcal{V}_{scat} = \mathcal{V} \cup \mathcal{V}_\mathcal{C}$ into the scattered triangulation $\mathcal{T}_{scat}$. This action is listed on line 34 of **Algorithm** 1, and again is a non-trivial task. Re-triangulation of $\mathcal{V}_{scat}$ using a Delaunay algorithm can destroy the Type I/II structure, because coplanar vertices have an ambiguous Voronoi diagram as demonstrated earlier in Figure 2.30. Therefore, in the method *retriangulate()* only simplices that contain at least 1 new vertex from $\mathcal{V}_\mathcal{C}$ will be added to the set $\mathcal{T}_{nrect}$ to form $\mathcal{T}_{scat}$. In the left hand plot of Figure 2.42 the resulting triangulation is shown. From this plot, it is immediately clear that the resulting triangulation is of very high quality in terms of the geometric simplex metrics from Table 2.4. There are no sliver simplices and only the simplices with vertices on the convex hull of the dataset are non-Type I/II. In Table 2.6 some quality measures of the created triangulation are shown (HCI Mode-3).

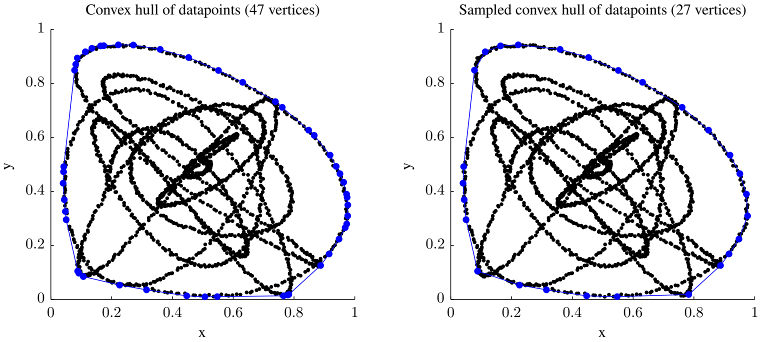


Figure 2.37: Step 1: determine convex hull, and downsample if required.

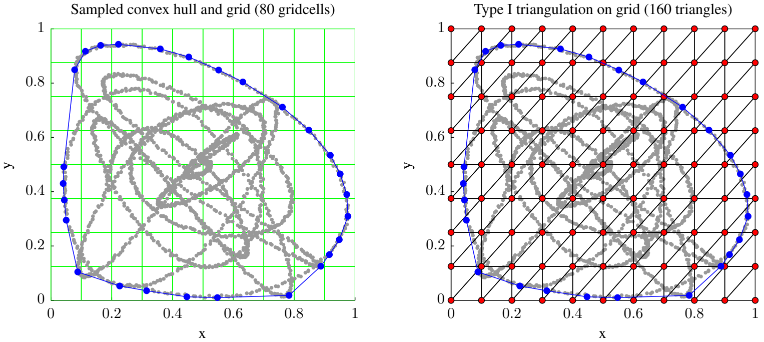


Figure 2.38: Step 2: add grid and create Type I triangulation using triangulated n-cubes.

These quality measures show that the created triangulation is of a high geometric quality, but that the SDP requirement has not been satisfied for all simplices in the triangulation.

The per-simplex data volume requirement SDP has not been taken into account during construction of the HCI triangulation. It was found, however, that the scattered triangulation $\mathcal{T}_{scat}$ is a desirable starting point for an additional post processing step in which the SDP is used as an optimization parameter. In the right hand plot of Figure 2.42 the results from this post-processing step are shown. Clearly, the geometric quality of some of the simplices has been degraded as a result of the SDP based optimization, see also Table 2.6. It should be noted, however, that geometric simplex quality is of secondary importance to the SDP; an inadequate geometric simplex quality can lead to local spline model inaccuracies while an inadequate SDP can prevent a solution from existing in the first place.

Triangulation optimization is a difficult problem because of the non-convex nature of the optimization problem and the ambiguity in the choice of optimization criterion.

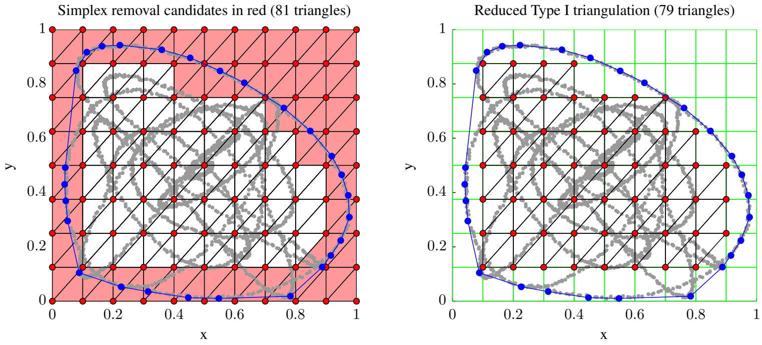


Figure 2.39: Step 3: remove any simplices that are outside the convex hull of the dataset.

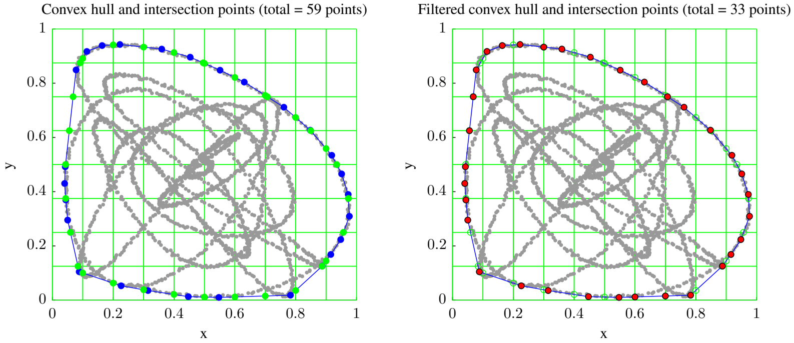


Figure 2.40: Step 4: determine the intersection points of the grid with the convex hull of the data and remove any that are too close to convex hull points.

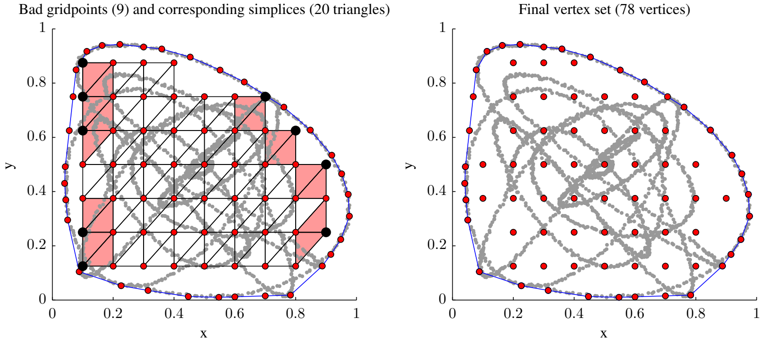


Figure 2.41: Step 5: construct the final vertex set by removing any vertices and corresponding simplices that are too close to convex hull and intersection points.

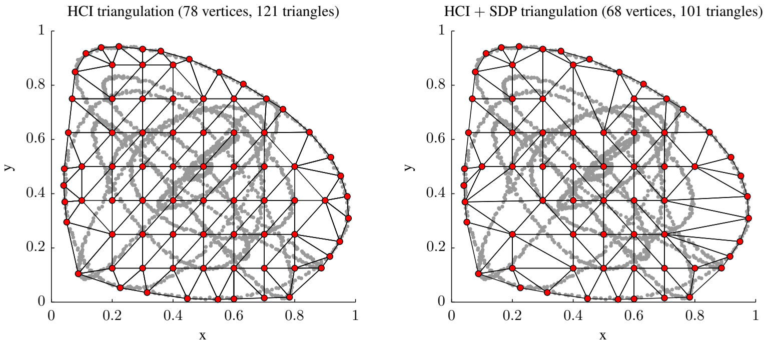


Figure 2.42: Step 6: finalize HCI triangulation, and post-process such that the simplex data volume requirement is met.

Chapter 3

Advances in Simplex Spline Theory

This thesis presents a new method for system identification based on multivariate simplex splines. Multivariate simplex splines are a special type of multivariate spline with many unique and powerful features. The most important of these features are the ability to model scattered multi-dimensional data on non-rectangular domains, and an approximation power that is only bounded by available computing resources. Multivariate simplex splines consist of sets of multivariate polynomials which are each defined on a simplex, a geometric structure which minimally spans a given set of dimensions. The simplices are joined together in a larger spatial structure called a triangulation. In theory, there is no limit to the size and complexity of a triangulation, which effectively means that there is no theoretical limit to the approximation power of a simplex spline function.

In the previous chapter it was shown that the basis functions of the simplex splines are Bernstein basis polynomials in terms of the barycentric coordinates. The Bernstein basis polynomials form a partition of unity which results in a stable local basis for function approximation and data modeling. The Bernstein basis polynomials are scaled by the B-coefficients which for every simplex uniquely determine the shape of the supported basis function. It was also shown that the B-coefficients have a distinct spatial location within their parent simplex, a property unique to simplex splines. It was demonstrated that creating triangulations for simplex splines is not a trivial task. In fact, it turns out that there are many different methods for creating triangulations, each with their own advantages and disadvantages.

It is the aim of this chapter to introduce the theory necessary to elevate the multivariate simplex spline from a tool for simple data modeling into a feasible new method for nonlinear system identification. For this, a number of new concepts need to be explored. The first of these is a new matrix form of the de Casteljau algorithm, which is introduced in Sec. 3.1. In Sec. 3.2 a number of different operations that are possible with B-form polynomials will be discussed. In the same section it will also be proved that the de Casteljau matrix is instrumental in a new formulation of the directional derivatives of the simplex splines. In Sec. 3.3 the equations that govern continuity between neighboring simplices are discussed. It will be shown that the continuity equations give rise to a new effect called B-net propagation, which is the propagation of disturbances between neighboring simplices. Two new theorems will be presented that quantify B-net propagation, and it will be proved that the Type I triangulation is especially susceptible to its effects. Then, in Sec. 3.4 a new type of constraint to be used in conjunction with simplex splines is introduced. This new constraint type are the differential constraints which constrain the directional derivatives of the simplex splines. The differential constraints have many uses, of which the most promising is the facilitating of bounded model extrapolation. Finally, in Sec. 3.5 a new formulation of the B-form in global coordinates is introduced. The Bernstein basis polynomials of the simplex splines are functions in terms of barycentric coordinates which are local coordinates. This means that the polynomial basis functions of the simplex spline are meaningless in the global coordinate system. In some cases, however, it is desirable to analyze a model in terms of the global physical variables. Additionally, a global formulation of the B-form makes it possible to optimize the triangulation and the B-coefficients in a single step, negating the need for separate triangulation optimization.

3.1 The de Casteljau algorithm

The de Casteljau algorithm was developed in 1959 by Paul de Casteljau, a physicist and mathematician working at the French car manufacturer Citroën. De Casteljau invented his algorithm for the computation of Beziér curves which, at the time, were used in the car industry as a method for efficiently storing and rendering curved line and surface data. The principle of the de Casteljau algorithm, presented in Sec. 3.1.1, is the recurrent degree reduction of the polynomial basis functions by shifting, one degree at a time, the polynomial degree of the basis functions to the polynomial degree of the B-coefficients.

In Sec. 3.1.2 a new one-step form of the de Casteljau algorithm is introduced which does not require the recursion of the original algorithm. The one-step form is easily translated in a new matrix form presented in Sec. 3.1.3. This new matrix form will prove to be essential in the formulation of differential constraints on the B-coefficients of the multivariate simplex splines. This one-step matrix form of the de Casteljau algorithm was first introduced in [41].

3.1.1 The ordinary de Casteljau algorithm

The de Casteljau algorithm is a well-known recursive method for iteratively calculating the value of a polynomial in the B-form, see e.g. [53]. A B-form polynomial can be expressed in terms of the m^{th} de Casteljau iteration as follows:

$$p(\mathbf{b}) = \sum_{|\kappa|=d-m} c_\kappa^{(m)}(\mathbf{b}) B_\kappa^{d-m}(\mathbf{b}), \tag{3.1}$$

in which the B-coefficient of iteration m is related to the B-coefficient of iteration $m-1$ as follows:

$$c_\kappa^{(m)}(\mathbf{b}) := \sum_{|\gamma|=1} b_\gamma c_{\kappa+\gamma}^{(m-1)}(\mathbf{b}), \;\; m \leq d, \tag{3.2}$$

with $c_\kappa^{(0)}(b) = c_\kappa$ and with $b_\gamma = b_{\gamma_0} b_{\gamma_1} \cdots b_{\gamma_n}$ a first order polynomial in the barycentric coordinate b. The final value of the polynomial at the evaluation point $\mathbf{b}$ is $c_0^{(d)}(\mathbf{b})$, which is the last iteration of the de Casteljau algorithm.

The de Casteljau algorithm requires d iterations for the evaluation of a B-form polynomial of degree d, which at first may seem inefficient. The de Casteljau algorithm does not require exponential operators, however, and is more stable numerically than direct evaluation, which in the time of 8-bit computers with limited floating point capabilities was an important advantage[1]. At current though, it makes more sense to directly evaluate the B-form polynomial using exponential operators as this requires significantly less computational steps.

Example 6 (The de Casteljau algorithm)**.** *In this example the de Casteljau algorithm is demonstrated by evaluating a bivariate B-form polynomial of degree* $d = 3$ *at the barycentric coordinate* $\mathbf{b} = (b_0, b_1)$*.*

Using Eq. 3.1, the iterations of the de Casteljau algorithm are:

$$\begin{aligned} p(\mathbf{b}) &= \sum_{|\kappa|=3} c_\kappa B_\kappa^3(\mathbf{b}) \\ &= \sum_{|\kappa|=2} c_\kappa^{(1)}(\mathbf{b}) B_\kappa^2(\mathbf{b}) \\ &= \sum_{|\kappa|=1} c_\kappa^{(2)}(\mathbf{b}) B_\kappa^1(\mathbf{b}) \\ &= c_0^{(3)}(\mathbf{b}). \end{aligned}$$

---

[1] Hardware support for floating point mathematical functions has only been present since 1980 when Intel introduced the 8087 mathematical co-processor [174].

Using Eq. 3.2 to expand $c_\kappa^{(1)}(\mathbf{b})$*, with* $\kappa \in \{(2,0),(1,1),(0,2)\}$ *we get:*

$$\begin{aligned} c_{2,0}^{(1)}(\mathbf{b}) &= b_0c_{3,0} + b_1c_{2,1} \\ c_{1,1}^{(1)}(\mathbf{b}) &= b_0c_{2,1} + b_1c_{1,2} \\ c_{0,2}^{(1)}(\mathbf{b}) &= b_0c_{1,2} + b_1c_{0,3}. \end{aligned}$$

For $c_\kappa^{(2)}(\mathbf{b})$ *and* $\kappa \in \{(1,0),(0,1)\}$ *we find:*

$$\begin{aligned} c_{1,0}^{(2)}(\mathbf{b}) &= b_0c_{2,0}^{(1)}(\mathbf{b}) + b_1c_{1,1}^{(1)}(\mathbf{b}) \\ &= b_0(b_0c_{3,0} + b_1c_{2,1}) + b_1(b_0c_{2,1} + b_1c_{1,2}) \\ c_{0,1}^{(2)}(\mathbf{b}) &= b_0c_{1,1}^{(1)}(\mathbf{b}) + b_1c_{0,2}^{(1)}(\mathbf{b}) \\ &= b_0(b_0c_{2,1} + b_1c_{1,2}) + b_1(b_0c_{1,2} + b_1c_{0,3}). \end{aligned}$$

Finally, for $c_\kappa^{(3)}(\mathbf{b})$ *and* $\kappa = \{0\}$ *we find*

$$\begin{aligned} c_0^{(3)}(\mathbf{b}) &= b_0c_{1,0}^{(2)}(\mathbf{b}) + b_1c_{0,1}^{(2)}(\mathbf{b}) \\ &= b_0(b_0c_{2,0}^{(1)}(\mathbf{b}) + b_1c_{1,1}^{(1)}(\mathbf{b})) + b_1(b_0c_{1,1}^{(1)}(\mathbf{b}) + b_1c_{0,2}^{(1)}(\mathbf{b})) \\ &= b_0(b_0(b_0c_{3,0} + b_1c_{2,1}) + b_1(b_0c_{2,1} + b_1c_{1,2})) + \\ &\quad + b_1(b_0(b_0c_{2,1} + b_1c_{1,2}) + b_1(b_0c_{1,2} + b_1c_{0,3})) \\ &= b_0^3c_{3,0} + 3b_0^2b_1c_{2,1} + 3b_0b^2c_{1,2} + b_1^3c_{0,3}. \end{aligned}$$

The final step in the de Casteljau algorithm results in the full expanded form of the B-form polynomial. ■

3.1.2 A multi-degree formulation of the de Casteljau recursion

In the following sections, the formulation of the multi-degree form of the de Casteljau algorithm as first derived in [41] is presented.

First, notice that the polynomial basis function $B_\kappa^d(b)$ from Eq. 2.52 can be expanded as follows:

$$\begin{aligned} B_\kappa^d(\mathbf{b}) &= b_0B^{d-1}_{\kappa_0-1,\kappa_1,\dots,\kappa_n}(\mathbf{b}) + b_1B^{d-1}_{\kappa_0,\kappa_1-1,\dots,\kappa_n}(\mathbf{b}) + \cdots + \\ &\quad b_nB^{d-1}_{\kappa_0,\kappa_1,\dots,\kappa_n-1}(\mathbf{b}),\ \ |\kappa| = d \\ &= \sum_{|\gamma|=1} b_\gamma B^{d-1}_{\kappa-\gamma}(\mathbf{b}). \end{aligned} \tag{3.3}$$

with $\gamma = (\gamma_0, \gamma_1, \dots, \gamma_n)$ a multi-index dependent on κ as follows:

$$|\gamma| = d - |\kappa| \tag{3.4}$$

In general, for any degree $m \leq d$ we have for a single basis function term:

$$\begin{aligned} B_{\kappa}^{d}(\mathbf{b}) &= P_{\gamma}^{m}(\mathbf{b}) B_{\kappa_0 - m, \kappa_1, \ldots, \kappa_n}^{d-m}(\mathbf{b}) + P_{\gamma}^{m}(\mathbf{b}) B_{\kappa_0, \kappa_1 - m, \ldots, \kappa_n}^{d-m}(\mathbf{b}) + \cdots + \\ & \quad P_{\gamma}^{m}(\mathbf{b}) B_{\kappa_0, \kappa_1, \ldots, \kappa_n - m}^{d-m}(\mathbf{b}) \\ &= \sum_{|\gamma| = m} P_{\gamma}^{m}(\mathbf{b}) B_{\kappa - \gamma}^{d-m}(\mathbf{b}). \end{aligned} \tag{3.5}$$

with the m^{th} degree basis function $P_{\gamma}^{m}(\mathbf{b})$ defined as follows:

$$P_{\gamma}^{m}(\mathbf{b}) = \frac{m!}{\gamma!} b_{\gamma}^{m}. \tag{3.6}$$

It is important to note that any polynomial terms with negative multi-indices are defined to be equal to zero.

Example 7 (Using the generalized de Casteljau algorithm)**.** *Use Eq. 3.5 to expand* $B_{2,1,0}^{3}(\mathbf{b})$ *with* $m = 2$*.*

In this case we have $\kappa = (2,1,0)$ *and* $|\gamma| = 2$*. Therefore, we have for* γ*:*

$$\gamma \in \{(2,0,0), (1,1,0), (1,0,1), (0,2,0), (0,1,1), (0,0,2)\}$$

using Eq. 3.5, $B_{2,1,0}^{3}(\mathbf{b})$ *can be expanded as follows:*

$$\begin{aligned} B_{2,1,0}^{3}(\mathbf{b}) &= P_{2,0,0}^{2}(\mathbf{b}) B_{0,1,0}^{1}(\mathbf{b}) + P_{1,1,0}^{2}(\mathbf{b}) B_{1,0,0}^{1}(\mathbf{b}) + P_{1,0,1}^{2}(\mathbf{b}) B_{1,1,-1}^{1}(\mathbf{b}) + \\ & \quad + P_{0,2,0}^{2}(\mathbf{b}) B_{2,-1,0}^{1}(\mathbf{b}) + P_{0,1,1}^{2}(\mathbf{b}) B_{2,0,-1}^{1}(\mathbf{b}) + P_{0,0,2}^{2}(\mathbf{b}) B_{2,1,-2}^{1}(\mathbf{b}), \\ &= P_{2,0,0}^{2}(\mathbf{b}) B_{0,1,0}^{1}(\mathbf{b}) + P_{1,1,0}^{2}(\mathbf{b}) B_{1,0,0}^{1}(\mathbf{b}), \end{aligned}$$

where all basis functions with negative multi-index components have been dropped. ■

Substitution of Eq. 3.5 and Eq. 3.6 in the B-form from Eq. 2.59 results in the following expression:

$$p(\mathbf{b}) = \sum_{|\kappa| = d} \left[c_{\kappa} \sum_{|\gamma| = m} P_{\gamma}^{m}(\mathbf{b}) B_{\kappa - \gamma}^{d-m}(\mathbf{b}) \right]. \tag{3.7}$$

When we let $|\kappa| = d - m$ Eq. 3.7 can be reformulated such that no negative multi-indices are produced:

$$p(\mathbf{b}) = \sum_{|\kappa| = d-m} \sum_{|\gamma| = m} c_{\kappa + \gamma} P_{\gamma}^{m}(\mathbf{b}) B_{\kappa}^{d-m}(\mathbf{b}). \tag{3.8}$$

We will now introduce a theorem for multi-degree de Casteljau algorithm, which is necessary for the definition of the one-step de Casteljau matrix.

Theorem 2. *The multi-degree de Casteljau algorithm, which relates the B-coefficients of iteration* q *with those of iteration* $m+q$ *is:*

$$c_{\kappa}^{(q+m)}(\mathbf{b}) = \sum_{|\gamma|=m} P_{\gamma}^{m}(\mathbf{b}) c_{\kappa+\gamma}^{(q)}(\mathbf{b}), \;\; q \geq 0, \tag{3.9}$$

with $c_{\kappa}^{(0)}(\mathbf{b}) = c_{\kappa}$.

Proof. Let $\delta^i, i = 1, 2, \ldots, m$ be a set of q independent multi-indices. The ordinary de Casteljau algorithm then is:

$$c_{\kappa}^{(q+1)}(\mathbf{b}) \;\; = \sum_{|\delta^1|=1} b_{\delta^1} c_{\kappa+\delta^1}^{(q)}(\mathbf{b}), \;\; q \geq 0. \tag{3.10}$$

The next iteration then is

$$\begin{aligned} c_{\kappa}^{(q+2)}(\mathbf{b}) &= \sum_{|\delta^2|=1} b_{\delta^2} c_{\kappa+\delta^2}^{(q+1)}(\mathbf{b}) \\ &= \sum_{|\delta^2|=1} b_{\delta^2} \sum_{|\delta^1|=1} b_{\delta^1} c_{\kappa+\delta^1+\delta^2}^{(q)}(\mathbf{b}). \end{aligned} \tag{3.11}$$

Repeating this process and letting $\gamma = \sum_{i=1}^{m} \delta^i$ we get for the m^{th} iteration:

$$c_{\kappa}^{(q+m)}(\mathbf{b}) \;\; = \sum_{|\delta^m|=1} b_{\delta^m} \sum_{|\delta^{m-1}|=1} b_{\delta^{m-1}} \cdots \sum_{|\delta^1|=1} b_{\delta^1} c_{\kappa+\gamma}^{(q)}(\mathbf{b}). \tag{3.12}$$

Using the multinomial theorem the product of sums can be rewritten as follows:

$$c_{\kappa}^{(q+m)}(\mathbf{b}) \;\; = \sum_{|\gamma|=m} \frac{d!}{\gamma!} b_{\gamma}^{m} c_{\kappa+\gamma}^{(q)}(\mathbf{b}) \tag{3.13}$$

$$= \sum_{|\gamma|=m} P_{\gamma}^{m}(\mathbf{b}) c_{\kappa+\gamma}^{(q)}(\mathbf{b}), \tag{3.14}$$

the last equality follows from the definition of $P_{\gamma}^{m}(\mathbf{b})$ from Eq. 3.6 and proves the theorem.
□

3.1.3 A one-step matrix form of the de Casteljau algorithm

All theory is now in place to be able to present the general, one-step matrix form of the de Casteljau algorithm from [41]. The de Casteljau matrix function of degree m, which reduces a set of B-coefficients of degree d into a set of B-coefficients of degree $d-m$, is defined as follows:

$$\mathbf{P}^{d,d-m}(\mathbf{b}) \in \mathbb{R}^{\hat{d}^* \times \hat{d}}, \tag{3.15}$$

with $\hat{d}^*$ the total number of basis function terms for degree $d-m$ and dimension n:

$$\hat{d}^* = \frac{(d-m+n)!}{(d-m)!n!}. \tag{3.16}$$

Example 8 (Using the de Casteljau matrix form)**.** *As an example of the use of Eq. 3.15, consider the case of the de Casteljau matrix* $\mathbf{P}^{3,1}(\mathbf{b})$ *for* $n = 1$*. In this case, the de Casteljau matrix reduces the degree of an original set of B-coefficients from degree* 3 *to degree* 1 *as follows:*

$$\begin{bmatrix} c_{1,0}(\mathbf{b}) & c_{0,1}(\mathbf{b}) \end{bmatrix}^\top = \mathbf{P}^{3,1}(\mathbf{b}) \cdot \begin{bmatrix} c_{3,0} & c_{2,1} & c_{1,2} & c_{0,3} \end{bmatrix}^\top.$$

■

The de Casteljau matrix has the following structure:

$$\left[\mathbf{P}^{d,d-m}(\mathbf{b})\right]_{i(\kappa),i(\theta)} = P^m_{\theta-\kappa}(\mathbf{b}), \ |\theta| = d, |\kappa| = d-m, \tag{3.17}$$

with $i(\kappa)$ and $i(\theta)$ index functions for the rows and columns of $\mathbf{P}^{d,d-m}(\mathbf{b})$, respectively. The index function $i(\kappa)$ for the rows of the de Casteljau matrix is defined as follows:

$$i(\kappa) := \sum_{|\gamma|=d-m} 1, \ \gamma \leq \kappa, \ |\kappa| = d-m, \tag{3.18}$$

while the index function $i(\theta)$ for the columns of the de Casteljau matrix has the following definition:

$$i(\theta) := \sum_{|\gamma|=d} 1, \ \gamma \leq \theta, \ |\theta| = d. \tag{3.19}$$

Example 9 (Using index functions)**.** *In this example the concept of the index functions from Eq. 3.18 and Eq. 3.19 will be demonstrated. We want to find the value of* $i(\theta)$ *for* $\theta = (1,3)$ *for a bivariate* 4^{th} *degree B-form polynomial. For a bivariate* 4^{th} *degree polynomial we have the following values for* θ*:* $\theta \in \{(4,0),(3,1),(2,2),(1,3),(0,4)\}$*. we then find for Eq. 3.19:*

$$\begin{aligned} i(\theta) &= \sum_{|\gamma|=4} 1, \ \gamma \leq (1,3), \ |\theta| = 4, \\ &= 1+1+1+1, \end{aligned}$$

so $i(1,3) = 4$. ■

Note that basis polynomials with negative multi-indices can be (and are) produced by Eq. 3.17. These basis polynomials are again zero by definition.

Before a vector form of the B-form based on the de Casteljau matrix can be derived, a proof is required for the following theorem.

Theorem 3. *Based on the results from Theorem 2, the following statement must hold for the de Casteljau matrix from Eq. 3.17:*

$$\mathbf{P}^{d,d-m}(\mathbf{b}) \cdot \mathbf{c}^{t_j} = \left[\sum_{|\gamma|=m} P^m_\gamma(\mathbf{b}) c_{\kappa+\gamma} \right]_{|\kappa+\gamma|=d}, \tag{3.20}$$

where the right hand term of this statement follows from Eq. 3.9 using $q = 0$.

Proof. The proof starts by reformulating Eq. 3.20 in a form that permits negative multi-indices. For this, we let $\theta = \kappa + \gamma$ such that $|\theta| = |\kappa + \gamma| = d$ and $\gamma = \theta - \kappa$. Substitution of θ in the right hand side of Eq. 3.20 then results in:

$$\mathbf{P}^{d,d-m}(\mathbf{b}) \cdot \mathbf{c}^{t_j} = \left[\sum_{|\theta|=d} P^m_{\theta-\kappa}(\mathbf{b}) c_\theta \right]_{|\kappa|=d-m}. \tag{3.21}$$

Using the definition of the structure of $\mathbf{P}^{d,d-m}(\mathbf{b})$ from Eq. 3.17 and the definitions of the index functions from Eq. 3.18 and Eq. 3.19, the following must hold for every single element of the de Casteljau matrix, and thus for every value of κ and θ:

$$\begin{aligned} \left[\mathbf{P}^{d,d-m}(\mathbf{b})\right]_{i(\kappa),i(\theta)} \cdot \mathbf{c}^{t_j}_{i(\theta)} &= P^m_{\theta-\kappa}(\mathbf{b}) \cdot \mathbf{c}^{t_j}_{i(\theta)}, \\ &= P^m_{\theta-\kappa}(\mathbf{b}) \cdot c_\theta, \end{aligned} \tag{3.22}$$

as long as $|\theta| = d$ and $|\kappa| = d - m$. Therefore, we get for the multiplication of a single row of the de Casteljau matrix with the vector of B-coefficients:

$$\left[\mathbf{P}^{d,d-m}(\mathbf{b})\right]_{i(\kappa),\bullet} \cdot \mathbf{c}^{t_j} = \sum_{|\theta|=d} P^m_{\theta-\kappa}(\mathbf{b}) c_\theta. \tag{3.23}$$

Finally, the multiplication of all rows of the de Casteljau matrix with the vector of B-coefficients immediately results in Eq. 3.21, thus proving the theorem.

□

With Theorem 3 a vector form of the B-form using the de Casteljau matrix can be defined. This vector form is essential for defining directional derivatives in terms of the original global vector of B-coefficients, as will be shown in Sec. 3.4. Substitution of the

B-coefficient vector from Eq. 2.64, the vector form of the basis function from Eq. 2.66, and the de Casteljau matrix from Eq. 3.17 in Eq. 3.8 results in the vector form of the de Casteljau B-form of degree d on a single simplex t_j:

$$p(\mathbf{b}) = \mathbf{B}^{d-m}(\mathbf{b})\mathbf{P}^{m,d-m}(\mathbf{b}) \cdot \mathbf{c}^{t_j}. \tag{3.24}$$

It is easy to check that this equation reduces to the ordinary vector form of the B-form from Eq. 2.67 by letting $m = 0$.

Example 10 (Constructing the de Casteljau matrix)**.** *Use Eq. 3.17 to construct the de Casteljau matrix* $\mathbf{P}^{3,1}(\mathbf{b})$ *that reduces a set of B-coefficients of degree* $d = 3$ *to degree* $d - m = 1$*, such that* $m = 2$*. The B-form polynomial in this case is a bivariate polynomial in barycentric coordinates.*

In this case we have for κ*:*

$$\kappa \in \{(1,0),(0,1)\}, \ \ |\kappa| = d - m.$$

The index function $i(\kappa)$ *from Eq. 3.18 for the rows of* $\mathbf{P}^{3,1}(\mathbf{b})$ *therefore has the following values:*

$$i(1,0) = 1, \ \ i(0,1) = 2.$$

The valid values for the multi-index θ *are:*

$$\theta \in \{(3,0),(2,1),(1,2),(0,3)\}, \ \ |\theta| = d,$$

with which the index function $i(\theta)$ *from Eq. 3.19 for the columns of* $\mathbf{P}^{3,1}(\mathbf{b})$ *takes the following values:*

$$i(3,0) = 1, \ \ i(2,1) = 2, \ \ i(1,2) = 3, \ \ i(3,0) = 4.$$

We are now ready to construct the rows and columns of $\mathbf{P}^{3,1}(\mathbf{b})$ *using Eq. 3.17. Expanding the row and column indices generated by* $i(\kappa)$ *and* $i(\theta)$*, and calculating the multi-index of the corresponding polynomials, we get:*

$i(\kappa)$	$i(\theta)$	θ	κ	$\theta - \kappa$	$P^m_{\theta-\kappa}(\mathbf{b})$
1	1	$(3,0)$	$(1,0)$	$(2,0)$	$P^2_{2,0}(\mathbf{b})$
1	2	$(2,1)$	$(1,0)$	$(1,1)$	$P^2_{1,1}(\mathbf{b})$
1	3	$(1,2)$	$(1,0)$	$(0,2)$	$P^2_{0,2}(\mathbf{b})$
1	4	$(0,3)$	$(1,0)$	$(-1,3)$	0
2	1	$(3,0)$	$(0,1)$	$(3,-1)$	0
2	2	$(2,1)$	$(0,1)$	$(2,0)$	$P^2_{2,0}(\mathbf{b})$
2	3	$(1,2)$	$(0,1)$	$(1,1)$	$P^2_{1,1}(\mathbf{b})$
2	4	$(0,3)$	$(0,1)$	$(0,2)$	$P^2_{0,2}(\mathbf{b})$

By filling in the polynomial values of the rightmost column in the de Casteljau matrix at the row and column indices provided by the index functions $i(\kappa)$ and $i(\gamma)$, the following result is obtained:

$$\mathbf{P}^{3,1}(\mathbf{b}) = \left[\begin{array}{cccc} P^2_{2,0}(\mathbf{b}) & P^2_{1,1}(\mathbf{b}) & P^2_{0,2}(\mathbf{b}) & 0 \\ 0 & P^2_{2,0}(\mathbf{b}) & P^2_{1,1}(\mathbf{b}) & P^2_{0,2}(\mathbf{b}) \end{array} \right].$$

■

The de Casteljau matrix $\mathbf{P}$ for a complete triangulation is constructed by placing per-simplex de Casteljau matrix blocks on the main diagonal of $\mathbf{P}$:

$$\mathbf{P} = \text{diag}\left(\left[\mathbf{P}^{d,d-m}(\mathbf{b})\right]_j, j = 1, 2, \ldots, J \right). \tag{3.25}$$

3.2 Calculations with B-form Polynomials

In his influential 1987 publication, the spline theoretician Carl de Boor stated that "*[The n-variate formulation of the B-form] forces careful consideration of notation and brings out the essential mathematical aspects and surprising beauty of the B-form.*" [31]. This statement is very much true; the elegance of the B-form is not just limited to its formulation, it also extends to many aspects of operations with B-form polynomials, such as directional derivatives and integrals.

In this section the most important operations that can be performed with polynomials in the B-form are discussed. In Sec. 3.2.1 a full-triangulation formulation of the B-form will be provided. This formulation will prove to be instrumental in all of the following. In Sec. 3.2.2, Sec. 3.2.3, Sec. 3.2.4 and Sec. 3.2.5 generalizations to the n-dimensional case of expressions for respectively the sum, the integral, the inner product, and the degree raising of B-form polynomials are presented. A new formulation for the directional derivatives of B-form polynomials in terms of the original vector of B-coefficients is then presented in Sec. 3.2.6. Finally, in Sec. 3.2.7 bounds on the accuracy of spline model approximations are given.

3.2.1 Full triangulation vector form of B-form polynomials

The vector form of B-form polynomials on a single simplex was already discussed in Sec. 2.2.5. In the following, this vector form will be used in the formulation of an expression for B-form polynomials defined on complete triangulations. In this thesis, such a full-triangulation form of the B-form is called the *global B-form*. The global B-form presented below was first derived in [40], where it was used in the definition of a linear regression scheme for simplex splines.

First, let $\mathcal{T}$ be a triangulation consisting of J simplices. The full-triangulation basis function vector $\mathbf{B}^d(b(\mathbf{x}))$ for a single point $\mathbf{x}$ in Cartesian coordinates then is defined as follows:

$$\mathbf{B}^d(b(\mathbf{x})) := [\ \mathbf{B}^d_{t_1}(b(\mathbf{x})) \quad \mathbf{B}^d_{t_2}(b(\mathbf{x})) \quad \cdots \quad \mathbf{B}^d_{t_J}(b(\mathbf{x}))\ ] \in \mathbf{R}^{1\times J\cdot\hat{d}}, \tag{3.26}$$

with $\mathbf{B}^d_{t_j}(b(\mathbf{x}))$ the vectors of basis polynomials as defined in Eq. 2.66, and with $b(\mathbf{x})$ the implicit barycentric coordinate transformation of the point $\mathbf{x}$ with respect to the simplex t_j as defined in Eq. 2.34.

Using the per-simplex vector of lexicographically sorted B-coefficients $\mathbf{c}^{t_j}$ from Eq. 2.64, the full triangulation vector of B-coefficients is constructed as follows:

$$\mathbf{c} = \left[\mathbf{c}^{t_j}\right]_{j=1}^{J} \in \mathbf{R}^{J\cdot\hat{d}\times 1}. \tag{3.27}$$

A non-local, full triangulation vector form of the B-form would then be given as:

$$P(b(\mathbf{x})) = \mathbf{B}^d(b(\mathbf{x}))\cdot\mathbf{c}, \tag{3.28}$$

While technically correct, the above expression would not lead to a meaningful piecewise defined polynomial because it does not consider the fact that a point $\mathbf{x}$ is in general contained by only a single simplex[2]. The barycentric coordinate transformation $b(\mathbf{x})$ of $\mathbf{x}$ with respect to the simplex t_j is defined regardless of $\mathbf{x}$ being located inside or outside of t_j, the only difference being that in the latter case $b(\mathbf{x})$ has at least one negative component, see Eq. 2.43.

Because $b(\mathbf{x})$ is defined for all simplices, an evaluation of Eq. 3.28 at $\mathbf{x}$ would have to involve the evaluation of *all* B-form basis functions of the complete triangulation, even though the point $\mathbf{x}$ is located in a single simplex. With Eq. 3.28 we have effectively destroyed the local basis function property of the multivariate simplex spline, thereby negating one of the main advantages of simplex splines over other approximation methods.

In order to transform Eq. 3.28 into a local-global form in which the actual location of data is taken into account, an additional entity is required in the form of the data sifting matrix (DSM). The function of the DSM is to assign a location $\mathbf{x}$ to the simplex in which it is contained, such that $b(\mathbf{x})$ is strictly positive. Before being able to define the DSM, the simplex membership operator $\delta_{jk(\mathbf{x})}$ must be defined:

$$\delta_{jk(\mathbf{x})} = \begin{cases} 1, & \text{if } j = k(\mathbf{x}) \\ 0, & \text{if } j \neq k(\mathbf{x}) \end{cases}, \tag{3.29}$$

---

[2] This statement holds only for scattered datasets in which the data or evaluation points are not ordered in any way. When a dataset has a gridded nature, there is a significant probability that some points are located exactly on vertices or simplex edges, causing them to be contained by more than one simplex.

with $k(\mathbf{x})$ an index function[3] which produces the index of the n-simplex which contains $\mathbf{x}$ as follows:

$$\mathbf{x} \in t_{k(\mathbf{x})},\ \forall \mathbf{x} \in \mathcal{X}, \tag{3.30}$$

with $\mathcal{X} \in \mathbb{R}^n$ the set of all data points.

In short, $\delta_{jk(\mathbf{x})}$ is a switch that is equal to 1 only when j equals the index $k(\mathbf{x})$, which is the case when $\mathbf{x}$ is located inside $t_{k(\mathbf{x})}$. When $\mathbf{x}$ is located outside of $t_{k(\mathbf{x})}$ then $\delta_{jk(\mathbf{x})}$ is equal to 0.

Using Eq. 3.29 the per-simplex $\hat{d} \times \hat{d}$ diagonal data membership matrix for the point $\mathbf{x}$ can be defined as follows:

$$\mathbf{D}_{t_j}(\mathbf{x}) = \left[(\delta_{j,k(\mathbf{x})})_{q,q}\right]_{q=1}^{\hat{d}} \in \mathbf{R}^{\hat{d} \times \hat{d}} \tag{3.31}$$

The block diagonal full-triangulation data membership matrix $\mathbf{D}(\mathbf{x})$ for a single observation is a matrix with $\mathbf{D}_{t_j}(\mathbf{x})$ blocks on the main diagonal:

$$\mathbf{D}(\mathbf{x}) = \left[\left(\mathbf{D}_{t_j}(\mathbf{x})\right)_{j,j}\right]_{j=1}^{J} \in \mathbf{R}^{(J \cdot \hat{d}) \times (J \cdot \hat{d})} \tag{3.32}$$

Using Eq. 3.27, Eq. 3.26 and Eq. 3.32 the global B-form of the multivariate simplex spline for the complete triangulation in vector form becomes:

$$P(\mathbf{b}) = \mathbf{B}^d(b(\mathbf{x})) \cdot \mathbf{D}(\mathbf{x}) \cdot \mathbf{c} \tag{3.33}$$

In general, the global DSM matrix $\mathbf{D}(\mathbf{x})$ is rarely explicitly defined or calculated. Instead, the points $\mathbf{x}$ are assigned to their parent simplices during the Cartesian to barycentric coordinate transformation. Therefore, in most cases the evaluation of the global B-form polynomial only requires the evaluation of the B-form polynomial on simplex t_j only for points located within t_j.

3.2.2 Sums of B-form polynomials

B-form polynomials are linear in the parameters, which means that any two B-form polynomials $p(\mathbf{b})$ and $q(\mathbf{b})$ of equal degree d and dimension n can be summed as follows:

$$\begin{aligned} p(\mathbf{b}) + q(\mathbf{b}) &= \sum_{|\kappa|=d} c_\kappa B_\kappa^d(\mathbf{b}) + \sum_{|\kappa|=d} d_\kappa B_\kappa^d(\mathbf{b}) \\ &= \sum_{|\kappa|=d} (c_\kappa + d_\kappa) B_\kappa^d(\mathbf{b}). \end{aligned} \tag{3.34}$$

---

[3] The algorithmic analogy of the simplex membership operator is a data membership search like the function *tsearchn* in Matlab.

Note that while the sum of any two B-form polynomials can be taken as long as their dimension and degree are equal, no meaningful results are obtained when they do not share the same triangulation.

3.2.3 Integrals of B-form polynomials

The integral of the B-form polynomial $p(\mathbf{b})$ of degree d and dimension n over any n-simplex t is given by:

$$\int_t p(\mathbf{b}) d\mathbf{b} = \frac{A_t}{\hat{d}} \sum_{|\kappa|=d} c_\kappa. \tag{3.35}$$

with A_t the volume of the n-simplex, which was given by Eq. 2.30. Interestingly, the analytical integral of a B-form polynomial is simply the sum of the B-coefficients multiplied with a constant!

Calculation of the numerical value of the full-triangulation integral of a spline function is quite trivial:

$$\sum_{j=1}^{J} \int_{t_j} p(\mathbf{b}) d\mathbf{b} = \sum_{j=1}^{J} \frac{A_{t_j}}{\hat{d}} \sum_{|\kappa|=d} c_\kappa^{t_j}. \tag{3.36}$$

3.2.4 Inner products of B-form polynomials

The inner product of two B-form polynomials is also relatively simple, and well known in the literature, see e.g. Lai and Schumaker [104]. If the two B-form polynomials of degree d and dimension n are given by:

$$p(\mathbf{b}) = \sum_{|\kappa|=d} c_\kappa B_\kappa^d(\mathbf{b}); \quad q(\mathbf{b}) = \sum_{|\gamma|=d} \tilde{c}_\gamma B_\gamma^d(\mathbf{b}),$$

then their inner product is given by:

$$\int_{t_j} p(\mathbf{b}) q(\mathbf{b}) d\mathbf{b} = \frac{A_t}{\begin{pmatrix} dn \\ d \end{pmatrix} \begin{pmatrix} nd+n \\ n \end{pmatrix}} \sum_{\substack{|\kappa| = d \\ |\gamma| = d}} c_\kappa \tilde{c}_\gamma \prod_{i=0}^{n} \begin{pmatrix} \kappa_i + \gamma_i \\ \kappa_i \end{pmatrix}. \tag{3.37}$$

3.2.5 Degree raising

Any B-form polynomial $p(\mathbf{b})$ of degree d can be transformed into a polynomial of higher degree $\tilde{d}$ with $\tilde{d} > d$ in a process called 'degree raising' . Degree raising can be useful when one wants to upgrade the approximation power of a spline model without influencing the existing model. The formula for raising by a single degree $p(\mathbf{b})$ from degree d to degree $d+1$ is:

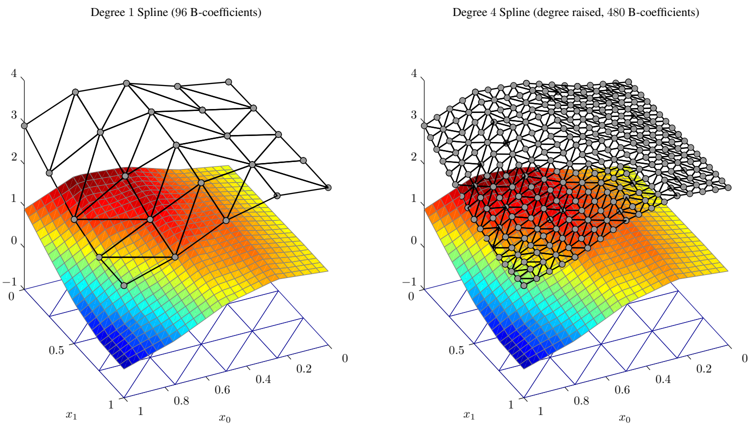


Figure 3.1: Spline function of degree 1 with B-net consisting of 96 B-coefficients (left), which after degree raising to degree 4 has a B-net consisting of 480 B-coefficients (right). The resulting spline function (colored surface) resulting from both B-nets is artificially lowered in both plots to aid visibility.

$$p(\mathbf{b}) = \sum_{|\kappa|=d+1} c_\kappa^{[d+1]} B_\kappa^{d+1}(\mathbf{b}), \tag{3.38}$$

with $c_\kappa^{[d+1]}$ the degree raised B-coefficients calculated as follows:

$$c_\kappa^{[d+1]} = \frac{1}{d+1} \sum_{\substack{j=0\\|\gamma|=1}}^{j=n} \kappa_j c_{\kappa-\gamma}^{[d]}. \tag{3.39}$$

Note that the degree raised B-coefficients are not to be confused with the B-coefficients produced with the de Casteljau algorithm. Multiple degree raising can be achieved by iteratively executing Eq. 3.39.

In Figure 3.1 the degree raising process is demonstrated. In the left plot of Figure 3.1, a first degree (linear) spline function is shown, together with its B-net consisting of 96 B-coefficient. Using the degree raising process, the degree of the spline function is increased to a fourth order spline function, which is shown in the right hand plot of Figure 3.1. Notice that both spline functions produce exactly the same results, only the density of the B-net has been affected by the degree raising.

3.2.6 Directional derivatives of B-form polynomials

The de Casteljau matrix, which was introduced in Sec. 3.1.3, allows for a very elegant formulation of the directional derivatives of B-form polynomials. This formulation comes in the form of a matrix equation, and was first introduced in [41]. Before being able to introduce the new formulation, the concept of the directional derivatives must first be introduced. The directional derivatives of B-form polynomials are well known in the literature, see e.g. Lai and Schumaker [104].

First, let $f(\mathbf{x})$ be some differential function in $\mathbf{x} \in \mathbb{R}^n$, and let $\mathbf{u}$ be a directional vector in Euclidean $\mathbb{R}^n$. The directional derivative of $f(\mathbf{x})$ at $\mathbf{x}$ in the direction $\mathbf{u}$ is defined as follows:

$$D_{\mathbf{u}} f(\mathbf{x}) := \frac{d}{dt} f(\mathbf{x} + t\mathbf{u}). \tag{3.40}$$

Note that both $\mathbf{x}$ and $\mathbf{u}$ are global Cartesian coordinates. Because we are interested in the directional derivatives of B-form polynomials in barycentric coordinates, a definition of the directional derivative in barycentric coordinates is required. Therefore, let $b(\mathbf{x}) \in \mathbb{R}^{n+1}$ be the barycentric coordinate of $\mathbf{x}$. Because $\mathbf{u}$ is a vector, and not a point, a direct translation into barycentric coordinates cannot be made. Instead, $\mathbf{u}$ can be described as the difference vector of two points $\mathbf{v}$ and $\mathbf{w}$ in Euclidean $\mathbb{R}^n$ as follows:

$$\mathbf{u} = \mathbf{v} - \mathbf{w} \quad \in \mathbb{R}^n. \tag{3.41}$$

The barycentric coordinates of $\mathbf{v}$ and $\mathbf{w}$ can be calculated directly; let $b(\mathbf{v})$ be the barycentric coordinate of $\mathbf{v}$, while $b(\mathbf{w})$ is the barycentric coordinate of $\mathbf{w}$. The barycentric representation of $\mathbf{u}$ then is:

$$b(\mathbf{u}) = \mathbf{a} = b(\mathbf{v}) - b(\mathbf{w}) \quad \in \mathbb{R}^{n+1} \tag{3.42}$$

The vector $\mathbf{a}$ is called the *directional coordinate* of $\mathbf{u}$, see e.g. [104]. Note that $|\mathbf{a}| = 0$ as it is the difference of two normalized barycentric coordinates.

From the literature, the first order derivative in the direction of a is found to be:

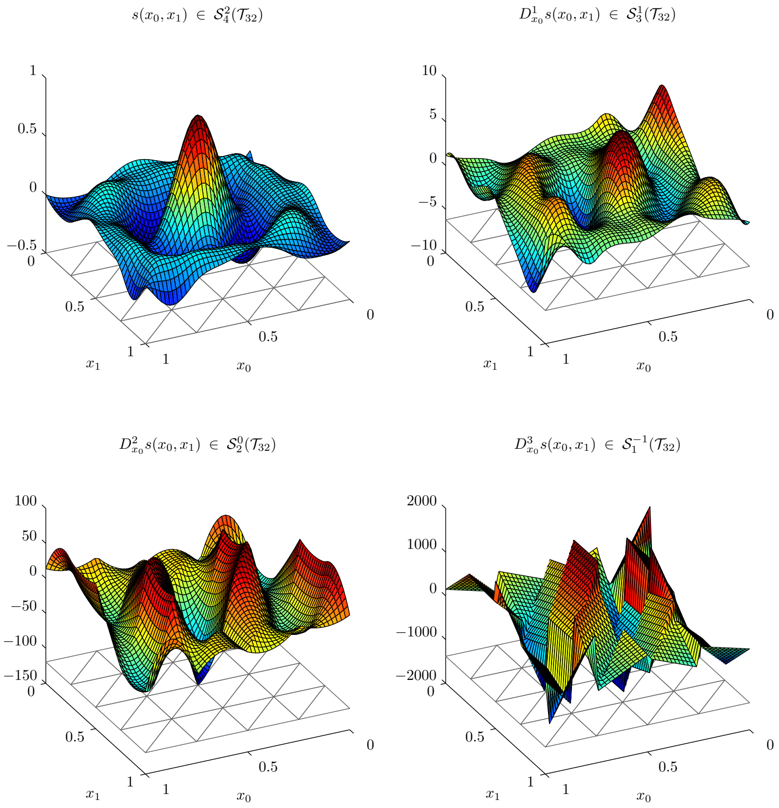
$$D_{\mathbf{u}} p(\mathbf{b}) = d \sum_{|\kappa|=d-1} c_\kappa^{(1)}(\mathbf{a}) B_\kappa^{d-1}(\mathbf{b}), \tag{3.43}$$

with $c_\kappa^{(1)}(\mathbf{a})$ the first de Casteljau iteration of the B-coefficients c_κ. Higher order derivatives are given by:

$$D_{\mathbf{u}}^m p(\mathbf{b}) = \frac{d!}{(d-m)!} \sum_{|\kappa|=d-m} c_\kappa^{(m)}(\mathbf{a}) B_\kappa^{d-m}(\mathbf{b}), \tag{3.44}$$

with $c_\kappa^{(m)}(\mathbf{a})$ the m^{th} de Casteljau iteration.

In Figure 3.2 the use of the directional derivative is demonstrated. In the figure, a fourth



$s(x_0, x_1) \in \mathcal{S}_4^2(\mathcal{T}_{32})$

$D_{x_0}^1 s(x_0, x_1) \in \mathcal{S}_3^1(\mathcal{T}_{32})$

$D_{x_0}^2 s(x_0, x_1) \in \mathcal{S}_2^0(\mathcal{T}_{32})$

$D_{x_0}^3 s(x_0, x_1) \in \mathcal{S}_1^{-1}(\mathcal{T}_{32})$

Figure 3.2: A fourth degree spline function with continuity order C^2 approximating the Mexican hat function, together with its first three directional derivatives in the x_0 direction.

degree spline function with C^2 continuity approximates the Mexican hat function. The first three directional derivatives of the spline function in the direction x_0 are also plotted. Notice that the first and second order directional derivatives are spline functions of continuity order C^1 and C^0, respectively. The third order directional derivative, however, is a discontinuous linear function.

In section Sec. 3.1 a one-step matrix form of the de Casteljau algorithm was presented. It was proved in [41] that this matrix form allows the directional derivatives of a B-form polynomial of order m to be expressed in terms of the original B-coefficient vector rather than in the m^{th} de Casteljau iteration of the original B-coefficients.

Theorem 4. *The matrix form of the directional derivative of order m of the B-form polynomial $p(\mathbf{b})$ in the direction $\mathbf{u}$ in terms of the original B-coefficients is:*

$$D_{\mathbf{u}}^m p(\mathbf{b}) = \frac{d!}{(d-m)!}\mathbf{B}^{d-m}(\mathbf{b})\mathbf{P}^{d,d-m}(\mathbf{a}) \cdot \mathbf{c}^{t_j}, \tag{3.45}$$

with $\mathbf{P}^{d,d-m}(\mathbf{a})$ the de Casteljau matrix from Eq. 3.20 of degree d to $d-m$ expressed in terms of the directional coordinate $\mathbf{a}$ of $\mathbf{u}$, with $\mathbf{B}^{d-m}(\mathbf{b})$ the vector form of the basis polynomials, and with $\mathbf{c}^{t_j}$ the vector of B-coefficients for a single simplex t_j.

Proof. Starting with the m^{th} order directional derivative of the B-form from Eq. 3.44, and substituting the multi-degree de Casteljau form from Eq. 3.9 with $q = 0$ and the directional coordinate $\mathbf{a}$ we get:

$$D_{\mathbf{u}}^m p(\mathbf{b}) = \frac{d!}{(d-m)!}\sum_{|\kappa|=d-m}\Big(\sum_{|\gamma|=m} P_\gamma^m(\mathbf{a})c_{\kappa+\gamma}\Big)B_\kappa^{d-m}(\mathbf{b}). \tag{3.46}$$

The theorem is proved by using Eq. 3.20 from **Theorem** 3 and Eq. 2.66 to replace the sums in Eq. 3.46 with their equivalent matrix forms:

$$\begin{aligned}\sum_{|\kappa|=d-m}\Big(\sum_{|\gamma|=m} P_\gamma^m(\mathbf{a})c_{\kappa+\gamma}\Big)B_\kappa^{d-m}(\mathbf{b}) &= \Big(\big[B_\kappa^{d-m}(\mathbf{b})\big]_{|\kappa|=d-m}\Big)\cdot \\ &\quad \left[\sum_{|\gamma|=m} P_\gamma^m(\mathbf{a})c_{\kappa+\gamma}\right]_{|\kappa+\gamma|=d} \\ &= \mathbf{B}^{d-m}(\mathbf{b})\mathbf{P}^{d,d-m}(\mathbf{a})\cdot\mathbf{c}^{t_j}. \end{aligned} \tag{3.47}$$

substitution of the final result of Eq. 3.47 in Eq. 3.46 then immediately results in Eq. 3.45, proving the theorem.

□

An example of the formulation of the directional derivatives in the matrix form from Eq. 3.45 will now be given.

Example 11 (Formulating directional derivatives). *In this example, the second order directional derivative of a bivariate spline function* $p(\mathbf{b})$ *of degree* $d = 3$ *in the direction* $\mathbf{u}$ *on the simplex* t_j *will be derived. In this case, we have* $m = 2$, $d - m = 1$ *and* $\mathbf{b} = (b_0, b_1)$*, with which Eq. 3.45 becomes:*

$$D^2_{\mathbf{u}} p(b_0, b_1) = \frac{3!}{1!} \mathbf{B}^1(b_0, b_1) \mathbf{P}^{3,1}(\mathbf{a}) \cdot \mathbf{c}^{t_j}, \tag{3.48}$$

with $\mathbf{a}$ *the directional coordinate of* $\mathbf{u}$.

In Eq. 3.48 we recognize the de Casteljau matrix $\mathbf{P}^{3,1}(\mathbf{a})$ *from* ***Example*** *10:*

$$\mathbf{P}^{3,1}(\mathbf{a}) = \begin{bmatrix} P^2_{2,0}(\mathbf{a}) & P^2_{1,1}(\mathbf{a}) & P^2_{0,2}(\mathbf{a}) & 0 \\ 0 & P^2_{2,0}(\mathbf{a}) & P^2_{1,1}(\mathbf{a}) & P^2_{0,2}(\mathbf{a}) \end{bmatrix}.$$

The vector of basis polynomials $\mathbf{B}^1(\mathbf{b})$ *contains only linear terms:*

$$\mathbf{B}^1(b_0, b_1) = \begin{bmatrix} b_0 & b_1 \end{bmatrix}.$$

Putting everything together, we find that the directional derivative is a linear function in b_0 *and* b_1*:*

$$D^2_{\mathbf{u}} p(b_0, b_1) = 6 \begin{bmatrix} b_0 & b_1 \end{bmatrix} \cdot \begin{bmatrix} P^2_{2,0}(\mathbf{a}) & P^2_{1,1}(\mathbf{a}) & P^2_{0,2}(\mathbf{a}) & 0 \\ 0 & P^2_{2,0}(\mathbf{a}) & P^2_{1,1}(\mathbf{a}) & P^2_{0,2}(\mathbf{a}) \end{bmatrix} \cdot \mathbf{c}^{t_j}.$$

■

Some further useful operators which can be expressed in terms of the directional derivatives are the gradient, Laplacian, and divergence operators. These operators can be simplified using the matrix form of the directional derivatives from Eq. 3.45.

The gradient of a spline function is given by:

$$\begin{aligned} \nabla p(b(\mathbf{x}))_{t_j} &= \left(D^1_{x_1} p(b(\mathbf{x}))_{t_j}, D^1_{x_2} p(b(\mathbf{x}))_{t_j}, \ldots, D^1_{x_n} p(b(\mathbf{x}))_{t_j} \right) \\ &= d\mathbf{B}^{d-1}(\mathbf{b}) \cdot \Big(\mathbf{P}^{d,d-1}(a(\mathbf{x}_1)) \cdot \mathbf{c}^{t_j}, \mathbf{P}^{d,d-1}(a(\mathbf{x}_2)) \cdot \mathbf{c}^{t_j}, \ldots, \\ &\qquad \mathbf{P}^{d,d-1}(a(\mathbf{x}_n)) \cdot \mathbf{c}^{t_j} \Big), \end{aligned} \tag{3.49}$$

with $a(\mathbf{x}_i)$ the directional coordinate of the (normalized) vector $\mathbf{x}_i$ in the direction of the i-axis.

The Laplacian is given by:

$$\begin{aligned}\Delta p(b(\mathbf{x}))_{t_j} &= \left(D^2_{x_1}p(b(\mathbf{x}))_{t_j}, D^2_{x_2}p(b(\mathbf{x}))_{t_j}, \ldots, D^2_{x_n}p(b(\mathbf{x}))_{t_j}\right) \\ &= \frac{d!}{(d-2)!}\mathbf{B}^{d-2}(\mathbf{b}) \cdot \Big(\mathbf{P}^{d,d-2}(a(\mathbf{x}_1)) \cdot \mathbf{c}^{t_j}, \mathbf{P}^{d,d-2}(a(\mathbf{x}_2)) \cdot \mathbf{c}^{t_j}, \ldots, \\ &\qquad \mathbf{P}^{d,d-2}(a(\mathbf{x}_n)) \cdot \mathbf{c}^{t_j}\Big).\end{aligned} \tag{3.50}$$

Finally, the divergence of a spline function is given by:

$$\begin{aligned}\operatorname{div} p(b(\mathbf{x}))_{t_j} &= \left(D^1_{x_1}p(b(\mathbf{x}))_{t_j} + D^1_{x_2}p(b(\mathbf{x}))_{t_j} + \cdots + D^1_{x_n}p(b(\mathbf{x}))_{t_j}\right) \\ &= d\mathbf{B}^{d-1}(\mathbf{b}) \cdot \Big(\mathbf{P}^{d,d-1}(a(\mathbf{x}_1)) \cdot \mathbf{c}^{t_j} + \mathbf{P}^{d,d-1}(a(\mathbf{x}_2)) \cdot \mathbf{c}^{t_j} + \cdots + \\ &\qquad \mathbf{P}^{d,d-1}(a(\mathbf{x}_n)) \cdot \mathbf{c}^{t_j}\Big).\end{aligned} \tag{3.51}$$

The expressions Eq. 3.49, Eq. 3.50 and Eq. 3.51 are given in the context of per-simplex calculations. They can easily be extended, however, by using the global B-form matrices from Eq. 3.33 and global de Casteljau matrix from Eq. 3.25.

In Figure 3.3 the divergence operator of a 5^{th} degree bivariate spline function is demonstrated. The spline function in this case is discontinuous, that is, there is no continuity between the individual spline pieces. When looking at the spline function in the left hand plot of Figure 3.3, this discontinuous nature may not be immediately apparent. In this case, the approximation power of the spline function is sufficient to accurately model the Mexican hat function, seemingly without discontinuities. The divergence operator tells a different story, however. In the right hand plot of Figure 3.3 the divergence operator from Figure 3.51 of the spline function is shown. The plot clearly shows significant discontinuities in the divergence vector field along the edges of simplices. As it turns out, the divergence operator provides a powerful method of visually inspecting the continuity of a simplex spline function.

3.2.7 Bounds on approximation with B-form polynomials

In [104] Lai and Schumaker derived a bound for how well a smooth function f can be approximated by a polynomial p_f of degree d. In this thesis, this expression is generalized to the n-dimensional case.

Lai and Schumaker showed in [104] that the smoothness of a function f on the domain Ω can be expressed as the following seminorm:

$$|f|_{m+1,\Omega} := \max_{\sum \alpha_i = m+1} \|D^{\alpha_0}_{x_0} D^{\alpha_1}_{x_1} \cdots D^{\alpha_n}_{x_n} f\|_{\Omega}, \tag{3.52}$$

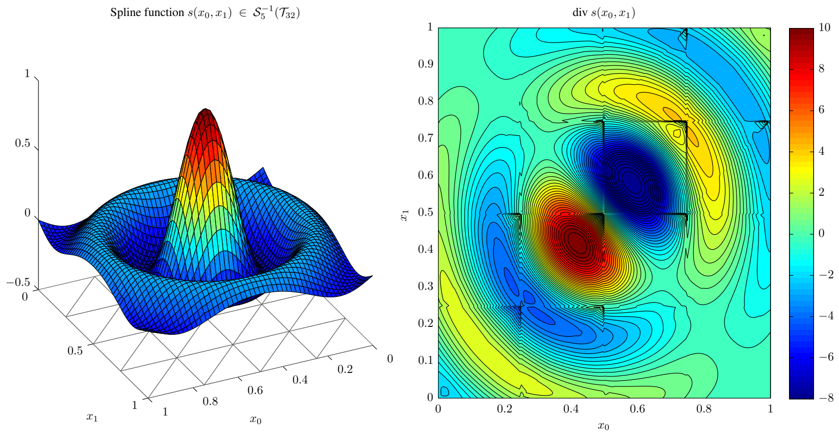


Figure 3.3: Discontinuous fifth degree spline function approximating the Mexican hat function (left) and a plot of its divergence operator (right). Notice that the discontinuities in the plot of the divergence operator coincide with simplex edges.

with $\| \bullet \|$ the notation for the ∞-norm as used in [104], and with $\mathbf{x}_i$ directional vectors.

Additionally, the diameter of the domain Ω was defined in [104] as follows:

$$|\Omega| := \max_{\mathbf{v},\mathbf{w} \in \Omega} |\mathbf{v} - \mathbf{w}|. \tag{3.53}$$

Using Eq. 3.52 and Eq. 3.53, the generalized bound on the accuracy of fit of a polynomial $p_f \in \mathcal{P}_d$ to the function $f \in C^{d+1}(\Omega)$ then is:

$$\|D_{x_0}^{\alpha_0} D_{x_1}^{\alpha_1} \cdots D_{x_n}^{\alpha_n}(f - p_f)\|_\Omega \leq K \cdot |\Omega|^{d+1-\sum \alpha_i} \cdot |f|_{d+1,\Omega}, \;\; 0 \leq \sum \alpha_i \leq d, \tag{3.54}$$

with K a constant dependent only on the polynomial order d and with $\mathcal{P}_d$ the space of all polynomials of degree d.

Because f is assumed to have a minimum continuity order C^{d+1} on Ω, the definition of the ∞-norm from [104] can be used to simplify this equation as follows:

$$\max |D_{x_0}^{\alpha_0} D_{x_1}^{\alpha_1} \cdots D_{x_n}^{\alpha_n}(f - p_f)|_\Omega \leq K \cdot |\Omega|^{d+1-\sum \alpha_i} \cdot |f|_{d+1,\Omega}, \;\; 0 \leq \sum \alpha_i \leq d, \tag{3.55}$$

Special cases of Eq. 3.55 are found for $\sum \alpha_i = 0$, the maximum absolute error between

f and p_f on Ω, and $\sum \alpha_i = d$ the maximum absolute error between the d^{th} order derivative of f and the maximum derivative of p_f. For $\sum \alpha_i = 0$ Eq. 3.55 can be written as follows:

$$\max |f - p_f|_\Omega \leq K \cdot |\Omega|^{d+1} \cdot |f|_{d+1,\Omega}, \tag{3.56}$$

while for $\sum \alpha_i = d$ Eq. 3.55 becomes:

$$\max |D_{x_0}^{\alpha_0} D_{x_1}^{\alpha_1} \cdots D_{x_n}^{\alpha_n} (f - P_f)|_\Omega \leq K \cdot |\Omega| \cdot |f|_{d+1,\Omega}, \tag{3.57}$$

with $P_f = D_{x_0}^{\alpha_0} D_{x_1}^{\alpha_1} \cdots D_{x_n}^{\alpha_n} p_f$ the maximum derivative of p_f, which is a constant.

An important implication of Eq. 3.56 is the case when both p_f and f are a polynomials of degree d. In that case $|f|_{m+1,\Omega} = 0$ according to Eq. 3.52, and therefore $\max |f - p_f|_\Omega = 0$. In other words, if f is a polynomial of degree d which is being approximated by a B-form polynomial p_f of degree d, then the approximation is exact.

In [104] Lai and Schumaker presented an important proof on the bounds of a B-form polynomial $p \in \mathcal{P}_d$ given a set of B-coefficients. These bounds were found to be dependent only on the B-coefficients and degree of a B-form polynomial. The expression found by Lai and Schumaker is the following:

$$\frac{\|\mathbf{c}\|}{K} \leq \|p\| \leq \|\mathbf{c}\|, \tag{3.58}$$

with $\|p\|$ the maximum norm of the polynomial p and with $\|\mathbf{c}\|$ the maximum absolute value in the vector of per-simplex B-coefficients as follows:

$$\|\mathbf{c}\| := \max_{|\kappa|=d} |c_\kappa|, \tag{3.59}$$

The constant K in Eq. 3.58 is dependent on the B-coefficients and a constant matrix $\mathbf{M}$ as follows:

$$K := \max \left\{ \frac{\|\mathbf{M}^{-1}\mathbf{c}\|}{\|\mathbf{c}\|} \right\}, \quad \|\mathbf{c}\| \neq 0. \tag{3.60}$$

The matrix $\mathbf{M}$ in Eq. 3.60 is defined as follows:

$$\mathbf{M} := \left[B_\kappa(b(c_\gamma))\right]_{|\kappa|=d, |\gamma|=d} \in \mathbb{R}^{\hat{d} \times \hat{d}} \tag{3.61}$$

in which $B_\kappa(b(c_\gamma))$ is the κ^{th} Bernstein basis polynomial from Eq. 2.52 evaluated at the γ^{th} B-coefficient location from Eq. 2.69 where κ and γ are independent, lexicographically sorted, multi-indices. In this case $\mathbf{M}$ is a square matrix of size $\hat{d} \times \hat{d}$.

The bounds on the error between the smooth function f and the polynomial p_f given in Eq. 3.55 will prove to be instrumental in the quantization of bounds for model extrapolation which will be discussed in Sec. 3.4.3.

3.3 Smoothness Constraints

A spline function is, per definition, a piecewise defined polynomial function with C^r continuity between its pieces. Continuity between the polynomial pieces of the simplex spline is achieved by imposing *smoothness constraints* on the B-coefficients of neighboring simplices in the estimator for the B-coefficients. These smoothness constraints are formulated by relating the B-coefficients on both sides of an edge facet with special equations called *continuity conditions*.

In this section the concept of the continuity conditions is presented, along with a number of related topics. While the continuity conditions as presented in Sec. 3.3.1 are well known in the literature, see e.g. [31, 4, 7, 96, 104], their formulation is not correct in general. In Sec. 3.3.2 a general formulation for the continuity conditions is provided which is valid for all possible triangulations, as long as B-nets are orientated using the B-net orientation rule from Sec. 2.3.5. A qualitative analysis of the effects of the continuity conditions on the global spline function will then be provided in Sec. 3.3.3.

Some aspects related to the continuity conditions have been little explored in the literature. One of these aspects is the propagation of disturbances through the B-net of a triangulation as the result of a localized excitation. When left unrestrained, propagation effectively transforms a simplex spline from a local approximator into a semi-global approximator. In Sec. 3.3.4 it is shown that the order of continuity with respect to the spline degree together with the triangulation configuration are the determining factors in the magnitude of disturbance propagation through the B-net.

3.3.1 Definition of the continuity conditions

Smoothness constraints are formed by relating the B-nets of two n-simplices that share an edge facet using special equations called continuity conditions. As already explained in Sec. 2.2.1, an edge facet is a unique simplex of dimension $n-1$ which is formed by the n vertices that define the edge between two simplices. Any set of two simplices share one, and only one, unique edge facet. The formal definition of an edge facet is the following. Let two neighboring n-simplices t_i and t_j be defined as follows:

$$t_i = \langle \mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}, \tilde{\mathbf{v}}_{i,j} \rangle\,,\ \ t_j = \langle \mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}, \tilde{\mathbf{v}}_{j,i} \rangle\,. \tag{3.62}$$

Then t_i and t_j meet along the edge facet $\tilde{t}_{i,j}$ given by:

$$\tilde{t}_{i,j} = t_i \cap t_j = \langle \mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1} \rangle\,. \tag{3.63}$$

Using Eq. 2.26 it is easy to see that $\tilde{t}_{i,j}$ is an $(n-1)$-simplex, because it consists of n vertices. Each of the simplices t_i and t_j has a single vertex which is not part of the edge facet. Given the edge facet $\tilde{t}_{i,j}$ these vertices are $\tilde{\mathbf{v}}_{i,j}$ for t_i and $\tilde{\mathbf{v}}_{j,i}$ for t_j. These vertices

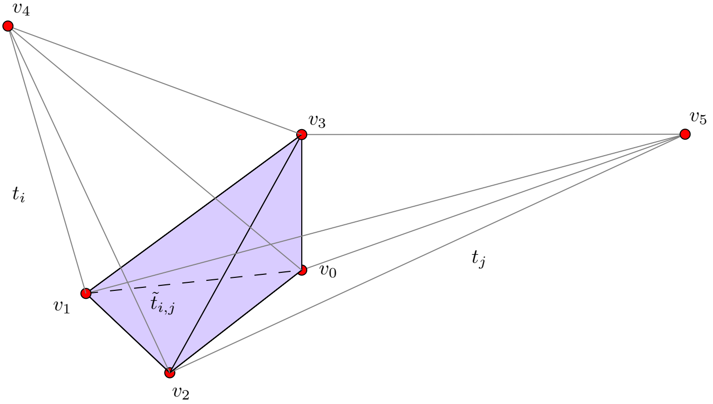


Figure 3.4: The edge facet $\tilde{t}_{i,j}$ of the two 4-simplices t_i and t_j is a tetrahedron.

are the so-called *out-of-edge vertices*, which have the following formal definition:

$$\tilde{\mathbf{v}}_{i,j}, \tilde{\mathbf{v}}_{j,i} \notin \tilde{t}_{i,j}. \tag{3.64}$$

As de Boor observed in [31] the edge facet $\tilde{t}_{i,j}$ is indirectly defined by either one of the out of edge vertices $\tilde{\mathbf{v}}_{i,j}$ and $\tilde{\mathbf{v}}_{j,i}$ as follows:

$$\tilde{t}_{i,j} = \begin{cases} t_i \setminus \tilde{\mathbf{v}}_{i,j}, \\ t_j \setminus \tilde{\mathbf{v}}_{j,i} \end{cases} \tag{3.65}$$

De Boor's observation will prove to be important, because it simplifies the implementation of the algorithm for formulating the generalized continuity conditions as will be shown at a later point.

In Figure 3.4 two neighboring 4-simplices are shown, together with their edge facet, which is the tetrahedron $\tilde{t}_{i,j}$ defined by the vertices $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$. Interestingly, if the 4^{th} spatial component of the vertices of $\tilde{t}_{i,j}$ are all equal to zero, then $\tilde{t}_{i,j}$ is a structure that is completely embedded in $\mathbb{R}^3$. In this case, the two 4-simplices are said to have a mutual tetrahedronal base in $\mathbb{R}^3$. From this perspective it can be reasoned that any tetrahedron in $\mathbb{R}^3$ is in fact an edge facet of two special 4-simplices with tetrahedronal bases in $\mathbb{R}^3$. Furthermore, any 3-D shape can in theory be triangulated with arbitrary precision with a set of $J < \infty$ tetrahedrons. Therefore, any 3-dimensional object in the real world can be considered an edge facet, or interface object, between two 4-dimensional objects!

Continuity of order C^r between B-form polynomials on two neighboring simplices t_i and t_j is achieved when their directional derivatives up to order C^r are equal at every point

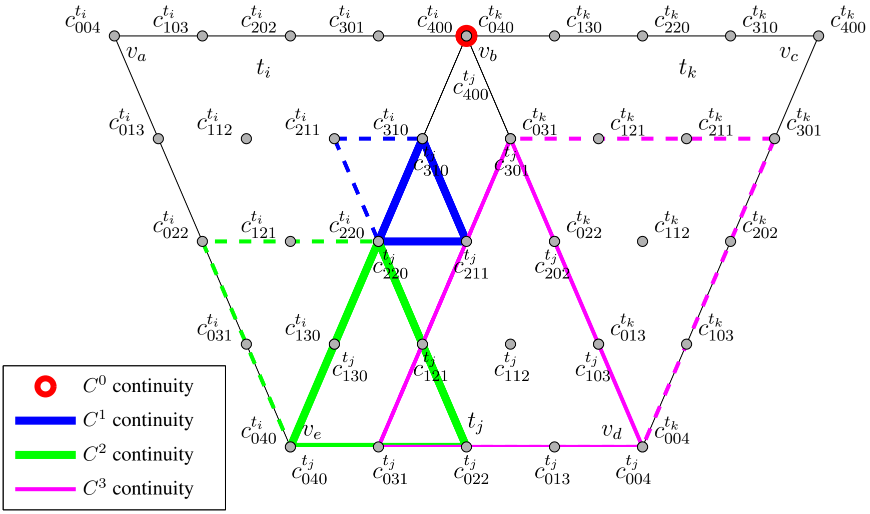


Figure 3.5: An example of the structure of continuity of 4 individual continuity conditions for C^0, C^1, C^2 and C^3 continuity on a fourth order B-net on the three simplices t_i, t_j and t_k.

on the edge facet $\tilde{t}_{i,j}$. Therefore, if p_{t_i} is a B-form polynomial on t_i and p_{t_j} is a B-form polynomial on t_j, then the following expression must hold for all points $\mathbf{x}$ in $\tilde{t}_{i,j}$:

$$D^m_{\mathbf{u}} p_{t_i}(b(\mathbf{x})) = D^m_{\mathbf{u}} p_{t_j}(b(\mathbf{x})), \ \forall \, \mathbf{x} \in \tilde{t}_{i,j}, \ \ m = 0, 1, \ldots, r \tag{3.66}$$

It has been proved in the literature that Eq. 3.66 holds when the B-nets of t_i and t_j are related through continuity conditions, see e.g. [4]. The most prevalent formulation of the continuity conditions is that offered by Awanou [4] and Lai [104] as follows:

$$c^{t_i}_{\kappa_0,\ldots,\kappa_{n-1},m} = \sum_{|\gamma|=m} c^{t_j}_{(\kappa_0,\ldots,\kappa_{n-1},0)+\gamma} B^m_\gamma(\tilde{\mathbf{v}}_{i,j}), \ \ 0 \leq m \leq r, \tag{3.67}$$

with $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_n) \in \mathbb{R}^{n+1}$ a multi-index independent of κ. Note that the basis function $B^m_\gamma(\tilde{\mathbf{v}}_{i,j})$ is a polynomial in terms of the barycentric coordinate of $\tilde{\mathbf{v}}_{i,j}$ with respect to simplex t_j.

The total number of continuity conditions required for C^r continuity depends on the dimension, degree, and continuity order of a spline function. In the following theorem, an expression for this number will be presented.

Theorem 5. *For C^r continuity of an n-variate spline function $s \in \mathcal{S}_d^r(\mathcal{T})$ there are a total of R_r continuity conditions per edge with R_r defined as follows:*

$$R_r := \sum_{m=0}^{r} \frac{(d-m+n-1)!}{(n-1)!(d-m)!} \tag{3.68}$$

Proof. The proof is based on the dimensionality of the edge facet $\tilde{t}_{i,j}$, which according to Eq. 3.63 is always equal to $n-1$. Furthermore, it should be observed that B-coefficients are located on a total of $d+1$ parallel planes of dimension $n-1$, see e.g. Figure 2.18 and Figure 2.19. The base plane (i.e. plane-0) in this case is $\tilde{t}_{i,j}$ itself, which using Eq. 2.51 contains R_0 B-coefficients:

$$R_0 = \frac{(d+n-1)!}{(n-1)!d!}. \tag{3.69}$$

In general, the m^{th} plane (i.e. plane-m) from $\tilde{t}_{i,j}$ contains R_m B-coefficients:

$$R_m = \frac{(d-m+n-1)!}{(n-1)!(d-m)!}. \tag{3.70}$$

Because every B-coefficient in the 0^{th} till m^{th} plane is subject to a single continuity condition, the proof of the theorem follows from summing Eq. 3.70 for every continuity order 0 till m which results in Eq. 3.68.

□

While it may not be immediately apparent from Eq. 3.67, there exists a strong structure in the continuity conditions. This structure, here called the *structure of continuity* is well known in the literature, see e.g. [52] and [96]. The principle of the structure of continuity is demonstrated in Figure 3.5, where parts of the continuity structure for C^0, C^1, C^2 and C^3 continuity are shown. For example, the continuity condition for $c^{t_i}_{2,1,1}$ involves three B-coefficients in t_j: $c^{t_i}_{3,1,0}$, $c^{t_i}_{2,2,0}$ and $c^{t_i}_{2,1,1}$. In this thesis, these coefficients are called the *continuity body* B-coefficients as they form the polynomial body of the continuity condition. The single B-coefficient for which the continuity conditions is formulated, in this case $c^{t_i}_{2,1,1}$, is called the *continuity point* B-coefficient. The continuity body and continuity point B-coefficients together form a diamond like structure which chord lies in the edge facet. This diamond structure actually is a miniature version of the two neighboring simplices. In higher dimensions this pattern continues: the structure of continuity of an n-simplex consists of two smaller n-simplices joined together at their edge facet. A special case is C^0 continuity; in this case the structure of continuity lies entirely in the edge facet.

Example 12 (Formulating continuity conditions)**.** *Using Eq. 3.67, the three continuity conditions associated with the edge between t_i and t_j in Figure 3.5 will be derived.*

For C^0 continuity at $c^{t_i}_{4,0,0}$ we have:

$$\gamma = (0,0,0).$$

The C^0 continuity condition then is:

$$\begin{aligned} c^{t_i}_{4,0,0} &= \sum_{|\gamma|=0} c^{t_j}_{(4,0,0)+\gamma} B^0_\gamma(\mathbf{v}_a), \\ &= c^{t_j}_{4,0,0}. \end{aligned}$$

For C^1 continuity at $c^{t_i}_{2,1,1}$ we have:

$$\gamma \in \{(1,0,0),(0,1,0),(0,0,1)\}.$$

The C^1 continuity condition then is:

$$\begin{aligned} c^{t_i}_{2,1,1} &= \sum_{|\gamma|=1} c^{t_j}_{(2,1,0)+\gamma} B^1_\gamma(\mathbf{v}_a), \\ &= c^{t_j}_{3,1,0} B^1_{1,0,0}(\mathbf{v}_a) + c^{t_j}_{2,2,0} B^1_{0,1,0}(\mathbf{v}_a) + c^{t_j}_{2,1,1} B^1_{0,0,1}(\mathbf{v}_a). \end{aligned}$$

For C^2 continuity at $c^{t_i}_{0,2,2}$ we have:

$$\gamma \in \{(2,0,0),(1,1,0),(1,0,1),(0,2,0),(0,1,1),(0,0,2)\}.$$

The C^2 continuity condition then is:

$$\begin{aligned} c^{t_i}_{0,2,2} &= \sum_{|\gamma|=2} c^{t_j}_{(0,2,0)+\gamma} B^2_\gamma(\mathbf{v}_a), \\ &= c^{t_j}_{2,2,0} B^2_{2,0,0}(\mathbf{v}_a) + c^{t_j}_{1,3,0} B^2_{1,1,0}(\mathbf{v}_a) + c^{t_j}_{1,2,1} B^2_{1,0,1}(\mathbf{v}_a) + c^{t_j}_{0,4,0} B^2_{0,2,0}(\mathbf{v}_a) + \\ &\quad + c^{t_j}_{0,3,1} B^2_{0,1,1}(\mathbf{v}_a) + c^{t_j}_{0,2,2} B^2_{0,0,2}(\mathbf{v}_a). \end{aligned}$$

■

3.3.2 A general formulation for the continuity conditions

There is a fundamental problem with the formulation of the continuity conditions in Eq. 3.67. It turns out that the formulation provided in the literature is valid only for a single very specific orientation of the two B-nets of the neighbor simplices. For all other $((n+1)!-1)$ B-net orientations, Eq. 3.67 produces invalid results. This failure is due to the fixed position of the 0 and m constants within the multi-indices of the B-coefficients in

Eq. 3.67. In the following example, the failure of Eq. 3.67 to describe the correct continuity structure will be demonstrated.

Example 13 (A problem with continuity)**.** *The expression for the continuity conditions from Eq. 3.67 is incomplete, as it only describes the correct continuity structure for a single B-net orientation, which is the orientation between t_i and t_j in Figure 3.5. For the continuity conditions between t_j and t_k the formulation from Figure 3.5 produces invalid results, as will be shown here.*

For C^0 continuity at $c^{t_k}_{0,4,0}$ we have:

$$\gamma = (0,0,0).$$

According to Eq. 3.67 the C^0 continuity condition then is:

$$\begin{aligned} c^{t_k}_{0,4,0} &= \sum_{|\gamma|=0} c^{t_j}_{(0,4,0)+\gamma} B^0_\gamma(\mathbf{v}_c), \\ &= c^{t_j}_{0,4,0}. \end{aligned}$$

which clearly is an invalid result as $c^{t_j}_{0,4,0}$ is not in the edge facet $\tilde{t}_{j,k}$.

For C^3 continuity at $c^{t_k}_{3,0,1}$ we have:

$$\gamma \in \{(3,0,0),(2,1,0),(2,0,1),(1,2,0),(1,1,1),(1,0,2),(0,3,0),(0,2,1),(0,1,2),(0,0,3)\}.$$

The C^3 continuity condition is:

$$c^{t_k}_{3,0,1} = \sum_{|\gamma|=3} c^{t_j}_{(3,0,0)+\gamma} B^3_\gamma(\mathbf{v}_c),$$

which can never produce correct results; the first right hand term would be $c^{t_j}_{6,0,0} B^2_{3,0,0}(\mathbf{v}_c)$, which contains a B-coefficient index that is not even in the B-net.

■

At this point it should be clear that Eq. 3.67 is only valid for a single orientation of B-nets, which is the orientation of the B-nets of t_i and t_j in Figure 3.5. This orientation is called the *maximum degree symmetric orientation*. The maximum degree symmetric orientation requires that the B-coefficient with the lowest lexicographical sorting order, i.e. $c_{0,0,\ldots,d}$, should be located at the out-of-edge vertex. It is easy to see that it is impossible to create a valid global B-net orientation scheme that at every set of two neighboring simplices has the same orientation as t_i and t_j in Figure 3.5. In the following we shall therefore assume that the B-net is oriented using the B-net orientation rule from Eq. 2.78.

At this point one can wonder as to why a global orientation scheme is used at all. Why not just simply reorganize, on-the-fly, the B-net for every set of two simplices such that

Eq. 3.67 becomes valid? The problem with this approach is that the B-coefficients have a unique fixed location in the global vector of B-coefficients from Eq. 3.27. This location is determined by the lexicographical sorting rule for B-coefficients together with the simplex index, and is essential for the correct functioning of the estimator for the B-coefficients. If the B-net is reorganized, the continuity conditions will refer to the wrong B-coefficients in the global vector of B-coefficients. The only way that on-the-fly reorganization can work is when the B-coefficients subject to the continuity conditions are translated back into the B-coefficients in the global B-coefficient vector. For example, we could flip the B-net of t_j and rotate the B-net of t_k in Figure 3.5 counterclockwise such that we get the orientation for which Eq. 3.67 is valid. In that case we get the following set of C^0 continuity conditions for the transformed B-coefficients:

$$\bar{c}^{t_j}_{4,0,0} = \bar{c}^{t_k}_{4,0,0},\ \bar{c}^{t_j}_{3,1,0} = \bar{c}^{t_k}_{3,1,0},\ \bar{c}^{t_j}_{2,2,0} = \bar{c}^{t_k}_{2,2,0},\ \bar{c}^{t_j}_{1,3,0} = \bar{c}^{t_k}_{1,3,0},\ \bar{c}^{t_j}_{0,4,0} = \bar{c}^{t_k}_{0,4,0},$$

where the overline denotes that these are on-the-fly transformed B-coefficients.

Now these equations should be translated back into the original B-net orientation using the inverse transformation rules:

$$\begin{aligned}
&\bar{c}^{t_j}_{4,0,0} \rightarrow c^{t_j}_{4,0,0},\ \bar{c}^{t_j}_{3,1,0} \rightarrow c^{t_k}_{3,0,1},\ \bar{c}^{t_j}_{2,2,0} \rightarrow c^{t_j}_{2,0,2},\ \bar{c}^{t_j}_{1,3,0} \rightarrow c^{t_k}_{1,0,3},\ \bar{c}^{t_j}_{0,4,0} \rightarrow c^{t_j}_{0,0,4},\\
&\bar{c}^{t_k}_{4,0,0} \rightarrow c^{t_k}_{4,0,0},\ \bar{c}^{t_k}_{3,1,0} \rightarrow c^{t_k}_{0,3,1},\ \bar{c}^{t_k}_{2,2,0} \rightarrow c^{t_k}_{0,2,2},\ \bar{c}^{t_k}_{1,3,0} \rightarrow c^{t_k}_{0,1,3},\ \bar{c}^{t_k}_{0,4,0} \rightarrow c^{t_k}_{0,0,4},
\end{aligned}$$

where the overline again denotes the on-the-fly transformed B-coefficients.

What just happened here is that the problem has been transformed from the formulation of the continuity conditions into the determination of the correct rotation and flip sequences to get to the maximum degree symmetric B-net orientation. While this is all technically possible, it requires much more complex mathematics and algorithmic implementations, especially for higher dimensional B-nets where it may be a very challenging task to just find the correct sequence of rotations and flips to get to the required B-net orientation. This is the reason why on-the-fly B-net reorganization is abandoned, and with it the formulation of the continuity conditions from Eq. 3.67.

It should be clear that a new, general, formulation of the continuity conditions is required. This new formulation is based on insights gained from the graphical interpretation of the continuity conditions, such as those shown in Figure 3.5 and Figure 3.6. First of all, it can be observed from these figures that for a globally indexed B-net, the location of the constant in the multi-index (i.e. the m and 0) is equal to the location of the single non-zero value in the multi-index of the B-coefficient located at the out-of-edge vertex. For example, the C^3 continuity condition for the B-coefficient $c^{t_k}_{3,0,1}$ in Figure 3.6 requires that the value $m = 3$ should be inserted at location 1 in the multi-index because this is also the location in the multi-index of the non-zero value of the B-coefficient $c^{t_k}_{4,0,0}$ at the out of edge vertex $\mathbf{v}_c$. At the same time, the corresponding continuity point B-coefficients are $c^{t_j}_{(0,0,1)+\gamma}$, which have a 0 inserted at location 2 in the multi-index because this is the location of the non-zero value of of the B-coefficient $c^{t_k}_{4,0,0}$ at the out of edge vertex $\mathbf{v}_e$.

There are only a finite number of possible values of this new multi-index created through the insertion of m or 0. It is therefore possible to define a function that produces the complete set of all multi-indices. Let the tuple function $\mathcal{M}(w, \kappa)$ be defined as the function that produces all new multi-indices for a given value w as follows:

$$\mathcal{M}(w,\kappa) := \begin{pmatrix} (w, \kappa_0, \kappa_1, \ldots, \kappa_{n-1}), \\ (\kappa_0, w, \kappa_1, \ldots, \kappa_{n-1}), \\ (\kappa_0, \kappa_1, w, \ldots, \kappa_{n-1}), \\ (\kappa_0, \kappa_1, \ldots, w, \kappa_{n-1}), \\ (\kappa_0, \kappa_1, \ldots, \kappa_{n-1}, w) \end{pmatrix}, \quad |\mathcal{M}_i(w,\kappa)| = d - m + w, \tag{3.71}$$

where the i^{th} multi-index is given by $\mathcal{M}_i(w, \kappa)$. Letting $w = m$, $\mathcal{M}(w, \kappa)$ produces all possible multi-indices of the continuity point (i.e. the left hand) B-coefficients, if $w = 0$ then $\mathcal{M}(0, \kappa) + \gamma$ produces all possible multi-indices of the continuity body (i.e. the right hand) B-coefficients.

Because κ in Eq. 3.71 has a total of $n - 1$ components and because $|\kappa| = d - w = d^*$, the total number of valid permutations of a single tuple $\mathcal{M}_i(w, \kappa)$ follows from Eq. 2.51 as follows:

$$|\{\mathcal{M}_i(w,\kappa)\}| = \frac{(d^* + n - 1)!}{(n-1)!(d^*)!} = R_w, \tag{3.72}$$

with R_w the total number of continuity conditions of order w from Eq. 3.70. Note that in this case the operator $|\bullet|$ is the set cardinality operator.

So what is the role of $\mathcal{M}(w, \kappa)$ in the context of the continuity conditions? The answer to this question is rather simple. The problem with the original continuity condition formulation from Eq. 3.67 was that its B-coefficient indexing scheme was only functional on a single, very specific B-net configuration. The tuple function $\mathcal{M}(w, \kappa)$ allows for a much more powerful indexing scheme that can be used on any combination of two B-nets as long as they are both oriented using the B-net orientation rule from Eq. 2.78. In this sense, $\mathcal{M}(w, \kappa)$ can be used as a multi-index generator as follows:

$$\bar{\kappa} = \mathcal{M}_{\rho(\mathbf{v})}(w,\kappa), \quad |\bar{\kappa}| = d - m + w, \tag{3.73}$$

with $\bar{\kappa}$ a new multi-index and with $\rho(\mathbf{v})$ a rank function that produces the index into $\mathcal{M}(m, \kappa)$. It is important to note that the actual values of the components of $\bar{\kappa}$ are determined by the requirement that $|\bar{\kappa}| = d - m + w$.

The rank function $\rho(\mathbf{v})$ returns the rank of a given vertex within the set of simplex

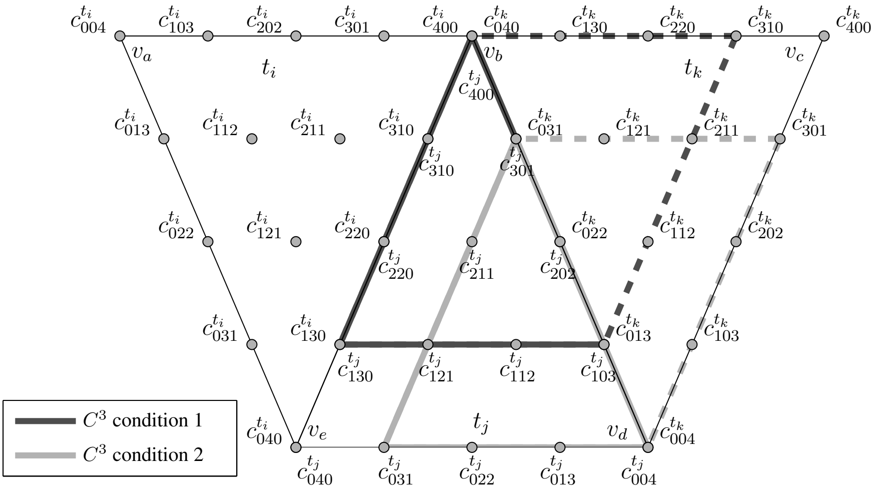


Figure 3.6: The two C^3 continuity conditions for the edge t_j and t_k.

vertices $\mathcal{V}_t$ based on the global vertex index. The rank function is defined as follows:

$$\rho(\mathbf{v}_{p_i}) := (n+2) - \sum_{j=0}^{p_i} k_{v_j}, \quad \begin{cases} k_{v_j} = 1 \text{ if } \mathbf{v}_j \in \mathcal{V}_t \\ k_{v_j} = 0 \text{ if } \mathbf{v}_j \notin \mathcal{V}_t \end{cases} \tag{3.74}$$

Because the output from the tuple function Eq. 3.71 is a tuple, pairs of tuples of continuity point-body indices are produced as follows:

$$(\mathcal{M}_i(m,\kappa), \mathcal{M}_j(0,\kappa)) = (\kappa_{\text{point}}, \kappa_{\text{body}}) \tag{3.75}$$

These pairs form the basis of the indexing scheme for the formulation of the general continuity conditions.

In the following example, the use of the tuple function from Eq. 3.71 together with the rank function from Eq. 3.74 will be demonstrated.

Example 14 (Using the tuple function $\mathcal{M}(w,\kappa)$). *In this example we want to derive the multi-indices of the B-coefficients for C^3 continuity for the edge between t_j and t_k for the 4^{th} degree B-net in Figure 3.6. The B-net of t_k will produce the continuity body B-coefficients, while the B-net of t_j produces the continuity point B-coefficients. The vertex indices are explicitly given as follows: $a = 3$, $b = 6$, $c = 8$, $d = 2$ and $e = 5$.*

In this case we have $t_j = \langle \mathbf{v}_6, \mathbf{v}_5, \mathbf{v}_2 \rangle$ *and* $t_k = \langle \mathbf{v}_8, \mathbf{v}_6, \mathbf{v}_2 \rangle$*.*

The first step in the derivation is the determination of the vertex rank of the out of edge vertex $\mathbf{v}_c = \mathbf{v}_8$ *for simplex* t_k*. Using Eq. 3.74 we find for the vertex rank function:*

$$\begin{aligned} \rho(\mathbf{v}_8) &= (n+2) - \sum_{j=0}^{8} k_{\mathbf{v}_8} \\ &= (n+2) - (0+0+1+0+0+0+1+0+1). \end{aligned}$$

so the rank of $\mathbf{v}_8$ *is given by as* $\rho(\mathbf{v}_8) = 1$*. We can now use the tuple function* $\mathcal{M}(w,\kappa)$ *to construct all possible multi-indices for* $w = 3$ *(and* $n = 2$*):*

$$\mathcal{M}(3,\kappa) = \left(\begin{array}{c} (3,\kappa_0,\kappa_1), \\ (\kappa_0,3,\kappa_1), \\ (\kappa_0,\kappa_1,3) \end{array} \right), \ |\mathcal{M}_i(3,\kappa)| = 4 - 3 + 3 = 4, \ 1 \leq i \leq 3.$$

Using the outcome from the vertex rank function, we find all possible multi-indices for C^3 *continuity in* t_k*:*

$$\begin{aligned} \mathcal{M}_{\rho(\mathbf{v}_8)}(3,\kappa) &= \mathcal{M}_1(3,\kappa) \\ &= (3,\kappa_0,\kappa_1) \in \{(3,1,0),(3,0,1)\}. \end{aligned}$$

We repeat this process to find the multi-indices for the continuity point B-coefficients for t_j*. First, we find for the rank of* $\mathbf{v}_e = \mathbf{v}_5$*:*

$$\begin{aligned} \rho(\mathbf{v}_5) &= (n+2) - \sum_{j=0}^{5} k \\ &= (n+2) - (0+0+1+0+0+1) = 2. \end{aligned}$$

Using the tuple function $\mathcal{M}(w,\kappa)$ *to construct all possible multi-indices for* $w = 0$ *we get:*

$$\mathcal{M}(0,\kappa) = \left(\begin{array}{c} (0,\kappa_0,\kappa_1), \\ (\kappa_0,0,\kappa_1), \\ (\kappa_0,\kappa_1,0) \end{array} \right), \ |\mathcal{M}_i(0,\kappa)| = 4 - 3 + 0 = 1, \ 1 \leq i \leq 3.$$

We find all possible multi-indices for C^3 *continuity in* t_j *as follows:*

$$\begin{aligned} \mathcal{M}_{\rho(\mathbf{v}_5)}(0,\kappa) &= \mathcal{M}_2(0,\kappa) \\ &= (\kappa_0,0,\kappa_1) \in \{(1,0,0),(0,0,1)\}. \end{aligned}$$

Combining results and using Eq. 3.75 we get the following set of continuity pairs:

$$(\mathcal{M}_1(w,\kappa), \mathcal{M}_2(0,\kappa)) \in \left\{ \begin{array}{c} ((3,1,0),(1,0,0)), \\ ((3,0,1),(0,0,1)) \end{array} \right\}.$$

■

All theory is now in place to introduce a theorem for the continuity conditions of B-nets oriented using the B-net orientation rule from Eq. 2.78.

Theorem 6. *The general formulation for the continuity conditions for two n-simplices t_f and t_g with d^{th} degree B-nets which are oriented using the B-net orientation rule from Eq. 2.78 is the following:*

$$c^{t_f}_{\mathcal{M}_{\rho(\tilde{\mathbf{v}}_{f,g})}(m,\kappa)} = \sum_{|\gamma|=m} c^{t_g}_{\left(\mathcal{M}_{\rho(\tilde{\mathbf{v}}_{g,f})}(0,\kappa)+\gamma\right)} B^m_\gamma(\tilde{\mathbf{v}}_{f,g}), \;\; 0 \leq m \leq r. \tag{3.76}$$

with $\mathcal{M}_{\rho(\bullet)}(\bullet)$ the tuple function Eq. 3.71 which uses the vertex rank function from Eq. 3.74 to index its members. The basis function $B^m_\gamma(\tilde{\mathbf{v}}_{f,g})$ is a polynomial in terms of the barycentric coordinate of $\tilde{\mathbf{v}}_{f,g}$ with respect to simplex t_g.

Proof. The proof of the theorem requires that the correct continuity structure is produced for all possible combinations of ranks of $\tilde{\mathbf{v}}_{f,g}$ and $\tilde{\mathbf{v}}_{g,f}$, that is, all $(n+1)!$ possible orientations of B-nets.

Observe that Eq. 3.76 reduces to Eq. 3.67 when $\rho(\tilde{\mathbf{v}}_{f,g}) = n+1$ and $\rho(\tilde{\mathbf{v}}_{g,f}) = n+1$, because this indexes the last multi-index value of the tuple function from Eq. 3.71. In that case we get the following result:

$$c^{t_f}_{(\kappa_0,\kappa_1,\ldots,\kappa_{n-1},m)} = \sum_{|\gamma|=m} c^{t_g}_{(\kappa_0,\kappa_1,\ldots,\kappa_{n-1},0)+\gamma} B^m_\gamma(\tilde{\mathbf{v}}_{f,g}), \;\; 0 \leq m \leq r. \tag{3.77}$$

which has been proved to be correct in the literature, see e.g. [4]. Now reorient both B-nets such that $\rho(\tilde{\mathbf{v}}_{f,g}) = n$ and $\rho(\tilde{\mathbf{v}}_{g,f}) = n$, that is, the rank of the out of edge vertices is increased by 1. Note that this B-net orientation is again symmetric. In this case we get for Eq. 3.76:

$$c^{t_f}_{(\kappa_0,\kappa_1,\ldots,m,\kappa_{n-1})} = \sum_{|\gamma|=m} c^{t_g}_{(\kappa_0,\kappa_1,\ldots,0,\kappa_{n-1})+\gamma} B^m_\gamma(\tilde{\mathbf{v}}_{f,g}), \;\; 0 \leq m \leq r. \tag{3.78}$$

It is straightforward to prove that this equation is correct; by simply swapping m and κ_{n-1} in the continuity body B-coefficients and 0 and κ_{n-1} in the continuity point B-coefficients we get the exact same continuity structure as that produced by Eq. 3.77. Through induction (i.e. swapping m and 0 with κ_{n-2} and then m and 0 with κ_{n-1}) we can prove that the theorem is valid for all possible symmetric B-net orientations, leaving us to prove the theorem for all asymmetric B-net orientations, which is now straightforward.

If $\mathcal{M}_{\rho(\tilde{\mathbf{v}}_{f,g})}(m, \kappa)$ produces the correct multi-indices for all ranks of $\tilde{\mathbf{v}}_{f,g}$, and $\mathcal{M}_{\rho(\tilde{\mathbf{v}}_{g,f})}(0, \kappa) + \gamma$ produces the correct multi-indices for all ranks of $\tilde{\mathbf{v}}_{g,f}$ then logically, Eq. 3.76 produces the correct continuity structures for all possible combinations of ranks of $\tilde{\mathbf{v}}_{f,g}$ and $\tilde{\mathbf{v}}_{g,f}$, which proves the theorem.

□

Example 15 (Formulating general continuity conditions)**.** *In this example, the general formulation of the continuity conditions from Eq. 3.76 will be demonstrated. Using the results form* ***Example*** *14, the two continuity conditions for C^3 continuity between the simplices t_j and t_k in Figure 3.6 will be derived.*

In ***Example*** *14 we found that $\rho(\mathbf{v}_8) = 1$ and $\rho(\mathbf{v}_5) = 2$, which led to the following expression for the two continuity pairs:*

$$(\mathcal{M}_1(w, \kappa), \mathcal{M}_2(0, \kappa)) \in \left\{ \begin{array}{c} ((3,1,0),(1,0,0)), \\ ((3,0,1),(0,0,1)) \end{array} \right\}$$

with each of the two pairs associated with a single continuity condition.

We now first use the continuity pair $((3,1,0),(1,0,0))$ to find the continuity condition in which $c^{t_k}_{\mathcal{M}_{\rho(\mathbf{v}_8)}(3,\kappa)} = c^{t_k}_{3,1,0}$ is the continuity point B-coefficient:

$$\begin{aligned}
c^{t_k}_{3,1,0} &= \sum_{|\gamma|=3} c^{t_j}_{\mathcal{M}_{\rho(\mathbf{v}_5)}(0,\kappa)+\gamma} B^3_\gamma(\mathbf{v}_8) \\
&= \sum_{|\gamma|=3} c^{t_j}_{(1,0,0)+\gamma} B^3_\gamma(\mathbf{v}_8) \\
&= c^{t_j}_{4,0,0} B^3_{3,0,0}(\mathbf{v}_8) + c^{t_j}_{3,1,0} B^3_{2,1,0}(\mathbf{v}_8) + c^{t_j}_{3,0,1} B^3_{2,0,1}(\mathbf{v}_8) + c^{t_j}_{2,2,0} B^3_{1,2,0}(\mathbf{v}_8) + \\
&\quad + c^{t_j}_{2,1,1} B^3_{1,1,1}(\mathbf{v}_8) + c^{t_j}_{2,0,2} B^3_{1,0,2}(\mathbf{v}_8) + c^{t_j}_{1,3,0} B^3_{0,3,0}(\mathbf{v}_8) + c^{t_j}_{1,2,1} B^3_{0,2,1}(\mathbf{v}_8) + \\
&\quad + c^{t_j}_{1,1,2} B^3_{0,1,2}(\mathbf{v}_8) + c^{t_j}_{1,0,3} B^3_{0,0,3}(\mathbf{v}_8)
\end{aligned}$$

We then use the second continuity pair $((3,0,1),(0,0,1))$ to find the continuity condition *in which* $c^{t_k}_{\mathcal{M}_{\rho(\mathbf{v}_8)}(3,\kappa)} = c^{t_k}_{3,0,1}$ *is the continuity point B-coefficient:*

$$\begin{aligned}
c^{t_k}_{3,0,1} &= \sum_{|\gamma|=3} c^{t_j}_{\mathcal{M}_{\rho(\mathbf{v}_5)}(0,\kappa)+\gamma} B^3_\gamma(\mathbf{v}_8) \\
&= \sum_{|\gamma|=3} c^{t_j}_{(0,0,1)+\gamma} B^3_\gamma(\mathbf{v}_8) \\
&= c^{t_j}_{3,0,1} B^3_{3,0,0}(\mathbf{v}_8) + c^{t_j}_{2,1,1} B^3_{2,1,0}(\mathbf{v}_8) + c^{t_j}_{2,0,2} B^3_{2,0,1}(\mathbf{v}_8) + c^{t_j}_{1,2,1} B^3_{1,2,0}(\mathbf{v}_8) + \\
&\quad + c^{t_j}_{1,1,2} B^3_{1,1,1}(\mathbf{v}_8) + c^{t_j}_{1,0,3} B^3_{1,0,2}(\mathbf{v}_8) + c^{t_j}_{0,3,1} B^3_{0,3,0}(\mathbf{v}_8) + c^{t_j}_{0,2,2} B^3_{0,2,1}(\mathbf{v}_8) + \\
&\quad + c^{t_j}_{0,1,3} B^3_{0,1,2}(\mathbf{v}_8) + c^{t_j}_{0,0,4} B^3_{0,0,3}(\mathbf{v}_8)
\end{aligned}$$

It is now easy to check in Figure 3.6 that both expressions indeed form the correct continuity structures.

■

3.3.3 The effects of continuity

It was explained in the previous section how the continuity of B-form polynomials on neighboring simplices could be enforced using the continuity conditions. But what exactly is continuity between simplices, and what are its effects on the complete spline function?

Because a d^{th} degree B-form polynomial is a differentiable function, its continuity order is C^d at every point inside its domain. In the definition of a spline function as a member of the spline space $\mathcal{S}^r_d(\mathcal{T})$ from Eq. 2.70, therefore, the continuity order C^r always refers to the continuity order on the edge facets.

Continuity comes at an unavoidable price. By relating B-coefficients through the continuity conditions, degrees of freedom of the spline polynomials are lost proportionally with increasing continuity order. A higher continuity order always comes at the cost of a lower approximation power. A good indicator of spline approximation power is the semi-degree δ which is defined as follows:

$$\delta := d - r - 1. \tag{3.79}$$

The semi-degree should be seen as the effective degree of the B-form polynomial with C^r continuity.

An additional spline approximation power indicator is ξ_c, which is a count of the number of B-coefficients that are not used in the continuity conditions for a single edge facet:

$$\xi_c := \hat{d} - R_r, \tag{3.80}$$

with $\hat{d}$ the total number of B-coefficients from Eq. 2.51 and with R_r the total number of continuity conditions for C^r continuity on a single edge facet from Eq. 3.68.

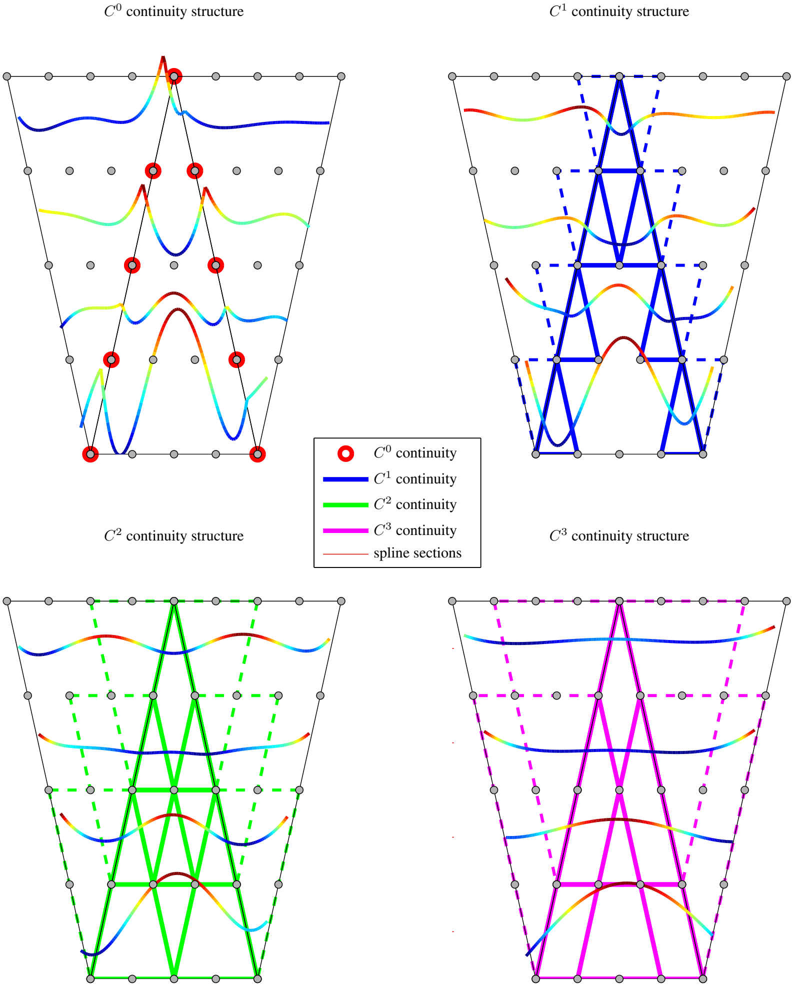
Up to this point only individual continuity conditions and their corresponding continuity structures have been considered. When C^r continuity is required at an edge facet, however, a total of R_r continuity conditions of orders C^0 to C^r are required, see Eq. 3.68. In Figure 3.7 the structures of continuity for C^0 to C^3 continuity of the 4^{th} degree spline function from Figure 3.5 and Figure 3.6 is shown. To achieve C^3 continuity across the two edge facets, the continuity conditions from C^0, C^1, C^2 and C^3 continuity should all be included in the optimization process for the B-coefficients of the spline function.

Also shown in Figure 3.7 are 1-dimensional sections of the spline function after appliance of the continuity conditions up to the given order. It is important to mention that all spline functions model the exact same objective function, which is the Mexican hat function plotted in Figure 3.8. The four plots in Figure 3.7 clearly demonstrate the effects of the continuity conditions on the resulting spline function. For C^0 continuity, for example, the spline function is discontinuous (i.e. not differentiable) at the edge facets. For C^1 continuity, this discontinuity is no longer present, and the spline function appears to be more smoothened. For C^2 and C^3 continuity, the spline function is completely smooth, so smooth in fact that one can wonder as to what degree the objective function is still being accurately approximated.

In Figure 3.8 the complete spline functions corresponding with the four continuity orders from Figure 3.7 are plotted, together with the function that is being approximated, the Mexican hat function. The Mexican hat function was designed such that none of the spline functions would be able to approximate it accurately. This was done to amplify the differences in approximation power between the four spline functions. Two important observations can be made from Figure 3.8. First, the figure demonstrates the global smoothening effect of the continuity conditions. Compare, for example, the plot of the C^0 continuous spline with the plot for the C^3 continuous spline. The former shows large gradients and many details on a local scale. The C^3 continuous spline function, on the other hand, looks very smooth and shows very little detail on a local scale. Second, the best performing spline function in this case turns out to be the C^1 continuous spline, which outperforms the C^0 continuous spline, even though it has one less degree of freedom. The large gradients in the C^0 continuous spline indicate that there are numerical instabilities in the estimator for its B-coefficients, which has a detrimental effect on its approximation power. The spline functions with higher continuity order clearly do not suffer from these numerical instabilities. This leads to another qualitative observation: increasing the continuity order of a spline increases the numerical stability of the estimator for its B-coefficients.

To summarize these findings it can be said that *increasing* the continuity order

- increases the continuity of a spline function at the edge facets of a triangulation,
- increases the global smoothness of a spline function,


C^0 continuity structure

C^1 continuity structure

C^2 continuity structure

C^3 continuity structure

Figure 3.7: Complete structures of continuity for C^0, C^1, C^2 and C^3 continuity together with 1-dimensional sections of the resulting spline function.

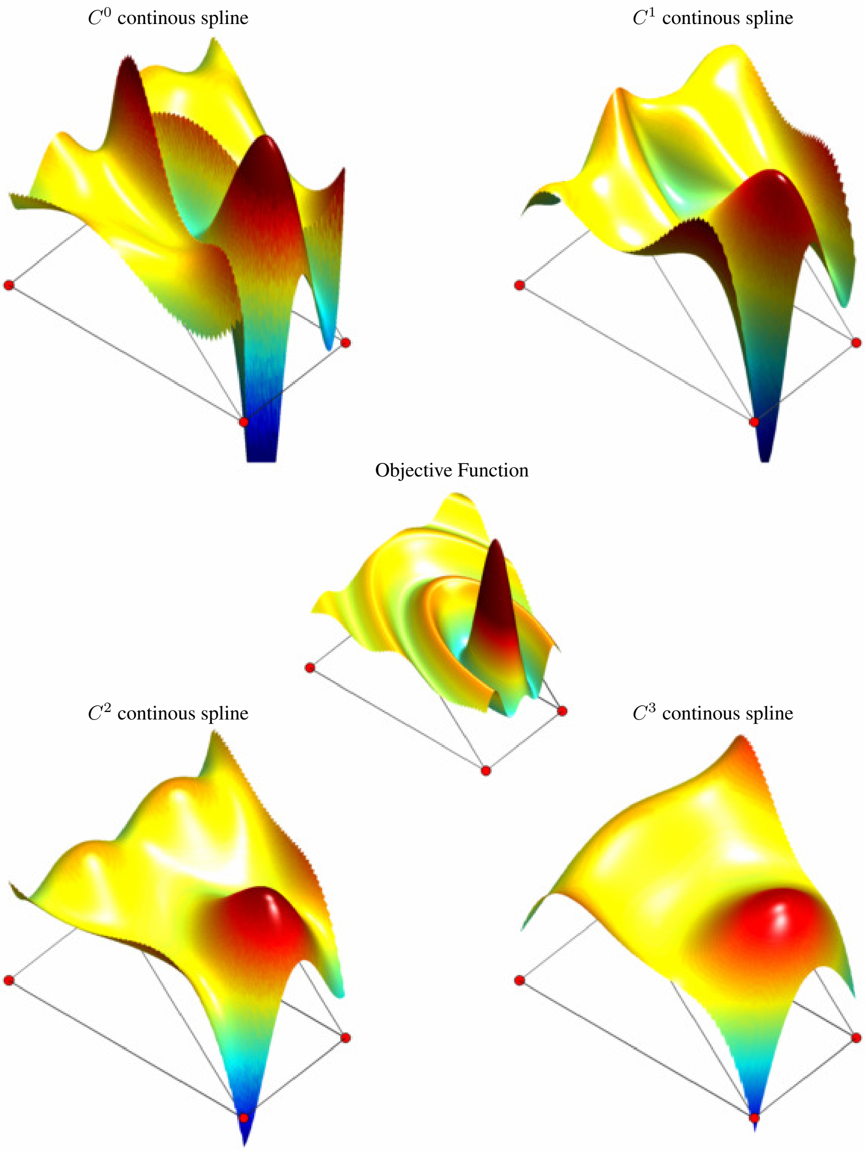


Figure 3.8: Four 4^{th} degree spline functions with continuity orders C^0, C^1, C^2 and C^3 approximating the same objective function plotted in the center.

- increases the numerical stability of the estimator for the B-coefficients of a spline function when data is scarce.
- results in a more shallow cost function,
- reduces the approximation power of a spline function,
- leads to a higher propagation factor.

The final item in this list, the propagation factor, is the subject of the next section.

3.3.4 B-net propagation

In the previous section, a qualitative analysis of the effects of continuity on the global spline function was provided. This section focuses on one specific, but highly important effect: *B-net propagation.* B-net propagation is the propagation of local disturbances in the B-net of one simplex to B-nets of other simplices. Propagation effectively transforms a simplex spline from a local into a semi-global approximator. One of the new findings in this thesis is that propagation is strongly dependent on the configuration of a triangulation. In particular, it was found that the Type I triangulation is much more susceptible to B-net propagation than the Type II triangulation.

It is found that B-net propagation consists of two components. Firstly, there is a polynomial component, which is damped proportionally with the total number of unconstrained B-coefficients at Star level-L from the perturbed simplex. Secondly, there is a geometric component, which is related to the configuration of the triangulation of the spline function, and which in some cases is undamped. For a local approximator, polynomial B-net propagation is acceptable, but geometric B-net propagation is not.

B-net propagation is demonstrated in Figure 3.9, Figure 3.10, Figure 3.11, and Figure 3.12. All spline functions are 4^{th} degree spline functions with a continuity order which is varied between C^1 and C^2. The propagation in this case is caused by fixing the values of the 15 B-coefficients of a single simplex t_p to the value 1. The spline function is then used to approximate the objective function $f(x_0, x_1) = 0$. The optimal solution in this case would be that all B-coefficients are equal to zero, according to Eq. 2.59. This means that any non-zero B-coefficients are produced entirely by the propagation effect.

The following theorem will introduce an expression for the factor G_p which determines the polynomial B-net propagation.

Theorem 7. *The polynomial component of B-net propagation at Star level-L is determined by the factor $G_p(L)$, which is defined as follows:*

$$G_p(L) := \frac{\hat{d}}{2L\xi_c} \tag{3.81}$$

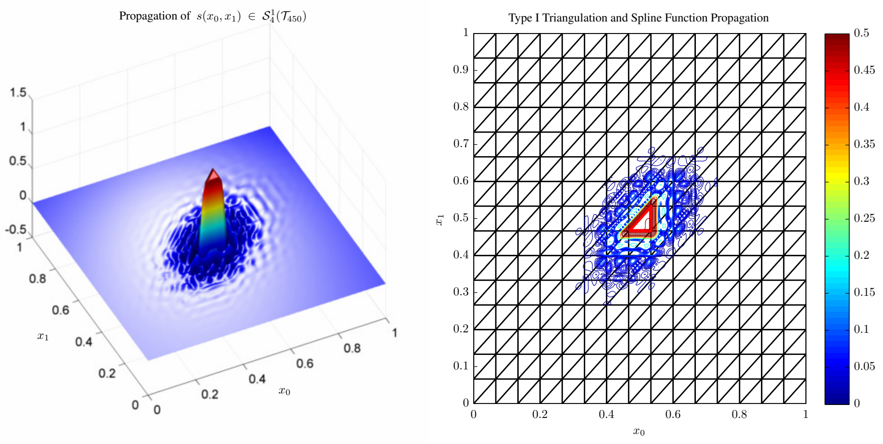


Figure 3.9: Propagation of a 4^{th} degree bivariate spline function of continuity order 1 on a Type I triangulation consisting of 450 triangles. The polynomial propagation is strongly damped, while no geometric propagation occurs. The perturbed simplex t_p is drawn in red (right).

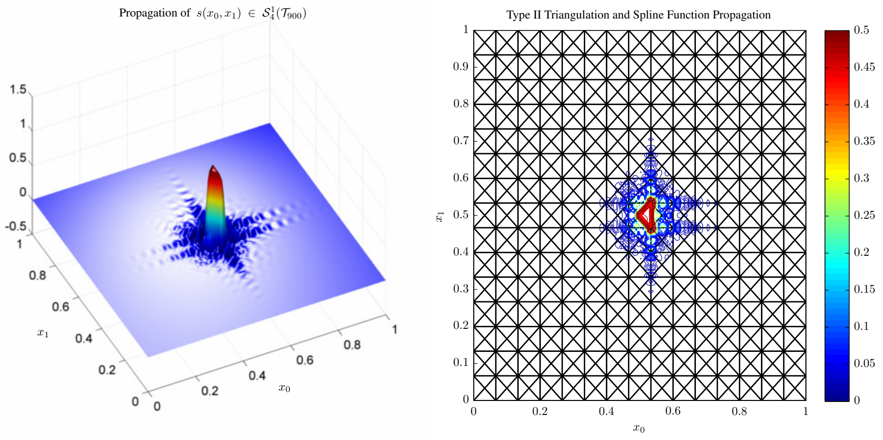


Figure 3.10: Propagation of a 4^{th} degree bivariate spline function of continuity order 1 on a Type II triangulation consisting of 900 triangles. The polynomial propagation is strongly damped, while no geometric propagation occurs. The perturbed simplex t_p is drawn in red (right)

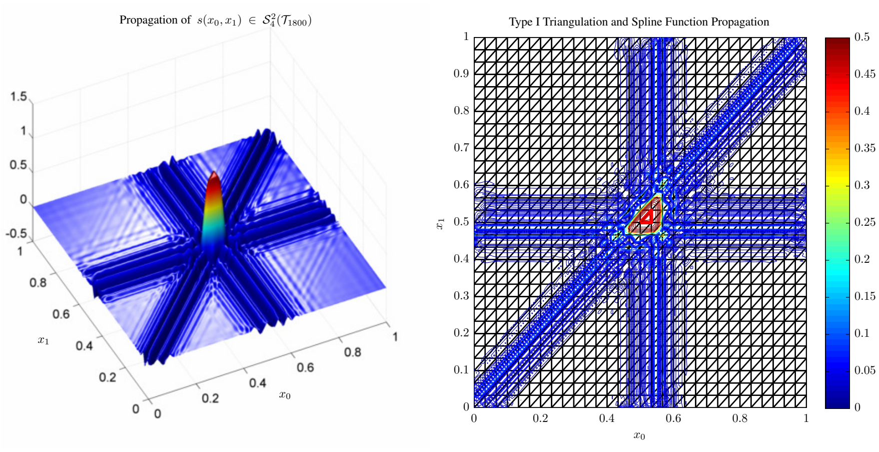


Figure 3.11: Propagation of a 4^{th} degree bivariate spline function of continuity order 2 on a Type I triangulation consisting of 1800 triangles. The geometric propagation is undamped, and is present throughout the entire spline domain. The perturbed simplex t_p is drawn in red (right)

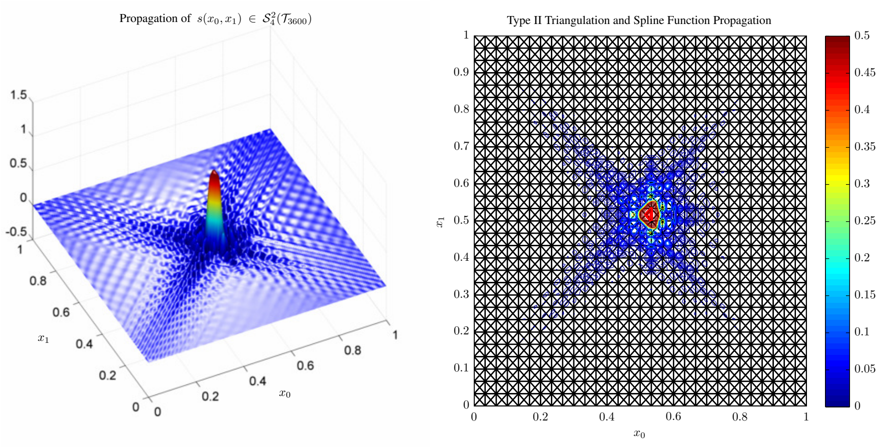


Figure 3.12: Propagation of a 4^{th} degree bivariate spline function of continuity order 2 on a Type II triangulation consisting of 3600 triangles. The geometric propagation is damped, and is present only on a part of the spline domain. The perturbed simplex t_p is drawn in red (right)

with ξ_c the total number of unconstrained B-coefficients for a simplex with a single constrained edge from Eq. 3.80.

The following ranges are found for $G_p(L)$:

$$G_p(L) \longrightarrow \begin{cases} G_p(L) \leq 1 & \Rightarrow \textit{ no propagation} \\ G_p(L) > 1 & \Rightarrow \textit{ damped propagation} \end{cases} \tag{3.82}$$

Proof. The proof of the theorem is based on an analysis of the equilibrium in approximation power between the constrained part and the unconstrained part of the B-form polynomial. The first step in the proof therefore is the decomposition of the B-form polynomial p_{t_L} in an unconstrained (free) and a constrained (fixed) part:

$$\begin{aligned} p_{t_L}(b(\mathbf{x})) &= p_{t_L}^{free}(b(\mathbf{x})) + p_{t_L}^{fix}(b(\mathbf{x})) && (3.83) \\ &= \sum_{k=1}^{|\mathcal{W}^r|} c_{\mathcal{W}_k^r}^{t_L, free} B_{\mathcal{W}_k^r}^d(b(\mathbf{x})) + \sum_{q=1}^{|\mathcal{M}^r|} c_{\mathcal{M}_q^r}^{t_L, fix} B_{\mathcal{M}_q^r}^d(b(\mathbf{x})), && (3.84) \end{aligned}$$

with $\mathcal{W}^r$ the set of multi-indices corresponding with unconstrained (free) B-coefficients and with $\mathcal{M}^r$ the set of multi-indices corresponding with constrained (fixed) B-coefficients. The sizes of $\mathcal{W}^r$ and $\mathcal{M}^r$ are related as follows:

$$|\mathcal{W}^r| + |\mathcal{M}^r| = \hat{d}, \tag{3.85}$$

It is the polynomial $p_{t_L}^{fix}$ in Eq. 3.84 that is responsible for the polynomial B-net propagation. If the polynomial $p_{t_L}^{free}$ has equal or higher approximation power than $p_{t_L}^{fix}$, then the disturbance can be fully damped on t_L. Because both $p_{t_L}^{free}$ and $p_{t_L}^{fix}$ have the same polynomial degree, the only way that the approximation power of $p_{t_L}^{free}$ can match or exceed that of $p_{t_L}^{fix}$ is if it has more polynomial terms, and therefore more B-coefficients. This implies the following:

$$|\mathcal{W}^r| \geq |\mathcal{M}^r| \implies \|p_{t_L}^{free} - p_{t_L}^{fix}\|_2 = 0. \tag{3.86}$$

It is not important what the actual members of $\mathcal{W}^r$ and $\mathcal{M}^r$ are. Instead, we are only interested in the total number of elements of $\mathcal{W}^r$ and $\mathcal{M}^r$, which are found to be:

$$\begin{aligned} |\mathcal{W}^r| &= L\xi_c && (3.87) \\ |\mathcal{M}^r| &= \hat{d} - |\mathcal{W}^r| && (3.88) \end{aligned}$$

The size of the set of free B-coefficients is determined by the total number of free B-coefficients multiplied by the Star level L. This is a direct result of the overlapping of continuity structures in which free B-coefficients are mixed with fixed B-coefficients.

We are now ready to prove the theorem. It was shown in Eq. 3.88 that the number of free B-coefficients accumulates with the Star level such that at Star level-L there are $|\mathcal{W}^r| = L\xi_c$ free B-coefficients.

From Eq. 3.86 it follows that $L\xi_c \geq |\mathcal{M}^r|$ for the disturbance to be damped out. It then follows that if $L\xi_c \geq \hat{d}/2$, the approximation power of $p_{t_L}^{free}$ matches or exceeds that of $p_{t_L}^{fix}$. With the definition of $G_p(L)$ from Eq. 3.81 we can then derive the ranges presented in Eq. 3.82:

$$L\xi_c \geq \frac{\hat{d}}{2} \implies 1 \geq \frac{\hat{d}}{2L\xi_c} = G_p(L), \tag{3.89}$$

which proves the theorem.

□

Using Eq. 3.81 and Eq. 3.82 we find the following expression for the Star level L_{damp} at which the polynomial B-net propagation is fully damped:

$$L_{damp} \geq \frac{\hat{d}}{2\xi_c} \tag{3.90}$$

In Figure 3.13 an example is shown of polynomial B-net propagation. In the figure a spline function $s \in \mathcal{S}_2^1$ defined on a Type II triangulation is shown. In this case, $\hat{d} = 6$ and $\xi_c = 1$, so according to Eq. 3.90 $L_{damp} \geq 3$. Therefore, the polynomial B-net propagation can be considered to be fully damped beyond Star level-3. When observing Figure 3.13, it may seem that there still is plenty of B-net propagation beyond Star level-3. Upon closer inspection, however, the propagation never exceeds the 5% level outside the Star level-3 bounds.

Using the general formulation of the continuity conditions from Eq. 3.76, a more exact expression for polynomial B-net propagation between simplices of Star level-L and Star level-$(L-1)$ can be derived. These expressions are meant for the precise calculation of local polynomial propagation for a given disturbance.

Let the B-coefficients $c_\kappa^{t_j}$ of a spline polynomial of degree d with C^r continuity on a simplex t_j be perturbed with the disturbance function ϕ_{t_j} as follows:

$$c_\kappa^{t_j} = \phi_{t_j}(b(c_\kappa^{t_j})), \; |\kappa| = d. \tag{3.91}$$

For Star level-1, the fixed B-coefficients $c_{\mathcal{M}^r}^{t_1,fix}$ from Eq. 3.84 then are completely determined by the general continuity conditions from Eq. 3.76 between the simplex t_1 and the simplex t_j as follows:

$$c_{\mathcal{M}_\rho(m,\kappa)}^{t_1,fix} = \sum_{|\gamma|=m} \phi_{t_j}\left(b\left(c_{\mathcal{M}_\rho(0,\kappa)+\gamma}^{t_j}\right)\right) B_\gamma^m(\tilde{\mathbf{v}}_{t_1,t_j}), \; 0 \leq m \leq r, \tag{3.92}$$

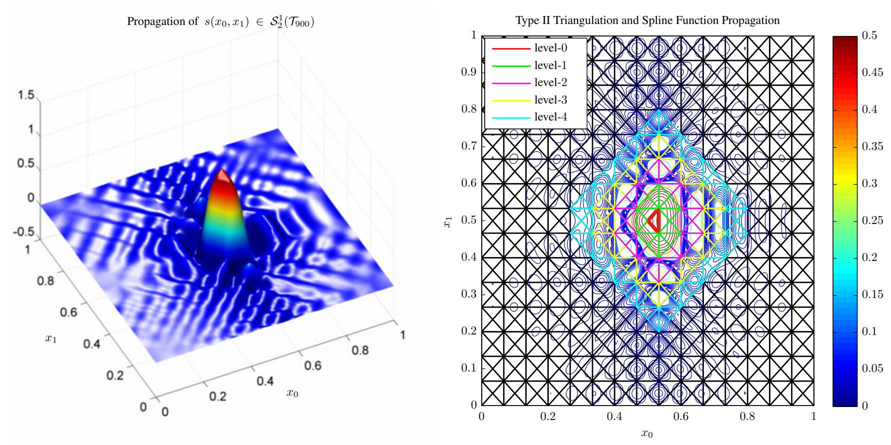


Figure 3.13: The propagation of a 2^{nd} degree spline function with C^1 continuity together with the Star structures of the perturbed simplex on a Type II triangulation consisting of 900 triangles. The polynomial propagation $G_p(L)$ is damped beyond Star level-3.

in which the argument lists of the rank functions ρ from Eq. 3.74 are dropped to aid readability.

For Star level-L, the fixed B-coefficients $c^{t_L,fix}_{\mathcal{M}^r}$ from Eq. 3.84 are completely determined by the general continuity conditions from Eq. 3.76 between the simplex t_L and the simplex $t_{(L-1)}$ as follows:

$$c^{t_L,fix}_{\mathcal{M}_\rho(m,\kappa)} = \sum_{|\gamma|=m} c^{t_{(L-1)}}_{\mathcal{M}_\rho(0,\kappa)+\gamma} B^m_\gamma(\tilde{\mathbf{v}}_{t_L,t_{(L-1)}}),\ 0 \leq m \leq r,\ L > 1, \tag{3.93}$$

It is important to note in Eq. 3.93 that the B-coefficients $c^{t_{(L-1)}}_{\mathcal{M}_\rho(0,\kappa)+\gamma}$ contain at least a single free B-coefficient. This is caused by the incomplete overlap of the continuity structures of neighboring simplices. As a result, Eq. 3.93 cannot be applied in a recursive form.

The second B-net propagation component is geometric B-net propagation. Geometric B-net propagation is an effect that is present in a special class of partially symmetric triangulations of dimension $n \geq 2$, like the Type I triangulation. Fully symmetric triangulations like the Type II triangulation are much less susceptible to geometric propagation. Geometric propagation is caused by a recursive interaction between identical continuity conditions on neighboring simplices. There are cases for which the Geometric B-net propagation on Type I triangulations is undamped, affecting the entire triangulation. In Figure 3.11 and Figure 3.14 two examples are given of geometric B-net propagation.

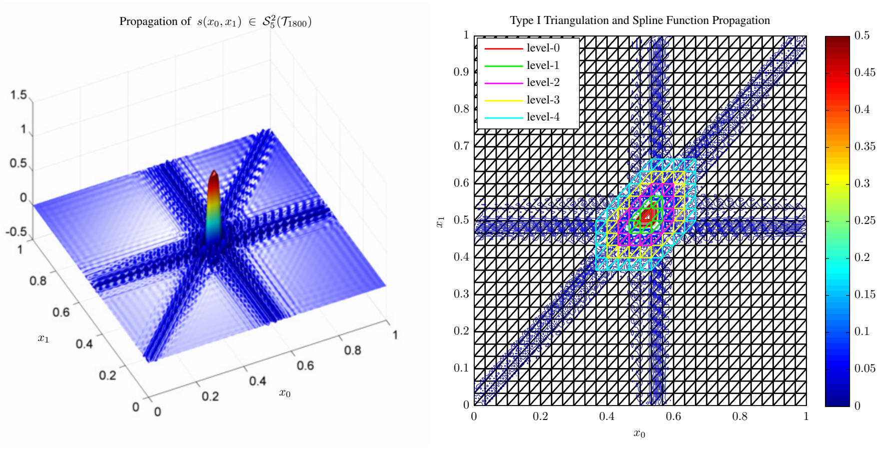


Figure 3.14: The propagation of a 5^{th} degree spline function with C^2 continuity together with the Star structures of the perturbed simplex on a Type I triangulation. The geometric propagation $G_{\mathcal{T}}$ is undamped and present across the entire triangulation.

From Figure 3.14 it is clear that the simplex Star does not have any influence on geometric propagation.

The following theorem presents an expression for geometric propagation.

Theorem 8. *The geometric component of B-net propagation is determined by $G_{\mathcal{T}}$ which is defined as follows:*

$$G_{\mathcal{T}} = \frac{3r}{d}, \quad 0 \leq G_{\mathcal{T}} \leq 3 - \frac{3}{d}, \tag{3.94}$$

where the following ranges are found for $G_{\mathcal{T}}$:

$$G_{\mathcal{T}} \longrightarrow \left\{ \begin{array}{rlcl} 0 \leq & G_{\mathcal{T}} < 1 & \Rightarrow & \textit{no propagation} \\ & G_{\mathcal{T}} = 1 & \Rightarrow & \textit{critically damped propagation} \\ & G_{\mathcal{T}} > 1 & \Rightarrow & \textit{undamped propagation} \end{array} \right. . \tag{3.95}$$

Proof. The proof of the theorem is based on the observation that geometric propagation only occurs when the C^r continuity structures within an n-simplex t_j overlap completely. Geometric propagation can therefore be prevented by choosing an order of continuity for which the overlap of the continuity structures is incomplete, see Figure 3.15. The continuity structure coverage is incomplete when there is a set of $n + 1$ B-coefficients in the center of the B-net which are covered by at most n continuity conditions.

The derivation of the expression Eq. 3.94 uses the fact that the set of $n + 1$ special B-coefficients forms a triangle $t_\triangle$ with a base consisting of $r+2$ B-coefficients on any ridge $\bar{t}$ of t_j. This triangle is present in any simplex with an incomplete coverage of continuity structures. For symmetry, there must be a total of *at least* r B-coefficients located on the line that connects the base of $t_\triangle$ with the closest vertex in t_j, see Figure 3.15.

Therefore, in order to support the existence of $t_\triangle$, the total number of B-coefficients in $\bar{t}$ is bound by the following inequality:

$$d+1 >= (r+2)+2r, \tag{3.96}$$

with the part $(r+2)$ the total number of B-coefficients in the base of $t_\triangle$, and with $2r$ the number of B-coefficients on the two shortest lines that connect the base of $t_\triangle$ with the two closest vertices in t_j.

This leads to the expression for damped geometric propagation:

$$\frac{3r+1}{d} \leq 1 \tag{3.97}$$

A special situation occurs when there is a single central B-coefficient which is present in at most $n+1$ continuity equations. In this case, geometric propagation is found to be critically damped. With the same reasoning as before, the single B-coefficient forms the peak of a triangle $t_\triangle$, but this time with a base consisting of only $r+1$ B-coefficients on any ridge $\bar{t}$ of t_j. Again for symmetry, a total of r B-coefficients must be located on the line that connects the base of $t_\triangle$ with the closest vertex in t_j, see Figure 3.15.

$$d+1 = (r+1)+2r \tag{3.98}$$

which leads to the expression for critically damped geometric propagation:

$$\frac{3r}{d} = 1 \tag{3.99}$$

Combining Eq. 3.97 with Eq. 3.99 and observing that $\frac{3r}{d} < \frac{3r+1}{d}$ for positive d and r we get the following expression:

$$\frac{3r}{d} < \frac{3r+1}{d} \leq 1 \tag{3.100}$$

Therefore, geometric propagation is avoided when the following inequality holds:

$$\frac{3r}{d} < 1 \tag{3.101}$$

Finally, Eq. 3.99 together with Eq. 3.101 results in the ranges for $G_{\mathcal{T}}$, and proves the theorem.

□

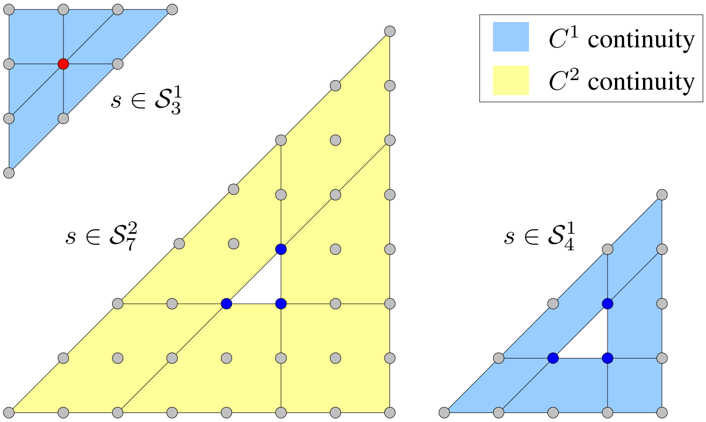


Figure 3.15: The coverage of the C^1 and C^2continuity structures for the B-nets of a 3^{rd} degree (top left), a 7^{th} degree (center) and a 4^{th} degree (right) spline function. For the 3^{rd} degree spline function we have $G_{\mathcal{T}} = \frac{3r}{d} = 1$, which means geometric propagation is critically damped. For the 7^{th} and 4^{th} degree spline functions we have $G_{\mathcal{T}} = \frac{3r}{d} < 1$, and therefore, no geometric propagation will occur. In the figure, the blue dots are special B-coefficients that are covered by at most n continuity conditions, while the red dot is the single B-coefficient which is covered by at most $n+1$ continuity conditions

Concluding this section, Figure 3.16 demonstrates a spline function $s \in \mathcal{S}_3^1$ for which $G_{\mathcal{T}} = 1$, that is, a spline function with critically damped geometric propagation.

3.3.5 The Smoothness Matrix

In the end all continuity conditions for all edges in a triangulation must be included in a matrix called the *smoothness matrix*. In this section the construction of the smoothness matrix will be discussed.

Let $\mathbf{H}$ be the smoothness matrix, which is used together with the global vector of B-coefficients $\mathbf{c}$ from Eq. 3.27 such that a homogeneous set of equations is produced:

$$\mathbf{H}\mathbf{c} = 0. \tag{3.102}$$

Each row in $\mathbf{H}$ contains a single continuity condition Eq. 3.76 which is equated to zero as follows:

$$-c^{t_f}_{\mathcal{M}_{\rho(\tilde{\mathbf{v}}_{f,g})(m,\kappa)}} + \sum_{|\gamma|=m} c^{t_g}_{\mathcal{M}_{\rho(\tilde{\mathbf{v}}_{g,f})(0,\kappa)+\gamma}} B^m_\gamma(\tilde{\mathbf{v}}_{f,g}) = 0, \quad 0 \leq m \leq r. \tag{3.103}$$

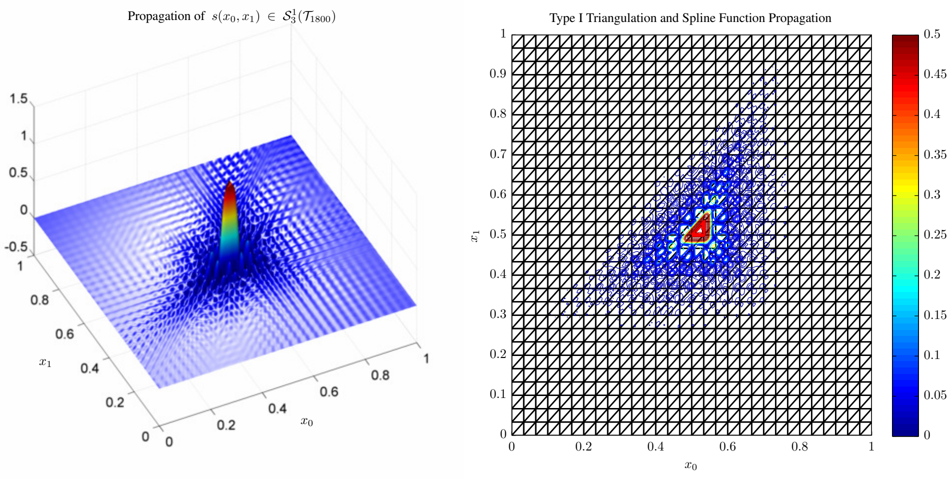


Figure 3.16: An example of critically damped geometric propagation: a 3^{rd} degree spline function with C^1 continuity defined on a Type I triangulation.

With C^r continuity between simplices it is found that $\mathbf{H} \in \mathbb{R}^{(E \cdot R_r) \times (J \cdot \hat{d})}$, with E the total number of edges in a triangulation and R_r and $\hat{d}$ as in Eq. 3.68 and Eq. 2.51 respectively.

In Figure 3.17 the smoothness matrix for C^2 continuity of the spline function $s \in \mathcal{S}_4^2(\mathcal{T}_{1800})$ is shown. Note that this is the smoothness matrix of the spline function which propagation was plotted in Figure 3.11. In this figure, every blue dot corresponds with a non-zero B-coefficient value in $\mathbf{H}$. The smoothness matrix has a very high sparseness factor which increases with the size of the triangulation and the polynomial degree. The smoothness matrix in Figure 3.17, for example, contains 124080 non-zero values spread out over a total of 31680 continuity conditions. The total number of elements of $\mathbf{H}$ in this case is more than 855 million, which leads to a sparseness factor that is more than 99.98%.

The rank of the smoothness matrix is given by:

$$\text{rank } \mathbf{H} \leq (E \cdot R_r) \tag{3.104}$$

However, only for the simplest of triangulations will $\mathbf{H}$ be of full rank. As Lai and Schumaker observed in [104], the rank deficiency of $\mathbf{H}$ is caused by the fact that there are redundant continuity equations for triangulations with an interior vertex. Rank deficiencies in $\mathbf{H}$ are easily produced, even for very simple triangulations and low continuity orders, such as that shown in Figure 3.18. In this case, the triangulation $\mathcal{T}_3$ has three edges, and thus three sets of C^0 continuity equations. At the internal vertex $\mathbf{v}_3$, we get three C^0 continuity

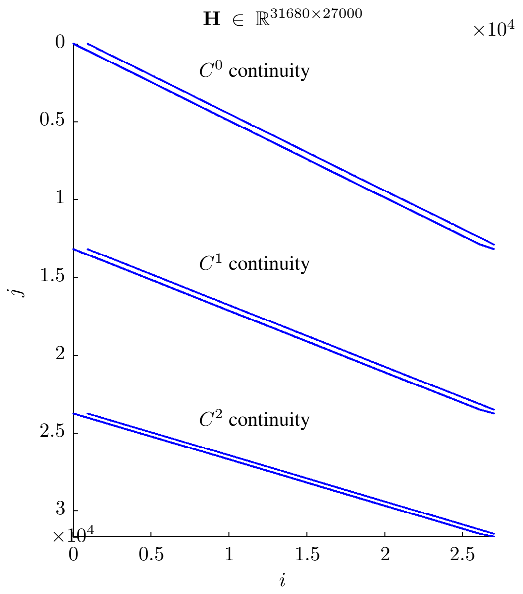


Figure 3.17: The rank-deficient smoothness matrix $\mathbf{H}$ for C^2 continuity for the spline function $s \in \mathcal{S}_4^2(\mathcal{T}_{1800})$. The bandedness of $\mathbf{H}$ is the result of formulating the continuity conditions per-continuity order, instead of per-edge.

equations which using Eq. 3.76 are found to be:

$$\begin{aligned} \tilde{t}_{1,2} &\longrightarrow c^{t_1}_{0,0,1} = c^{t_2}_{0,0,1} \\ \tilde{t}_{1,3} &\longrightarrow c^{t_1}_{0,0,1} = c^{t_3}_{0,0,1} \\ \tilde{t}_{2,3} &\longrightarrow c^{t_2}_{0,0,1} = c^{t_3}_{0,0,1}. \end{aligned} \tag{3.105}$$

It is easy to check that one of these three equations is redundant. For example, when the equation for $\tilde{t}_{1,2}$ is substituted in the equation for $\tilde{t}_{1,3}$ the result is $c^{t_2}_{0,0,1} = c^{t_3}_{0,0,1}$, which is equal to the equation for $\tilde{t}_{2,3}$, which has therefore become redundant.

Constructing a smoothness matrix of full rank is not a trivial task. Redundancies do not only arise between continuity conditions of equal order, but also between conditions of different orders. For example, it can be shown that in some cases a C^r continuity condition can be formed by combining a number of C^{r-1} conditions. This makes the creation of an analytical algorithm for formulating full rank smoothness matrices a very challenging task.

In this thesis a more practical approach to creating full rank smoothness matrices is taken. Small to medium sized smoothness matrices containing less than 1000 continuity conditions can be transformed into a row reduced echelon formation, after which the rows containing only zeros can be removed. The problem with this approach is that it destroys the sparseness of the original smoothness matrix and also introduces roundoff errors in the new smoothness matrix.

For larger smoothness matrices a different approach is taken which is based on an estimate of the condition number of $\mathbf{H}$. The condition number estimator is very efficient computationally, and can handle very large (sparse) matrices, see [73]. The algorithm for constructing a full rank smoothness matrix starts by initializing $\mathbf{H}$ with a single continuity condition. Then, a new candidate smoothness matrix $\mathbf{H}_c$ is created by adding a new continuity condition $\mathbf{h}$ as follows:

$$\mathbf{H}_c = \begin{bmatrix} \mathbf{H} \\ \mathbf{h} \end{bmatrix} \tag{3.106}$$

The condition number estimator requires a square matrix, while in general $\mathbf{H}$ and $\mathbf{H}_c$ are non-square. A square matrix can be obtained by multiplying $\mathbf{H}_c$ with its transpose. As a rule, the rank of the product of two matrices is equal to, or less than, the rank of the individual matrices, see e.g. [74]. Therefore, the following statement holds:

$$\text{rank } \mathbf{H}_c \geq \text{rank } \mathbf{H}_c\mathbf{H}_c^\top \tag{3.107}$$

This means that a conservative estimate of the condition number of the candidate smoothness matrix $\mathbf{H}_c$ can be obtained by estimating the condition number of $\mathbf{H}_c\mathbf{H}_c^\top$. Now, if $\mathbf{H}_c$ is singular, this condition number will be some very large number. In this thesis it is assumed that any candidate smoothness matrix with an estimated condition number larger than 10^{10} is singular. In that case, it can be reasoned that the continuity condition $\mathbf{h}$ caused $\mathbf{H}_c$ to become singular, which means that $\mathbf{h}$ is redundant and should be dropped. If the condition number has a lower value than 10^{10}, it is safe to assume that $\mathbf{h}$ is not redundant. In that case the candidate matrix becomes the new smoothness matrix, and the process can be repeated until all $E \cdot R_r - 1$ continuity conditions have been checked for redundancy.

Example 16 (Construction of a smoothness matrix)**.** *In this example, the complete* $\mathbf{H}$ *matrix for* C^0 *continuity for the spline function* $s \in \mathcal{S}_1^0(\mathcal{T}_3)$ *defined on the triangulation shown in Figure 3.18 is constructed. The continuity conditions for the given triangulation are formulated using Eq. 3.76.*

The continuity conditions of t_1 *with respect to* t_2 *are:*

$$c^{t_1}_{(\kappa_0,0,\kappa_1)} = \sum_{|\gamma|=0} c^{t_2}_{(\kappa_0,0,\kappa_1)} B^0_\gamma(\mathbf{v}_1) = c^{t_2}_{(\kappa_0,0,\kappa_1)},$$

which leads to the continuity conditions

$$c^{t_1}_{1,0,0} = c^{t_2}_{(1,0,0)}, \;\; c^{t_1}_{0,0,1} = c^{t_2}_{(0,0,1)}.$$

The continuity conditions of t_1 *with respect to* t_3 *are given by:*

$$c^{t_1}_{(0,\kappa_0,\kappa_1)} = \sum_{|\gamma|=0} c^{t_3}_{(\kappa_0,0,\kappa_1)} B^0_\gamma(\mathbf{v}_0) = c^{t_2}_{(\kappa_0,0,\kappa_1)}.$$

leading to the continuity conditions

$$c^{t_1}_{0,1,0} = c^{t_3}_{(1,0,0)}, \quad c^{t_1}_{0,0,1} = c^{t_3}_{(0,0,1)}.$$

Finally, the continuity conditions of t_2 *with respect to* t_3 *are:*

$$c^{t_2}_{(0,\kappa_0,\kappa_1)} = \sum_{|\gamma|=0} c^{t_3}_{(0,\kappa_0,\kappa_1)} B^0_\gamma(\mathbf{v}_0) = c^{t_2}_{(0,\kappa_0,\kappa_1)}.$$

which results in the continuity conditions

$$c^{t_2}_{0,1,0} = c^{t_3}_{(0,1,0)}, \quad c^{t_2}_{0,0,1} = c^{t_3}_{(0,0,1)}.$$

The global vector of B-coefficients is:

$$\mathbf{c} = \begin{bmatrix} c^{t_1}_{1,0,0} & c^{t_1}_{0,1,0} & c^{t_1}_{0,0,1} & c^{t_2}_{1,0,0} & c^{t_2}_{0,1,0} & c^{t_2}_{0,0,1} & c^{t_3}_{1,0,0} & c^{t_3}_{0,1,0} & c^{t_3}_{0,0,1} \end{bmatrix}^\top.$$

It can then be checked that the complete, full rank smoothness matrix $\mathbf{H}$ *for* C^0 *continuity for this example is:*

$$\mathbf{H} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Note that the continuity condition $c^{t_2}_{0,0,1} = c^{t_3}_{0,0,1}$ *was rejected because it was redundant as was explained in Eq. 3.105.*

3.4 Differential Constraints

In this section, a new type of constraint for simplex splines is introduced in the form of the differential constraints. Differential constraints locally constrain the directional derivatives of the spline polynomials. The differential constraints can be used, for example, to limit polynomial divergence in areas of the spline domain with insufficient data coverage or to enforce differential boundary conditions on the edges of the spline domain. In order to

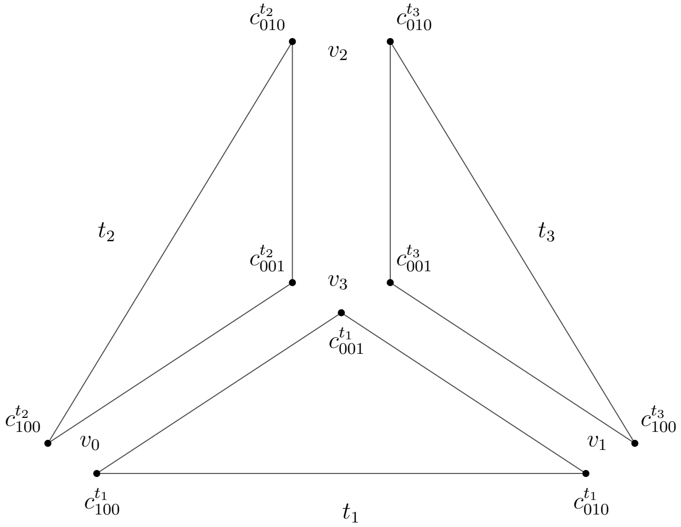


Figure 3.18: Exploded view of a triangulation $\mathcal{T}_3$ consisting of 3 simplices with 1^{st} degree B-nets. The triangulation contains a single interior vertex $\mathbf{v}_3$.

define differential constraints on the B-coefficients of a spline function, an expression is required for the directional derivatives of the simplex splines in terms of the original vector of B-coefficients. It will be shown in this section, that the derivation of the directional derivatives in terms of the original B-coefficients is based on the one-step matrix form of the de Casteljau algorithm introduced in Sec. 3.1. The theory presented in this section is based on the work presented in [41].

This section consists of three subsections. The first, Sec. 3.4.1, introduces the concept of differential constraints. Following the formal definition of the differential constraints, a theory for the required total number of differential constraints for constraining the spline polynomials along edges and sub-simplices is presented in Sec. 3.4.2. The section concludes with an in depth analysis of bounded model extrapolation in Sec. 3.4.3.

3.4.1 Definition of differential constraints

In [41] it was proved that the differential constraints can be expressed in the form of the following linear matrix equation:

$$\mathbf{D} \cdot \mathbf{c} = \mathbf{d}, \tag{3.108}$$

with $\mathbf{D}$ a block diagonal matrix holding M differential constraints for the optimization problem. The vector $\mathbf{c}$ is the global vector of B-coefficients from Eq. 3.27, while $\mathbf{d} \in \mathbb{R}^{M \times 1}$ is the vector of differential constraint values.

The blocks of $\mathbf{D}$ are formed by evaluating the differential constraint matrix function $\mathbf{D}_m^{t_j}(\mathbf{a}, \mathbf{b})$ for any simplex t_j:

$$\mathbf{D} = \text{diag}\left(\mathbf{D}_m^{t_j}(\mathbf{a}, \mathbf{b}), 1 \le j \le J\right) \in \mathbb{R}^{M \times J \cdot \hat{d}} \tag{3.109}$$

with the matrix function $\mathbf{D}_m^{t_j}(\mathbf{a}, \mathbf{b})$ defined using Eq. 3.45 as follows:

$$\mathbf{D}_m^{t_j}(\mathbf{a}, \mathbf{b}) := \frac{d!}{(d-m)!}\mathbf{B}^{d-m}(\mathbf{b})\mathbf{P}^{d,d-m}(\mathbf{a}), \tag{3.110}$$

with $\mathbf{P}^{d,d-m}(\mathbf{a})$ the de Casteljau matrix from Eq. 3.20 of degree d to $d-m$ expressed in terms of the directional coordinate $\mathbf{a}$ of the constraint direction $\mathbf{u}$ and with $\mathbf{B}^{d-m}(\mathbf{b})$ the vector of basis polynomials from Eq. 2.66.

For every differential constraint, $\mathbf{D}_m^{t_j}(\mathbf{a}, \mathbf{b})$ is evaluated for a given differential order m, directional coordinate $\mathbf{a}$ and evaluation location $\mathbf{b}$ such that

$$\mathbf{D}_m^{t_j}(\mathbf{a}, \mathbf{b}) \cdot \mathbf{c}^{t_j} = D_u^m p(\mathbf{b})\Big|_{t_j} = \mathbf{d}_i. \tag{3.111}$$

with $\mathbf{d}_i$ a single constraint value.

Example 17 (Formulating differential constraints)**.** *In this example a single differential constraint is derived for a bivariate 2^{nd} degree B-form polynomial on a single simplex, with vertex coordinates as defined in Figure 3.19. We want to constrain the B-form polynomial with a first order (i.e. $m = 1$) differential constraint at the vertex $\mathbf{v}_1$ in the direction $\mathbf{u} = (0, 1)$.*

In this case the differential constraint matrix function from Eq. 3.110 becomes

$$\mathbf{D}_1^{t}(\mathbf{a}, \mathbf{b}) = \frac{2!}{(2-1)!}\mathbf{B}^{2-1}(b(\mathbf{v}_1))\mathbf{P}^{2,2-1}(b(\mathbf{u})), \tag{3.112}$$

with $b(\mathbf{v}_1)$ the barycentric coordinate of $\mathbf{v}_1$, and with $a(\mathbf{u})$ the directional coordinate of $\mathbf{u}$. Using the definition of the barycentric coordinate transformation from Eq. 2.41 and Eq. 2.36 we find $b(\mathbf{v}_1) = (0, 1, 0)$. Using the definition of the directional coordinate from Eq. 3.42, we find $b(\mathbf{u}) = \mathbf{a} = (-1, -1, 2)$. Using the definition of the vector of basis polynomials from Eq. 2.66, we find for $\mathbf{B}^{2-1}(b(\mathbf{v}_1))$:

$$\mathbf{B}^{2-1}(\mathbf{b}) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \tag{3.113}$$

The de Casteljau matrix $\mathbf{P}^{2,2-1}(b(\mathbf{u}))$, which reduces a set of B-coefficients of degree $d = 2$

*to degree $d - m = 1$, can be constructed using Eq. 3.17. For this, we use the same procedure as that in **Example** 10, which results in the following table of values.*

$i(\kappa)$	$i(\theta)$	θ	κ	$\theta - \kappa$	$P^m_{\theta-\kappa}(\mathbf{a})$
1	1	$(2,0,0)$	$(1,0,0)$	$(1,0,0)$	$P^1_{1,0,0}(\mathbf{a})$
1	2	$(1,1,0)$	$(1,0,0)$	$(0,1,0)$	$P^1_{0,1,0}(\mathbf{a})$
1	3	$(1,0,1)$	$(1,0,0)$	$(0,0,1)$	$P^1_{0,0,1}(\mathbf{a})$
1	4	$(0,2,0)$	$(1,0,0)$	$(-1,2,0)$	0
1	5	$(0,1,1)$	$(1,0,0)$	$(-1,1,1)$	0
1	6	$(0,0,2)$	$(1,0,0)$	$(-1,0,2)$	0
2	1	$(2,0,0)$	$(0,1,0)$	$(2,-1,0)$	0
2	2	$(1,1,0)$	$(0,1,0)$	$(1,0,0)$	$P^1_{1,0,0}(\mathbf{a})$
2	3	$(1,0,1)$	$(0,1,0)$	$(1,-1,1)$	0
2	4	$(0,2,0)$	$(0,1,0)$	$(0,1,0)$	$P^1_{0,1,0}(\mathbf{a})$
2	5	$(0,1,1)$	$(0,1,0)$	$(0,0,1)$	$P^1_{0,0,1}(\mathbf{a})$
2	6	$(0,0,2)$	$(0,1,0)$	$(0,-1,2)$	0
3	1	$(2,0,0)$	$(0,0,1)$	$(2,0,-1)$	0
3	2	$(1,1,0)$	$(0,0,1)$	$(1,1,-1)$	0
3	3	$(1,0,1)$	$(0,0,1)$	$(1,0,0)$	$P^1_{1,0,0}(\mathbf{a})$
3	4	$(0,2,0)$	$(0,0,1)$	$(0,2,-1)$	0
3	5	$(0,1,1)$	$(0,0,1)$	$(0,1,0)$	$P^1_{0,1,0}(\mathbf{a})$
3	6	$(0,0,2)$	$(0,0,1)$	$(0,0,1)$	$P^1_{0,0,1}(\mathbf{a})$

Using $P^1_{1,0,0}(\mathbf{a}) = a_0^1 a_1^0 a_2^0 = -1$, $P^1_{0,1,0}(\mathbf{a}) = a_0^0 a_1^1 a_2^0 = -1$, and $P^1_{0,0,1}(\mathbf{a}) = a_0^0 a_1^0 a_2^1 = 2$, the de Casteljau matrix is:

$$
\begin{aligned}
\mathbf{P}^{2,2-1}(\mathbf{a}) &= \begin{bmatrix} P^1_{1,0,0}(\mathbf{a}) & P^1_{0,1,0}(\mathbf{a}) & P^1_{0,0,1}(\mathbf{a}) & 0 & 0 & 0 \\ 0 & P^1_{1,0,0}(\mathbf{a}) & 0 & P^1_{0,1,0}(\mathbf{a}) & P^1_{0,0,1}(\mathbf{a}) & 0 \\ 0 & 0 & P^1_{1,0,0}(\mathbf{a}) & 0 & P^1_{0,1,0}(\mathbf{a}) & P^1_{0,0,1}(\mathbf{a}) \end{bmatrix} \\
&= \begin{bmatrix} -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix}
\end{aligned}
\tag{3.114}
$$

Substituting Eq. 3.114 and Eq. 3.113 in the expression for the differential constraint matrix function from Eq. 3.112 we get:

$$
\begin{aligned}
\mathbf{D}_1^t(b(\mathbf{u}), b(\mathbf{v}_1)) &= 2 \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix} \\
&= \begin{bmatrix} 0 & -2 & 0 & -2 & 4 & 0 \end{bmatrix}
\end{aligned}
$$

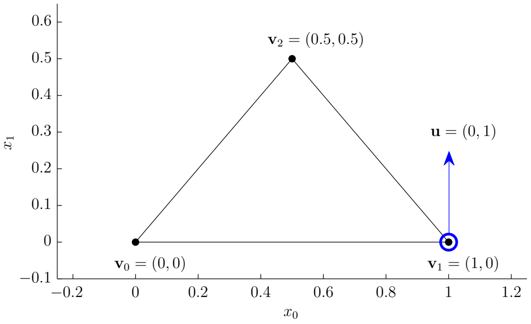


Figure 3.19: Depiction of the problem described in **Example** 17. The spline function has a single differential constraint at the location of $\mathbf{v}_1$ in the direction $(0, 1)$.

Using these results, the complete constraint matrix equation from Eq. 3.108 is:

$$\mathbf{D} \cdot \mathbf{c} = \begin{bmatrix} 0 & -2 & 0 & -2 & 4 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_{2,0,0} & c_{1,1,0} & c_{1,0,1} & c_{0,2,0} & c_{0,1,1} & c_{0,0,2} \end{bmatrix}^\top$$

This equation constrains the first order derivative of the second degree B-form polynomial at the vertex $\mathbf{v}_1$ *in the direction* $\mathbf{u} = (0, 1)$.

■

3.4.2 Differentially constraining polynomials on subsimplices

In this section it is proved that simplex polynomials can be differentially constrained on any $(n-k)$-dimensional sub-simplex t_s contained within the convex hull of the n-simplex t by formulating the constraints for a specific number of points located in t_s. A specific case of this proof was first provided in [41] where it was shown that a polynomial can be differentially constrained along simplex edge facets if constraints are imposed on N_δ unique points in the edge facet. It was shown that in this case, N_δ is given by:

$$N_\delta = \frac{(d - m + n - 1)!}{(n-1)!(d-m)!}. \tag{3.115}$$

A new theorem will be now be presented that generalizes the results in [41] to all $(n-k)$-dimensional subsimplices $t_s \in t$.

Theorem 9. *The directional derivative in the direction* $\mathbf{u}$ *of a B-form polynomial* $D_u^m p(\mathbf{b})$ *on a simplex* t_s *of dimension* $n-k$*, contained within the convex hull of the* n*-simplex* t*, is* *fully constrained at a value* $\mathbf{d}$ *by constraining a total of* N_δ *unique points in* t_s*, where* N_δ *is given by:*

$$N_\delta := \frac{(d-m+n-k)!}{(n-k)!(d-m)!}. \tag{3.116}$$

Proof. From Eq. 3.45 it is clear that the directional derivative $D_u^m p(\mathbf{b})$ is a polynomial of degree $d-m$ in $\mathbf{b}$. Now let $q(\mathbf{b}_s)$ be a B-form polynomial of degree $d-m$ which is defined on the $(n-k)$-sub-simplex t_s as follows:

$$q(\mathbf{b}_s) = \sum_{|\kappa|=d-m} c_\kappa^{t_s} B_\kappa^{d-m}(\mathbf{b}_s) \tag{3.117}$$

with $\mathbf{b}_s$ barycentric coordinates on t_s. Using Eq. 2.51, it follows that $q(\mathbf{b}_s)$ consists of a linear combination of N_δ basis polynomials and B-coefficients, with N_δ as in Eq. 3.116. The complete polynomial $q(\mathbf{b}_s)$ is therefore fully constrained when N_δ constraint equations are present, which proves the theorem.

□

3.4.3 Differential constraints for bounded model extrapolation

Model extrapolation is the expansion of the domain of a model using the information present in the model. In general, a model can only provide accurate results inside the model domain. The model can be extrapolated, however, such that it can make predictions outside the bounds of the model domain. The region outside the model domain inside which valid predictions can be made is defined as the valid extrapolation region, or VER. The volume of the VER is determined primarily by the complexity of the modeled system and the physical accuracy of the model. Pure physical models of simple systems result in the largest VER's while black box phenomenological models of complex systems have extremely limited VER's. For example, the orbit of a planet around the sun can be extrapolated accurately by using Newtonian mechanics together with a few measurements. In contrast, a purely phenomenological aerodynamic model of an aircraft can hardly be extrapolated with any accuracy at all, especially in nonlinear areas of the flight envelope [46].

Models based on simplex splines are pure phenomenological models. They are created by fitting, in some optimal sense, a set of special polynomials to a set of measurement data. While a simplex spline model may contain physical information[4] on the system it is not directly based on the actual physical laws that determine the dynamics of the

---

[4] In Sec. 3.5 a method is presented to extract physical information from simplex spline models using a global (i.e. non-barycentric) formulation of the simplex spline polynomials.

modeled system. This lack of a physical basis can result in divergent behavior of the spline polynomials outside the domain of the dataset used to create the simplex spline model.

The polynomial divergence is caused by unbounded behavior of the directional derivatives of the simplex polynomials outside the dataset domain. Essentially, the simplex polynomials attempt to approximate the data generating function as well as its directional derivatives as closely as possible inside the domain of the dataset. Outside the dataset domain, however, the directional derivatives are no longer constrained by any data and therefore remain unchanged. If the directional derivatives of the data generating function have large values near the bounds of the dataset domain, the directional derivatives of the spline polynomials will be equally large in those areas. This causes the simplex polynomial to diverge proportionally. It is important to note that the divergence of the simplex polynomials takes place on a per-simplex level, that is, the divergence is present even on a single simplex with incomplete data coverage.

The differential constraints introduced in this section provide a powerful method for limiting polynomial divergence beyond the boundaries of the data domain by constraining the directional derivatives of the simplex polynomials. This effectively makes the differential constraints a very promising tool for bounded model extrapolation which aims to increase the VER's of simplex spline models. In the following, an expression will be derived for the polynomial divergence at the boundary of the data domain given some data generating function f.

In [104] Lai and Schumaker derived a bound for how well a smooth function f can be approximated by a polynomial p_f of degree d. In this thesis, this expression was generalized to the n-dimensional case in Eq. 3.55 as follows:

$$\max |D_{x_0}^{\alpha_0} D_{x_1}^{\alpha_1} \cdots D_{x_n}^{\alpha_n}(f - p_f)|_\Omega \leq K \cdot |\Omega|^{d+1-\sum \alpha_i} \cdot |f|_{d+1,\Omega}, \;\; 0 \leq \sum \alpha_i \leq d.$$

This generalization will be used in two theorems that quantify the polynomial divergence given a data generating function f. Before introducing the theorems, a clear definition is required for when a B-form polynomial is interpolating, and when it is extrapolating. For this, a new domain related to the domain Ω in Eq. 3.55 must be defined. This new domain is ω which is the smallest n-dimensional sub-simplex contained by the n-simplex t which contains all the data Ξ_t present in t, see Figure 3.20. The sub-simplex ω is defined as follows:

$$\omega \subseteq t, \;\; \Xi_t \in \omega \tag{3.118}$$

The seminorm $|\omega|$ is defined as follows:

$$|\omega| := \max_{\mathbf{y},\mathbf{y} \in \omega} |\mathbf{y} - \mathbf{z}|. \tag{3.119}$$

Using Eq. 3.118 the following behavior can now be defined for the B-form polynomial

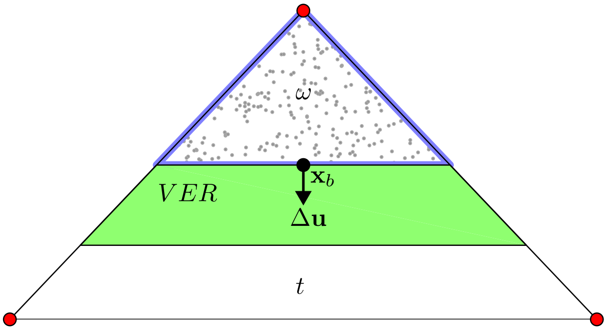


Figure 3.20: The simplex t together with the sub-simplex ω containing all the data. The valid extrapolation region (VER) is the shaded area, with $\Delta\mathbf{u}$ the increment in the direction $\mathbf{u}$ from the point $\mathbf{x}_b$ on the edge of ω.

p:

$$p(b(\mathbf{x})) \longrightarrow \begin{cases} \mathbf{x} \in \omega : \text{interpolating} \\ \mathbf{x} \notin \omega : \text{extrapolating} \end{cases} \tag{3.120}$$

with $b(\mathbf{x})$ the barycentric coordinates with respect to the n-simplex t of the Cartesian point $\mathbf{x} \in \mathbb{R}^n$.

Finally, the polynomial divergence is defined implicitly in the form of the divergence function ψ as follows:

$$\psi(\mathbf{x}_b, \Delta\mathbf{u}) := p(b(\mathbf{x}_b + \Delta\mathbf{u})) - p(b(\mathbf{x}_b)), \quad (\mathbf{x}_b + \Delta\mathbf{u}) \notin \omega \tag{3.121}$$

with $\mathbf{x}_b$ a point located on the convex hull of ω and with $\Delta\mathbf{u}$ an increment in the direction $\mathbf{u}$, see Figure 3.20 for a simple example.

All definitions are now in place for the introduction of the theorems on polynomial divergence of B-form polynomials inside simplices due to incomplete data coverage. In the first theorem, an explicit definition of ψ will be given. In the second theorem, minimum bounds will be derived for the directional derivatives of a B-form polynomial approximating a data generating function on the sub-simplex ω. In the third theorem, these minimum bounds will be used to derive an expression for the lower bound of the divergence function ψ.

Theorem 10. *If p is a B-form polynomial of degree d, then the divergence function ψ from Eq. 3.121 has the following explicit definition:*

$$\psi(\mathbf{x}_b, \Delta\mathbf{u}) = \sum_{m=1}^{d} \left\{ \frac{D^m_{\mathbf{u}} p(b(\mathbf{x}_b))}{m!} \cdot (\Delta\mathbf{u})^m \right\} \tag{3.122}$$

with $\mathbf{x}_b \in \omega$ *a point on the convex hull of* ω *and with* $(\mathbf{x}_b + \Delta\mathbf{u}) \notin \omega$.

Proof. The proof makes use of the Taylor series expansion of p centered at the location $\mathbf{x}_b$ in the direction $\mathbf{u}$ as follows:

$$\begin{aligned} p(b(\mathbf{x}_b + \Delta\mathbf{u})) = p(b(\mathbf{x}_b)) + D^1_{\mathbf{u}} p(b(\mathbf{x}_b)) \cdot \Delta\mathbf{u} + \frac{D^2_{\mathbf{u}} p(b(\mathbf{x}_b))}{2!} \cdot (\Delta\mathbf{u})^2 + \cdots \\ + \frac{D^d_{\mathbf{u}} p(b(\mathbf{x}_b))}{d!} \cdot (\Delta\mathbf{u})^d. \end{aligned} \tag{3.123}$$

It is worth noting that this Taylor series expansion of p is exact, and thus has no remainder. By reorganizing Eq. 3.123 and collecting the differential terms, we get:

$$p(b(\mathbf{x}_b + \Delta\mathbf{u})) - p(b(\mathbf{x}_b)) = \sum_{m=1}^{d} \left\{ \frac{D^m_{\mathbf{u}} p(b(\mathbf{x}_b))}{m!} \cdot (\Delta\mathbf{u})^m \right\}. \tag{3.124}$$

The proof then follows immediately from the definition of ψ from Eq. 3.121.

□

Theorem 11. *The lower bound for the* m^{th} *order directional derivatives in the direction* $\mathbf{u} \in \mathbb{R}^n$ *of the B-form polynomial* p *of degree* d *approximating the data generating function* f *inside the simplical data domain* ω *are given by:*

$$\max |D^m_{\mathbf{u}} p|_\omega \geq K \cdot |\omega|^{d+1-m} \cdot \max |D^{d+1}_{\mathbf{u}} f|_\omega - \max |D^m_{\mathbf{u}} f|_\omega, \ \ 0 \leq m \leq d \tag{3.125}$$

with ω *the simplical domain from Eq. 3.118,* $|\omega|$ *the diameter of* ω *from Eq. 3.119, and with* K *a constant depending only on* d*.*

Proof. The proof is based on a reformulation of Eq. 3.55. Because we are only interested in polynomial divergence in unary directions, i.e. $\mathbf{u} = \mathbf{x}_0 = \mathbf{x}_1 = \ldots = \mathbf{x}_n$, the expression for the maximum approximation error from Eq. 3.55 can be simplified as follows:

$$\max |D^m_{\mathbf{u}} (f - p)|_\omega \leq K \cdot |\omega|^{d+1-m} \cdot |f|_{d+1,\omega}, \ \ 0 \leq m \leq d, \tag{3.126}$$

with $\sum \alpha_i = m$. The linear relation between f and p in Eq. 3.126 allows the following simplification:

$$\max |D^m_{\mathbf{u}} f - D^m_{\mathbf{u}} p|_\omega \leq K \cdot |\omega|^{d+1-m} \cdot |f|_{d+1,\omega}, \ \ 0 \leq m \leq d. \tag{3.127}$$

Now notice that

$$\max|D_{\mathbf{u}}^m f|_\omega - \max|D_{\mathbf{u}}^m p|_\omega \leq \max|D_{\mathbf{u}}^m f - D_{\mathbf{u}}^m p|_\omega. \tag{3.128}$$

The term left of the inequality can be substituted for the term left of the inequality in Eq. 3.128 at the price of a more conservative expression for the bound:

$$\max|D_{\mathbf{u}}^m f|_\omega - \max|D_{\mathbf{u}}^m p|_\omega \leq K \cdot |\omega|^{d+1-m} \cdot |f|_{d+1,\omega}, \;\; 0 \leq m \leq d. \tag{3.129}$$

which after reorganization and substitution of the definition of $|f|_{d+1,\Omega}$ from Eq. 3.52 immediately results in Eq. 3.125.

$\square$

Using **Theorem** 10 and **Theorem** 11 a lower bound can now be given for the polynomial divergence of p for $\mathbf{x}_b$ outside ω given a data generating function f.

Theorem 12. *A lower bound for the polynomial divergence of the B-form polynomial p in the direction $\mathbf{u}$ for $(\mathbf{x}_b + \Delta\mathbf{u}) \notin \omega$ is:*

$$\max_{\mathbf{x}_b\in\omega}|\psi(\mathbf{x}_b,\Delta\mathbf{u})| \geq \max_{\mathbf{x}_b\in\omega}\left|\sum_{m=1}^{d}\left\{\left(K\cdot|\omega|^{d+1-m}\cdot\max|D_{\mathbf{u}}^{d+1}f|_\omega - \max|D_{\mathbf{u}}^m f|_\omega\right)\cdot\frac{\Delta\mathbf{u}}{m!}\right\}\right|. \tag{3.130}$$

Proof. The proof is based on the combination of the results from **Theorem** 10 and **Theorem** 11. First, the maximum value for ψ can be found using Eq. 3.122 as follows:

$$\max_{\mathbf{x}_b\in\omega}|\psi(\mathbf{x}_b,\Delta\mathbf{u})| = \max_{\mathbf{x}_b\in\omega}\left|\sum_{m=1}^{d}\left\{\frac{D_{\mathbf{u}}^m p(b(\mathbf{x}_b))}{m!}\cdot(\Delta\mathbf{u})^m\right\}\right|. \tag{3.131}$$

Now notice that the right hand term in Eq. 3.131 is actually equal to:

$$\max_{\mathbf{x}_b\in\omega}\left|\sum_{m=1}^{d}\left\{\frac{D_{\mathbf{u}}^m p(b(\mathbf{x}_b))}{m!}\cdot(\Delta\mathbf{u})^m\right\}\right| = \max_{\mathbf{x}_b\in\omega}\left|\sum_{m=1}^{d}\left\{\frac{\max|D_{\mathbf{u}}^m p(b(\mathbf{x}_b))|}{m!}\cdot(\Delta\mathbf{u})^m\right\}\right|. \tag{3.132}$$

Using Eq. 3.125 we find that

$$\max_{\mathbf{x}_b\in\omega}\left|\sum_{m=1}^{d}\left\{\frac{\max|D_{\mathbf{u}}^m p(b(\mathbf{x}_b))|}{m!}\cdot(\Delta\mathbf{u})^m\right\}\right| \geq$$

$$\max_{\mathbf{x}_b\in\omega}\left|\sum_{m=1}^{d}\left\{\left(K\cdot|\omega|^{d+1-m}\cdot\max|D_{\mathbf{u}}^{d+1}f|_\omega - \max|D_{\mathbf{u}}^m f|_\omega\right)\cdot\frac{(\Delta\mathbf{u})^m}{m!}\right\}\right|. \tag{3.133}$$

Substitution of Eq. 3.133 into Eq. 3.131 then proves the theorem.

□

Based on the results from the theorems, a full explanation for the polynomial divergence limiting effect of the differential constraints can be offered. By imposing m^{th} order differential constraints with value 0 at N_δ locations on the boundaries of ω, the m^{th} term in the summation for ψ in Eq. 3.122 is effectively made constant. The result is that the lower bound for $\max|\psi|$ from Eq. 3.130 is reduced because the m^{th} term in the summation is equal to 0.

The use of differential constraints for bounded model extrapolation is demonstrated with a numerical experiment. In the experiment, the approximation performance of an ordinary 6^{th} degree spline function with C^2 continuity is compared to a differentially constrained spline function of the same degree and continuity order. The spline functions are both defined on a triangulation consisting of 4 triangles. The dataset used in the experiment is $\mathcal{X}_D$, which only partly covers the triangulation, see Figure 3.21. The green line in Figure 3.21 is the boundary of the data domain, while the magenta line in Figure 3.21 is the boundary of the spline domain on simplex t_4. The data values at $\mathcal{X}_D$ are generated using Franke's function which is shown in the right hand plot of Figure 3.21. In [58] Franke's function is defined as follows:

$$\begin{aligned} f(x_0, x_1) \quad = \quad & \frac{3}{4}exp\left(-\frac{(9x_0-2)^2+(9x_1-2)^2}{4}\right)+ \\ & \frac{3}{4}exp\left(-\left(\frac{(9x_0+1)^2}{49}-\frac{9x_1+1}{10}\right)\right)+ \\ & \frac{1}{2}exp\left(-\frac{(9x_0-7)^2+(9x_1-3)^2}{4}\right)- \\ & \frac{1}{5}exp\left(-\left((9x_0-4)^2-(9x_1-7)^2\right)\right) \qquad (3.134) \end{aligned}$$

Franke's function is a sum of exponential functions, and thus has an infinite number of derivatives. Furthermore, the maximum values for the derivatives tend to increase in magnitude, as the order of the polynomial terms in the derivatives of $f(x_0, x_1)$ increases with the derivative order. Based on Eq. 3.130, it is now clear that the divergence of a B-form polynomial p approximating Franke's function increases with increasing polynomial degree because the lower bound increases with increasing d.

Two bivariate 6^{th} degree simplex spline functions s_1 and s_2 of continuity order C^2 are then used to approximate $f(\mathcal{X}_D)$ using the dataset $\mathcal{X}_D$. The spline function s_1 is only constrained by smoothness constraints, while the second spline function s_2 is constrained by smoothness constraints and differential constraints. In total, s_2 is constrained by four sets of second order (i.e. $m = 2$) differential constraints in the x_1 direction, such that $\mathbf{u} = (0, 1)$, see the left hand plot in Figure 3.21. Three sets of constraints are positioned on the boundary

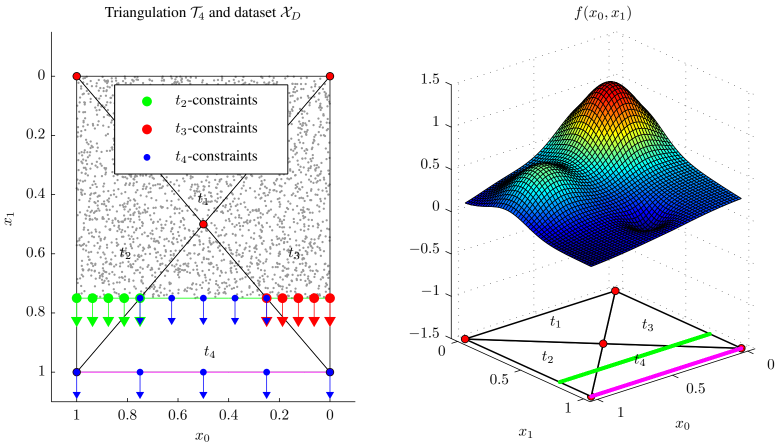


Figure 3.21: The dataset, triangulation and differential constraint locations (left) used for demonstrating the effects of differential constraints. The data values (right) are generated with Franke's function $f(x_0, x_1)$.

of the data domain (the green line in Figure 3.21) such that every one of the three simplices on this edge has $N_\delta = 5$ constraint equations associated with it. One set of constraints is positioned on the edge of the spline domain (the magenta line in Figure 3.21). In all, the simplex polynomials of s_2 on t_2 and t_3 are constrained by 5 differential constraint equations while the simplex polynomial on t_4 is constrained by a total of 10 differential constraints.

The resulting spline functions s_1 and s_2 are plotted in Figure 3.22. The spline function s_1 in the left hand plot of Figure 3.22 diverges rapidly beyond the bounds of the domain of the dataset. This is as expected because the spline function effectively extrapolates the data beyond these bounds. The differentially constrained spline function s_2, plotted in the right hand plot of Figure 3.22, does not show significant divergence. The differential constraints have greatly reduced polynomial divergence, resulting in a bounded model extrapolation of s_2.

In Figure 3.23 the second order derivatives of s_1 and s_2 in the x_1 direction are shown. This figure clearly demonstrates the effects of the differential constraints on the directional derivatives of a simplex spline function. The left hand plot of Figure 3.23 shows that $\frac{\delta^2 s_1}{(\delta x_1)^2}$ diverges uncontrollably beyond the bounds of the data domain. In contrast, the right hand plot shows that $\frac{\delta^2 s_2}{(\delta x_1)^2} = 0$ at the data domain bounds and the edge of the spline domain.

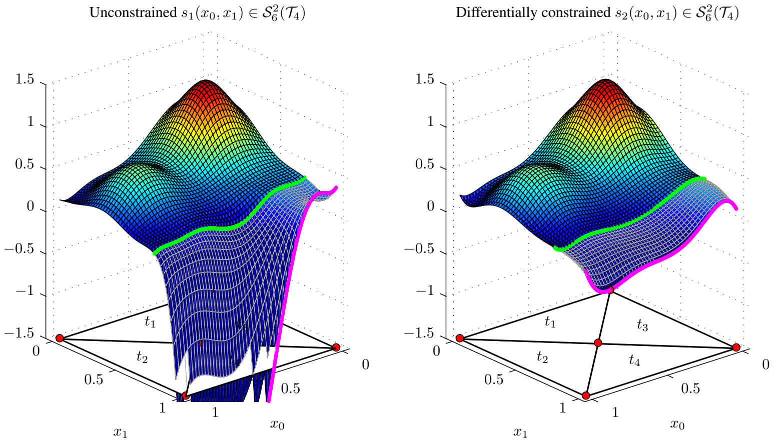


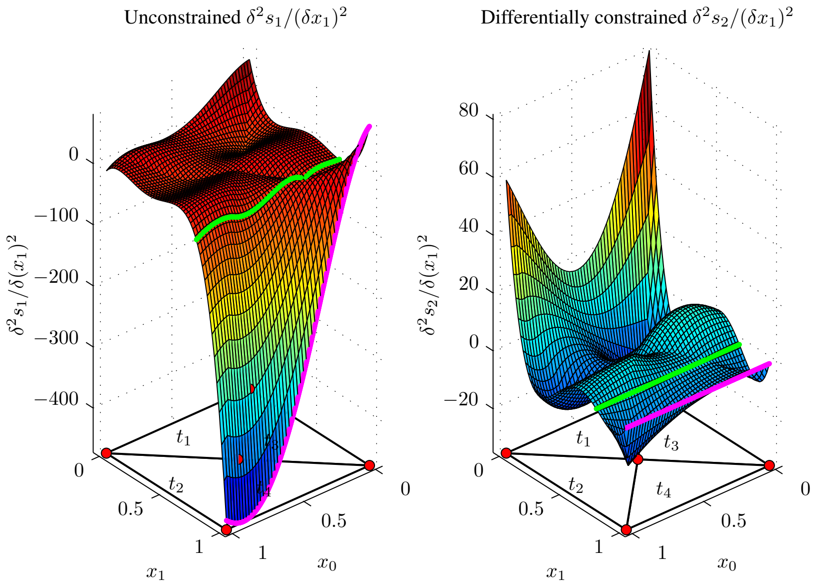
Figure 3.22: Two spline functions of degree 6 and continuity order C^2 in a least squares fit to the identification dataset $(\mathcal{X}_D, f(\mathcal{X}_D))$. The spline function in the left hand plot does not use differential constraints, while the spline function in the right hand plot is differentially constrained. The gray grid shows where the model is extrapolating the data.

3.5 A B-form in Global Coordinates

The barycentric coordinate system is a coordinate system defined in terms of a weighted vertex sum of the vertices of an n-simplex t. The barycentric coordinate system is instrumental in the definition of the stable local basis of the simplex spline basis polynomials. Because the barycentric coordinate system is a local coordinate system, defined on a per-simplex basis, the polynomials defined in terms of these coordinates have a local interpretation only. In many cases, it is desirable to obtain a global physical interpretation of the simplex polynomials, for instance when defining physical constraints on the B-coefficients. Such a physical interpretation requires a reformulation of the simplex polynomials in terms of global physical coordinates. Additionally, the global formulation of the B-form enables the creation of a combined triangulation and B-coefficient optimization algorithm. Combined with a global optimization method like interval analysis [137, 69, 210, 209], the global formulation of the B-form could finally solve the triangulation optimization problem.

In this section, a new formulation of the B-form multivariate simplex splines in terms of global coordinates will be introduced. It will be shown how this new formulation should be used to derive a global interpretation of the simplex spline polynomials.

The section starts with a review of the barycentric coordinate transformation in Sec. 3.5.1. This leads to the formulation of the B-form in global coordinates in Sec. 3.5.2. Finally, in Sec. 3.5.3 an example is given of the use of the B-form in global coordinates for



Unconstrained $\delta^2 s_1/(\delta x_1)^2$

Differentially constrained $\delta^2 s_2/(\delta x_1)^2$

Figure 3.23: The second order directional derivatives of the unconstrained and constrained spline functions s_1 and s_2 in the x_1 direction. The unconstrained spline function is shown in the left plot, while the differentially constrained spline function is shown in the right plot.

the physical interpretation of polynomials. It will be demonstrated that a B-form polynomial of degree d uniquely approximates any function, thereby negating the need for the selection of a specific model structure.

3.5.1 Barycentric coordinates revisited

Before starting the derivation of the B-form in global coordinates, a better insight into the barycentric coordinate system is required. It was shown in Sec. 2.2.2 that the global coordinate $\mathbf{x} = (x_0, x_1, \ldots, x_n) \in \mathbb{R}^n$ can be written as follows:

$$\mathbf{x} = \sum_{i=0}^{n} b_i \mathbf{v}_i, \tag{3.135}$$

with $\mathbf{v}_i$ the i-th vertex of the n-simplex t and b_i the i-th barycentric component. This expression can be rewritten in matrix formulation as follows:

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots \mathbf{v}_n \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} && (3.136) \\ &= \mathbf{V} \cdot \mathbf{b}, && (3.137)\end{aligned}$$

with $\mathbf{V} \in \mathbb{R}^{n \times n+1}$ a singular matrix having the $n+1$ vertices of t as columns. The singular nature of $\mathbf{V}$ shows that the system is under determined, or in other words, contains dependent variables. This is where the barycentric normalization property from Eq. 2.35 is introduced:

$$b_0 = 1 - \sum_{i=1}^{n} b_i. \tag{3.138}$$

Substitution of Eq. 3.138 in Eq. 3.137 results in:

$$\mathbf{x} = \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots \mathbf{v}_n \end{bmatrix} \begin{bmatrix} 1 - \sum_{i=1}^{n} b_i \\ b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

Expanding Eq. 3.139, leads to the following expression:

$$\mathbf{x} = \begin{bmatrix} (\mathbf{v}_1 - \mathbf{v}_0) & (\mathbf{v}_2 - \mathbf{v}_0) & \cdots (\mathbf{v}_n - \mathbf{v}_0) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} + \mathbf{v}_0. \tag{3.139}$$

So finally,

$$\begin{aligned}\mathbf{x} - \mathbf{v}_0 &= \begin{bmatrix} (\mathbf{v}_1 - \mathbf{v}_0) & (\mathbf{v}_2 - \mathbf{v}_0) & \cdots (\mathbf{v}_n - \mathbf{v}_0) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \\ &= \tilde{\mathbf{V}} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, && (3.140)\end{aligned}$$

with the matrix $\tilde{\mathbf{V}}$ defined as:

$$\tilde{\mathbf{V}} = \left[\begin{array}{ccc} (\mathbf{v}_1 - \mathbf{v}_0) & (\mathbf{v}_2 - \mathbf{v}_0) & \cdots (\mathbf{v}_n - \mathbf{v}_0) \end{array} \right] \in \mathbb{R}^{n \times n}. \tag{3.141}$$

It is now easy to see that the matrix $\tilde{\mathbf{V}}$ is invertible when the n-simplex has $n+1$ unique vertices. Define $\mathbf{\Lambda}$ as the inverse of $\tilde{\mathbf{V}}$:

$$\mathbf{\Lambda} = \tilde{\mathbf{V}}^{-1} \in \mathbb{R}^{n \times n}. \tag{3.142}$$

The barycentric components $(b_1, b_2, \ldots, b_n)$ of $\mathbf{x}$ with respect to the simplex t then are:

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \mathbf{\Lambda}(\mathbf{x} - \mathbf{v}_0) \tag{3.143}$$

Now let $\mathbf{z} = (z_1, z_2, \ldots, z_n) \in \mathbb{R}^n$ be the relative coordinate of $\mathbf{x}$ with respect to t as follows:

$$\mathbf{z} = \mathbf{x} - \mathbf{v}_0 \in \mathbb{R}^n. \tag{3.144}$$

With the relative coordinate, Eq. 3.143 can be simplified as follows:

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \mathbf{\Lambda}\mathbf{z} \tag{3.145}$$

Using the normalization property of the barycentric coordinates from Eq. 3.138, the b_0 component becomes:

$$b_0 = 1 - |\mathbf{\Lambda}\mathbf{z}|, \tag{3.146}$$

with $|\bullet|$ the 1-norm of a vector.

At this point the representation in global coordinates of the barycentric coordinate $\mathbf{b} = (b_0, b_1, \ldots, b_n)$ is completely determined, with Eq. 3.145 providing an expression for $(b_1, b_2, \ldots, b_n)$ and with Eq. 3.146 providing an expression for b_0.

3.5.2 A B-form in global coordinates

In Sec. 2.2.5 the vector formulation of the B-form was introduced. This formulation simplifies the definition of the global mapping operator. In Eq. 2.66 the vector formulation of the B-form of polynomial degree d for a single n-simplex t was found to be:

$$p^d(\mathbf{b}) = \mathbf{B}^d(\mathbf{b}) \cdot \mathbf{c}^t.$$

Now, let the equivalent vector form of the B-form in terms of the global coordinate $\mathbf{z}$ from Eq. 3.144 be defined as follows:

$$p^d(\mathbf{z}) = \mathbf{Z}^d(\mathbf{z}) \cdot \mathbf{c}^t, \tag{3.147}$$

with $\mathbf{Z}^d(\mathbf{z}) \in \mathbb{R}^{1\times\hat{d}}$ the vector of basis polynomials which is constructed as follows:

$$\mathbf{Z}^d(\mathbf{z}) = [Z^d_\kappa(\mathbf{z})]_{|\kappa|=d}, \tag{3.148}$$

with $Z^d_\kappa(\mathbf{z})$ the individual basis polynomials in multi-index notation.

At this point, the actual elements of $\mathbf{Z}^d(\mathbf{z})$ are unknown. In the following two theorems, expressions will be derived for the actual basis polynomials in terms of global coordinate.

Theorem 13. *Any B-form polynomial of degree $d = 1$ on the n-simplex t has the following representation in the global coordinate $\mathbf{z}$ from Eq. 3.144:*

$$p^1(\mathbf{z}) = \mathbf{Z}^1(\mathbf{z}) \cdot \mathbf{c}^t, \tag{3.149}$$

with $\mathbf{Z}^1(\mathbf{z}) \in \mathbb{R}^{1\times n+1}$ the vector of Bernstein basis polynomials from Eq. 3.148 for $d = 1$. The vector of first degree Bernstein basis polynomials in the global coordinate $\mathbf{z}$ is then given by:

$$\mathbf{Z}^1(\mathbf{z}) = \begin{bmatrix} 1 - \Lambda_1 z_1 - \Lambda_2 z_2 - \cdots - \Lambda_n z_n \\ \Lambda_{1,1} z_1 + \Lambda_{1,2} z_2 + \cdots + \Lambda_{1,n} z_n \\ \Lambda_{2,1} z_1 + \Lambda_{2,2} z_2 + \cdots + \Lambda_{2,n} z_n \\ \vdots \\ \Lambda_{n,1} z_1 + \Lambda_{n,2} z_2 + \cdots + \Lambda_{n,n} z_n \end{bmatrix}, \tag{3.150}$$

with $\Lambda_{i,j}$ the individual components of the matrix $\mathbf{\Lambda}$ from Eq. 3.142 and with Λ_j the sum of all n elements of the j^{th} column of $\mathbf{\Lambda}$ as follows:

$$\Lambda_j := \sum_{i=1}^{n} \Lambda_{i,j} \tag{3.151}$$

Proof. Expanding the vector form of the B-form in barycentric coordinates for a first degree simplex polynomial leads to:

$$\begin{aligned} p^1(\mathbf{b}) &= [b^{\kappa}]_{|\kappa|=1} \cdot \mathbf{c}^t \\ &= [b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n}]_{|\kappa|=1} \cdot \mathbf{c}^t \\ &= \begin{bmatrix} b_0 & b_1 & \cdots & b_n \end{bmatrix} \cdot \mathbf{c}^t \end{aligned} \tag{3.152}$$

In Eq. 3.145 it was found that the barycentric coordinate components $(b_1, b_2, \ldots, b_n)$ are given by:

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \mathbf{\Lambda z}.$$

with Λ the inverted normalized vertex matrix from Eq. 3.142.

Expanding the matrix multiplication from Eq. 3.145 results in the following expression for the individual barycentric components b_i with $i > 0$:

$$b_i = \Lambda_{i,1} z_1 + \Lambda_{i,2} z_2 + \cdots + \Lambda_{i,n} z_n, \ \ i > 0, \tag{3.153}$$

Using Eq. 3.146, the b_0 component can be reformulated as follows:

$$\begin{aligned} b_0 &= 1 - |\mathbf{\Lambda z}| \\ &= 1 - ((\Lambda_{1,1} + \Lambda_{2,1} + \cdots + \Lambda_{n,1}) z_1 + (\Lambda_{1,2} + \Lambda_{2,2} + \cdots + \Lambda_{n,2}) z_2 + \cdots + \\ &\quad + (\Lambda_{1,n} + \Lambda_{2,n} + \cdots + \Lambda_{n,n}) z_n) \\ &= 1 - \sum_{j=1}^{n} (\Lambda_{1,j} + \Lambda_{2,j} + \cdots + \Lambda_{n,j}) z_j \end{aligned} \tag{3.154}$$

Using Eq. 3.151, Eq. 3.154 can be simplified as follows:

$$\begin{aligned} b_0 &= 1 - \sum_{j=1}^{n} \Lambda_j z_j \\ &= 1 - \Lambda_1 z_1 - \Lambda_2 z_2 - \cdots - \Lambda_n z_n \end{aligned} \tag{3.155}$$

Substitution of the expression for b_0 from Eq. 3.155 and the expression for $(b_1, b_2, \ldots, b_n)$ from Eq. 3.153 in Eq. 3.152 then immediately results in Eq. 3.150, which proves the theorem.

$\square$

Higher order polynomials in terms of the global coordinate $\mathbf{z}$ can be found easily by recursively multiplying the vector of basis polynomials from Eq. 3.150. Based on **Theorem** 13, the following theorem for the general B-form in global coordinates can be introduced.

Theorem 14. *The B-form polynomial of degree d on the n-simplex t has the following representation in the global coordinate $\mathbf{z}$:*

$$p^d(\mathbf{z}) = \mathbf{Z}^d(\mathbf{z}) \cdot \mathbf{c}^t, \tag{3.156}$$

with $\mathbf{Z}^d(\mathbf{z})$ the vector of basis polynomials of degree d in the global coordinate $\mathbf{z}$ which is defined as follows:

$$\mathbf{Z}^d(\mathbf{z}) := \left[\frac{d!}{\kappa!} \cdot (\mathbf{Z}^1(\mathbf{z}))^{\kappa}\right]_{|\kappa|=d}, \quad \mathbf{Z}^d(\mathbf{z}) \in \mathbb{R}^{1\times\hat{d}}, \tag{3.157}$$

with $\mathbf{Z}^1(\mathbf{z})$ the vector of basis polynomials of degree 1 from Eq. 3.150

Proof. By treating every individual polynomial basis function in $\mathbf{Z}^1_i(\mathbf{z})$ as a single term, the multinomial theorem can be used to create a new polynomial of any degree d:

$$\left(\mathbf{Z}^1_1(\mathbf{z}) + \mathbf{Z}^1_2(\mathbf{z}) + \cdots + \mathbf{Z}^1_{n+1}(\mathbf{z})\right)^d = \sum_{|\kappa|=d} \frac{d!}{\kappa!}(\mathbf{Z}^1(\mathbf{z}))^{\kappa}. \tag{3.158}$$

Using the method for constructing the vector of basis polynomials from Eq. 2.66, the right hand term in Eq. 3.158 can be represented in the vector formulation of Eq. 3.157, thereby proving the theorem.

□

A number of examples of the use of the global B-form will be given below.

Example 18 (The global B-form for a first degree polynomial on a 1-dimensional simplex)**.** *A 1-dimensional simplex is formed by the convex hull of two vertices:*

$$t = \langle(v_0, v_1)\rangle \tag{3.159}$$

In this case we have $n = 1$, $d = 1$ and $\kappa \in \{(1,0),(0,1)\}$. The transformation matrix $\mathbf{\Lambda}$ from Eq. 3.142 is actually a scalar:

$$\mathbf{\Lambda} = \tilde{\mathbf{V}}^{-1} = \frac{1}{v_1 - v_0}, \tag{3.160}$$

which implies that $\mathbf{\Lambda} = \Lambda_1 = \Lambda_{1,1}$. The global coordinate in this case is $\mathbf{z} = z_1 = x - v_0$, *which is a scalar. Using Eq. 3.150 the following vector of basis polynomials is obtained:*

$$
\begin{aligned}
p^1(\mathbf{z}) &= \begin{bmatrix} 1 - \Lambda_{1,1} z_1 \\ \Lambda_{1,1} z_1 \end{bmatrix}^\top \cdot \mathbf{c}^t \\
&= \begin{bmatrix} 1 - \frac{1}{v_1 - v_0} z_1 \\ \frac{1}{v_1 - v_0} z_1 \end{bmatrix}^\top \cdot \mathbf{c}^t
\end{aligned}
$$

Example 19 (The global B-form for a second degree polynomial on a 1-dimensional simplex)**.** *In this case we have $n = 1$, $d = 2$ and $\kappa \in \{(2,0),(1,1),(0,2)\}$. The transformation matrix $\mathbf{\Lambda}$, again a scalar, is the same as in Eq. 3.160. Using Eq. 3.157 together with the results of* ***Example*** *18 results in:*

$$
\begin{aligned}
p^2(\mathbf{z}) &= \begin{bmatrix} \frac{2!}{2!0!}(1 - \Lambda_1 z_1)^2 (\Lambda_{1,1} z_1)^0 \\ \frac{2!}{1!1!}(1 - \Lambda_1 z_1)^1 (\Lambda_{1,1} z_1)^1 \\ \frac{2!}{0!2!}(1 - \Lambda_1 z_1)^0 (\Lambda_{1,1} z_1)^2 \end{bmatrix}^\top \cdot \mathbf{c}^t \\
&= \begin{bmatrix} 1 - 2\Lambda_1 z_1 + \Lambda_1^2 z_1^2 \\ 2\Lambda_{1,1} - 2\Lambda_1 \Lambda_{1,1} z_1^2 \\ \Lambda_{1,1}^2 z_1^2 \end{bmatrix}^\top \cdot \mathbf{c}^t \\
&= \begin{bmatrix} 1 - \frac{2}{v_1 - v_0} z_1 + \frac{1}{(v_1 - v_0)^2} z_1^2 \\ \frac{2}{v_1 - v_0} z_1 - \frac{2}{(v_1 - v_0)^2} z_1^2 \\ \frac{1}{(v_1 - v_0)^2} z_1^2 \end{bmatrix}^\top \cdot \mathbf{c}^t
\end{aligned}
$$

Example 20 (The global B-form of a first degree polynomial on a 2-dimensional simplex)**.** *A 2-dimensional simplex is formed by the convex hull of three vertices:*

$$
t = \langle (\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2) \rangle \tag{3.161}
$$

In this case we have $n = 2$, $d = 1$ and $\kappa \in \{(1,0,0),(0,1,0),(0,0,1)\}$. The transformation matrix $\mathbf{\Lambda}$ from Eq. 3.142 is:

$$
\begin{aligned}
\mathbf{\Lambda} &= \tilde{\mathbf{V}}^{-1} && (3.162) \\
&= \begin{bmatrix} \mathbf{v}_{1_1} - \mathbf{v}_{0_1} & \mathbf{v}_{2_1} - \mathbf{v}_{0_1} \\ \mathbf{v}_{1_2} - \mathbf{v}_{0_2} & \mathbf{v}_{2_2} - \mathbf{v}_{0_2} \end{bmatrix}^{-1}. && (3.163)
\end{aligned}
$$

Using Eq. 3.151 we find for Λ_1 *and* Λ_2*:*

$$\begin{aligned}\Lambda_1 &= (\mathbf{v}_{1_1} - \mathbf{v}_{0_1}) + (\mathbf{v}_{1_2} - \mathbf{v}_{0_2})\\ \Lambda_2 &= (\mathbf{v}_{2_1} - \mathbf{v}_{0_1}) + (\mathbf{v}_{2_2} - \mathbf{v}_{0_2}),\end{aligned}$$

while $\Lambda_{1,1}$*,* $\Lambda_{1,2}$*,* $\Lambda_{2,1}$*, and* $\Lambda_{2,2}$ *are the elements of* Λ *from Eq. 3.163. Combining these results results in the global B-form of degree* 1 *and dimension* 2*:*

$$p(\mathbf{z}) = \begin{bmatrix} 1 - \Lambda_1 z_1 - \Lambda_2 z_2 \\ \Lambda_{1,1} z_1 + \Lambda_{1,2} z_2 \\ \Lambda_{2,1} z_1 + \Lambda_{2,2} z_2 \end{bmatrix}^\top \cdot \mathbf{c}^t$$

Example 21 (The global B-form of a second degree polynomial on a 2-dimensional simplex)**.** *In this example the simplified construction method for B-form polynomials in global coordinates will be demonstrated. In this case, the global B-form polynomial of degree* $d = 2$ *on a single* 2*-simplex will be derived using* ***Theorem*** *14.*

First, we need the first degree basis function vector in global coordinates from Eq. 3.150 for $n = 2$*. In* ***Example*** *20 this basis was found to be:*

$$\mathbf{Z}^1(\mathbf{z}) = \begin{bmatrix} 1 - \Lambda_1 z_1 - \Lambda_2 z_2 \\ \Lambda_{1,1} z_1 + \Lambda_{1,2} z_2 \\ \Lambda_{2,1} z_1 + \Lambda_{2,2} z_2 \end{bmatrix}^\top$$

Using Eq. 3.157, together with the valid values for κ we can derive $\mathbf{Z}^2(\mathbf{z})$:

$$\begin{aligned}
\mathbf{Z}^2(\mathbf{z}) &= \left[\frac{2!}{\kappa!}\cdot(\mathbf{Z}^1(\mathbf{z}))^{\kappa}\right]_{|\kappa|=2}, \quad \mathbf{Z}^2(\mathbf{z}) \in \mathbb{R}^{1\times 6} \\
&= \begin{bmatrix}
\frac{2!}{2!0!0!}\cdot(\mathbf{Z}^1(\mathbf{z}))^{2,0,0} \\
\frac{2!}{1!1!0!}\cdot(\mathbf{Z}^1(\mathbf{z}))^{1,1,0} \\
\frac{2!}{1!0!1!}\cdot(\mathbf{Z}^1(\mathbf{z}))^{1,0,1} \\
\frac{2!}{0!2!0!}\cdot(\mathbf{Z}^1(\mathbf{z}))^{0,2,0} \\
\frac{2!}{0!1!1!}\cdot(\mathbf{Z}^1(\mathbf{z}))^{0,1,1} \\
\frac{2!}{0!0!2!}\cdot(\mathbf{Z}^1(\mathbf{z}))^{0,0,2}
\end{bmatrix}^{\top} \\
&= \begin{bmatrix}
(1-\Lambda_1 z_1-\Lambda_2 z_2)^2 \\
2\left((1-\Lambda_1 z_1-\Lambda_2 z_2)(\Lambda_{1,1}z_1+\Lambda_{1,2}z_2)\right) \\
2\left((1-\Lambda_1 z_1-\Lambda_2 z_2)(\Lambda_{2,1}z_1+\Lambda_{2,2}z_2)\right) \\
(\Lambda_{1,1}z_1+\Lambda_{1,2}z_2)^2 \\
2\left((\Lambda_{1,1}z_1+\Lambda_{1,2}z_2)(\Lambda_{2,1}z_1+\Lambda_{2,2}z_2)\right) \\
(\Lambda_{2,1}z_1+\Lambda_{2,2}z_2)^2
\end{bmatrix}^{\top} \\
&= \begin{bmatrix}
1-2\Lambda_1 z_1-2\Lambda_2 z_2+2\Lambda_1\Lambda_2 z_1 z_2+\Lambda_1^2 z_1^2+\Lambda_2^2 z_2^2 \\
2\Lambda_{1,1}z_1+2\Lambda_{1,2}z_2-2\Lambda_1\Lambda_{1,2}z_1z_2-2\Lambda_{1,1}\Lambda_2 z_1z_2-2\Lambda_1\Lambda_{1,1}z_1^2-2\Lambda_{1,2}\Lambda_2 z_2^2 \\
2\Lambda_{2,1}z_1+2\Lambda_{2,2}z_2-2\Lambda_1\Lambda_{2,2}z_1z_2-2\Lambda_2\Lambda_{2,1}z_1z_2-2\Lambda_1\Lambda_{2,1}z_1^2-2\Lambda_2\Lambda_{2,2}z_2^2 \\
2\Lambda_{1,1}\Lambda_{1,2}z_1z_2+\Lambda_{1,1}^2 z_1^2+\Lambda_{1,2}^2 z_2^2 \\
2\Lambda_{1,1}\Lambda_{2,2}z_1z_2+2\Lambda_{1,2}\Lambda_{2,1}z_1z_2+2\Lambda_{1,1}\Lambda_{2,1}z_1^2+2\Lambda_{1,2}\Lambda_{2,2}z_2^2 \\
2\Lambda_{2,1}\Lambda_{2,2}z_1z_2+\Lambda_{2,1}^2 z_1^2+\Lambda_{2,2}^2 z_2^2
\end{bmatrix}^{\top}
\end{aligned}$$

3.5.3 Globally interpreting B-form polynomials

In the previous section, a new formulation of the B-form of the multivariate simplex spline was introduced. This new formulation uses global coordinates instead of local barycentric coordinates. The advantage of this approach is that the resulting basis polynomials have an actual meaning in a physical context. In this section, an example will be given of the use of the global B-form formulation for physically interpreting simplex polynomials.

First, define α and $\bar{q}$ as the physical variables in which the B-form polynomial will be expressed. Then, let $\mathcal{X}_D$ be a scattered dataset as shown in Figure 3.24. Let the data values be generated by the bivariate quadratic function $C_m(\alpha, q)$ as follows:

$$C_m(\alpha, q) = C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\alpha^2}}\alpha^2 + C_{m_q}\bar{q}, \tag{3.164}$$

where the values of the parameters are $C_{m_0} = 0.6$, $C_{m_\alpha} = -0.5$, $C_{m_{\alpha^2}} = -5.2$, and $C_{m_q} = -7.0$.

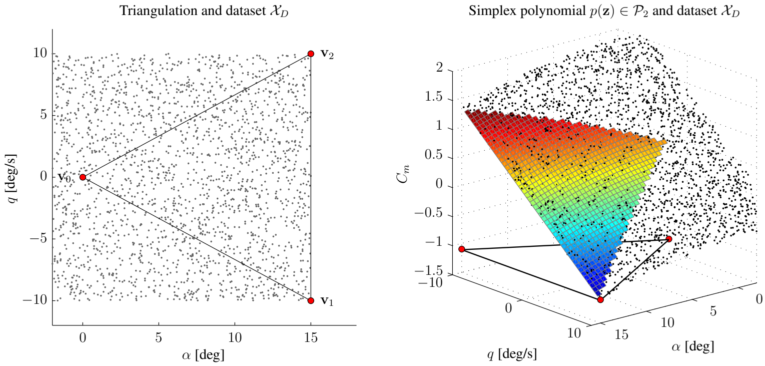


Figure 3.24: The dataset $\mathcal{X}_D$, the triangulation consisting of a single simplex, and the simplex polynomial that best fits $C_m(\alpha, \bar{q})$.

A simplex is now created inside the domain of $\mathbf{X}_D$:

$$t = \langle(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2)\rangle\,, \tag{3.165}$$

where the vertices are assumed to have the following values:

$$\mathbf{v}_0 = (0,0), \quad \mathbf{v}_1 = (15\frac{180}{\pi}, -10\frac{180}{\pi}), \quad \mathbf{v}_2 = (15\frac{180}{\pi}, 10\frac{180}{\pi}). \tag{3.166}$$

In this case, the relative global coordinate $\mathbf{z}$ is given by:

$$\begin{aligned}\mathbf{z} &= \begin{bmatrix} \alpha \\ \bar{q} \end{bmatrix} - \mathbf{v}_0 \\ &= \begin{bmatrix} \alpha \\ \bar{q} \end{bmatrix}\end{aligned}$$

Using Eq. 3.156 and $\mathbf{z} = (\alpha, \bar{q})$, the function $C_m(\alpha, q)$ is then approximated with a quadratic simplex polynomial in global coordinates $p \in \mathcal{P}_2$ as follows

$$p^2(\alpha, \bar{q}) = \mathbf{Z}^2(\alpha, \bar{q}) \cdot \mathbf{c}^t.$$

The vector of bivariate, second degree basis polynomials was already derived in **Exam-**

ple 21, and will be used here with $\mathbf{z} = (\alpha, \bar{q})$ as follows:

$$\mathbf{Z}^2(\alpha,\bar{q}) = \begin{bmatrix} 1 - 2\Lambda_1\alpha - 2\Lambda_2\bar{q} + 2\Lambda_1\Lambda_2\alpha\bar{q} + \Lambda_1^2\alpha^2 + \Lambda_2^2\bar{q}^2 \\ 2\Lambda_{1,1}\alpha + 2\Lambda_{1,2}\bar{q} - 2\Lambda_1\Lambda_{1,2}\alpha\bar{q} - 2\Lambda_{1,1}\Lambda_2\alpha\bar{q} - 2\Lambda_1\Lambda_{1,1}\alpha^2 - 2\Lambda_{1,2}\Lambda_2\bar{q}^2 \\ 2\Lambda_{2,1}\alpha + 2\Lambda_{2,2}\bar{q} - 2\Lambda_1\Lambda_{2,2}\alpha\bar{q} - 2\Lambda_2\Lambda_{2,1}\alpha\bar{q} - 2\Lambda_1\Lambda_{2,1}\alpha^2 - 2\Lambda_2\Lambda_{2,2}\bar{q}^2 \\ 2\Lambda_{1,1}\Lambda_{1,2}\alpha\bar{q} + \Lambda_{1,1}^2\alpha^2 + \Lambda_{1,2}^2\bar{q}^2 \\ 2\Lambda_{1,1}\Lambda_{2,2}\alpha\bar{q} + 2\Lambda_{1,2}\Lambda_{2,1}\alpha\bar{q} + 2\Lambda_{1,1}\Lambda_{2,1}\alpha^2 + 2\Lambda_{1,2}\Lambda_{2,2}\bar{q}^2 \\ 2\Lambda_{2,1}\Lambda_{2,2}\alpha\bar{q} + \Lambda_{2,1}^2\alpha^2 + \Lambda_{2,2}^2\bar{q}^2 \end{bmatrix}^\top .$$

After explicitly calculating the values of $\mathbf{\Lambda}$ this results in:

$$\mathbf{Z}^2(\alpha,\bar{q}) = \begin{bmatrix} \frac{144\,\alpha^2}{\pi^2} - \frac{24\,\alpha}{\pi} + 1 \\ \frac{12\,\alpha}{\pi} - \frac{18\,\bar{q}}{\pi} - \frac{144\,\alpha^2}{\pi^2} + \frac{216\,\alpha\,\bar{q}}{\pi^2} \\ \frac{12\,\alpha}{\pi} + \frac{18\,\bar{q}}{\pi} - \frac{144\,\alpha^2}{\pi^2} - \frac{216\,\alpha\,\bar{q}}{\pi^2} \\ \frac{36\,\alpha^2}{\pi^2} - \frac{108\,\alpha\,\bar{q}}{\pi^2} + \frac{81\,\bar{q}^2}{\pi^2} \\ \frac{72\,\alpha^2}{\pi^2} - \frac{162\,\bar{q}^2}{\pi^2} \\ \frac{36\,\alpha^2}{\pi^2} + \frac{108\,\alpha\,\bar{q}}{\pi^2} + \frac{81\,\bar{q}^2}{\pi^2} \end{bmatrix}^\top .$$

Using a least squares estimator, the vector of B-coefficients is found to be:

$$\mathbf{c}^t = \begin{bmatrix} 0.6000 & 1.1454 & -0.0763 & 1.3344 & 0.1127 & -1.1090 \end{bmatrix}^\top . \qquad (3.167)$$

The complete B-form polynomial in global coordinates then becomes:

$$\begin{aligned} p^2(\alpha,\bar{q}) &= \mathbf{Z}^2(\alpha,\bar{q}) \cdot \mathbf{c}^t \\ &= \begin{bmatrix} \frac{144\,\alpha^2}{\pi^2} - \frac{24\,\alpha}{\pi} + 1 \\ \frac{12\,\alpha}{\pi} - \frac{18\,\bar{q}}{\pi} - \frac{144\,\alpha^2}{\pi^2} + \frac{216\,\alpha\,\bar{q}}{\pi^2} \\ \frac{12\,\alpha}{\pi} + \frac{18\,\bar{q}}{\pi} - \frac{144\,\alpha^2}{\pi^2} - \frac{216\,\alpha\,\bar{q}}{\pi^2} \\ \frac{36\,\alpha^2}{\pi^2} - \frac{108\,\alpha\,\bar{q}}{\pi^2} + \frac{81\,\bar{q}^2}{\pi^2} \\ \frac{72\,\alpha^2}{\pi^2} - \frac{162\,\bar{q}^2}{\pi^2} \\ \frac{36\,\alpha^2}{\pi^2} + \frac{108\,\alpha\,\bar{q}}{\pi^2} + \frac{81\,\bar{q}^2}{\pi^2} \end{bmatrix}^\top \begin{bmatrix} 0.6000 \\ 1.1454 \\ -0.0763 \\ 1.3344 \\ 0.1127 \\ -1.1090 \end{bmatrix} \\ &= 0.6 - 0.5\alpha - 5.2\alpha^2 - 7.0\bar{q}. \end{aligned}$$

The final result shows that the function $C_m(\alpha,\bar{q})$ has been approximated exactly to within numerical accuracy! Even more profoundly, this shows that any polynomial of degree d is indeed *uniquely* approximated using a B-form polynomial of degree $\geq d$. In fact, it can be shown that a third degree B-form polynomial $p \in \mathcal{P}_3$ produces the exact same results as the second degree polynomial shown above:

$$\begin{aligned}
p^3(\alpha,\bar{q}) &= \mathbf{Z}^3(\alpha,\bar{q})\cdot\mathbf{c}^t \\
&= \begin{bmatrix}
-\frac{1728\,\bar{\alpha}^3}{\pi^3}+\frac{432\,\bar{\alpha}^2}{\pi^2}-\frac{36\,\bar{\alpha}}{\pi}+1 \\
\frac{18\,\bar{\alpha}}{\pi}-\frac{27\,\bar{q}}{\pi}-\frac{432\,\bar{\alpha}^2}{\pi^2}+\frac{2592\,\bar{\alpha}^3}{\pi^3}-\frac{3888\,\bar{\alpha}^2\,\bar{q}}{\pi^3}+\frac{648\,\bar{\alpha}\,\bar{q}}{\pi^2} \\
\frac{18\,\bar{\alpha}}{\pi}+\frac{27\,\bar{q}}{\pi}-\frac{432\,\bar{\alpha}^2}{\pi^2}+\frac{2592\,\bar{\alpha}^3}{\pi^3}+\frac{3888\,\bar{\alpha}^2\,\bar{q}}{\pi^3}-\frac{648\,\bar{\alpha}\,\bar{q}}{\pi^2} \\
-\frac{1296\,\bar{\alpha}^3}{\pi^3}+\frac{3888\,\bar{\alpha}^2\,\bar{q}}{\pi^3}+\frac{108\,\bar{\alpha}^2}{\pi^2}-\frac{2916\,\bar{\alpha}\,\bar{q}^2}{\pi^3}-\frac{324\,\bar{\alpha}\,\bar{q}}{\pi^2}+\frac{243\,\bar{q}^2}{\pi^2} \\
-\frac{2592\,\bar{\alpha}^3}{\pi^3}+\frac{216\,\bar{\alpha}^2}{\pi^2}+\frac{5832\,\bar{\alpha}\,\bar{q}^2}{\pi^3}-\frac{486\,\bar{q}^2}{\pi^2} \\
-\frac{1296\,\bar{\alpha}^3}{\pi^3}-\frac{3888\,\bar{\alpha}^2\,\bar{q}}{\pi^3}+\frac{108\,\bar{\alpha}^2}{\pi^2}-\frac{2916\,\bar{\alpha}\,\bar{q}^2}{\pi^3}+\frac{324\,\bar{\alpha}\,\bar{q}}{\pi^2}+\frac{243\,\bar{q}^2}{\pi^2} \\
\frac{216\,\bar{\alpha}^3}{\pi^3}-\frac{972\,\bar{\alpha}^2\,\bar{q}}{\pi^3}+\frac{1458\,\bar{\alpha}\,\bar{q}^2}{\pi^3}-\frac{729\,\bar{q}^3}{\pi^3} \\
\frac{648\,\bar{\alpha}^3}{\pi^3}-\frac{972\,\bar{\alpha}^2\,\bar{q}}{\pi^3}-\frac{1458\,\bar{\alpha}\,\bar{q}^2}{\pi^3}+\frac{2187\,\bar{q}^3}{\pi^3} \\
\frac{648\,\bar{\alpha}^3}{\pi^3}+\frac{972\,\bar{\alpha}^2\,\bar{q}}{\pi^3}-\frac{1458\,\bar{\alpha}\,\bar{q}^2}{\pi^3}-\frac{2187\,\bar{q}^3}{\pi^3} \\
\frac{216\,\bar{\alpha}^3}{\pi^3}+\frac{972\,\bar{\alpha}^2\,\bar{q}}{\pi^3}+\frac{1458\,\bar{\alpha}\,\bar{q}^2}{\pi^3}+\frac{729\,\bar{q}^3}{\pi^3}
\end{bmatrix}^{\top}
\begin{bmatrix}
0.6000 \\ 0.9636 \\ 0.1491 \\ 1.2084 \\ 0.3939 \\ -0.4206 \\ 1.3344 \\ 0.5199 \\ -0.2945 \\ -1.1090
\end{bmatrix} \\
&= 0.6 - 0.5\alpha - 5.2\alpha^2 - 7.0\bar{q}.
\end{aligned}$$

This example demonstrates that the usage of B-form polynomials negates the need for the selection of a specific polynomial model structure. The only choice in model selection with B-form polynomials is the selection of the maximum polynomial degree. This property of B-form polynomials will prove to be extremely useful when using simplex splines for system identification purposes.

Chapter 4

System Identification with Simplex Splines

In this thesis, a new method for system identification in the time domain based on multivariate simplex splines is presented. The theoretical basis for this method was first introduced in [40] and [41], and is based on a new formulation of the standard linear regression model. The crux of this new formulation is the inclusion of the B-form basis polynomials of the multivariate simplex splines in a linear regression scheme. The new approach to linear regression with multivariate splines results in a powerful new method for parameter estimation and system identification of complex time-variant nonlinear systems. Using the new regression model structure, a number of different equality constrained parameter estimation techniques can be employed in the estimation of the B-coefficients of the simplex splines. In this thesis two such methods for parameter estimation will be introduced. The first is a generalized least squares estimator, which enables the estimation of B-coefficients on simplices containing noise of varying magnitudes. The second parameter estimator is a differentially constrained recursive least squares estimator. The recursive estimator has the potential to revolutionize the field of adaptive model based control as it allows, in real-time, the reconfiguration of spline based models based on incoming observations. The quality of the estimated spline models can be assessed using existing methods based on an analysis of the model residuals and parameter variances. Additionally, a number of completely new quality assessment methods are enabled by the use of the B-form polynomials. For example, because the coefficients of the simplex splines have a spatial location within the spline domain, the variances of the B-coefficients can be pinpointed to specific locations within the model. This means that regions of high parameter variance can be isolated within the global model and subjected to further analysis. These unique and powerful properties together may result in a new perspective on system identification and parameter estimation, potentially leading to further innovations in the field.

In this chapter, the theory of the multivariate simplex spline will be cast into a framework for system identification. First, in Sec. 4.1 the process of model structure selection for simplex splines will be presented. It will be explained that the model structure selection process can be split into two parts, the first of which is the geometric structure selection, and the second the polynomial structure selection. The geometric part together with the polynomial part are then fused together in the new linear regression model for simplex splines. In Sec. 4.2 the new linear regression model is used in the definition of two new parameter estimators for the B-coefficients of the multivariate simplex splines. The first new estimator is a generalized least squares estimator which allows the accurate estimation of B-coefficients and their variances on triangulations with locally varying noise intensities. The second new estimator is a differentially constrained recursive least squares estimator which allows for the real-time estimation of the B-coefficients. This is useful when the simplex splines are used in an adaptive model based control system which must adapt to changing system dynamics. The recursive B-coefficient estimator is enabled by the inclusion of the differential constraints from Chpt. 3. The differential constraints bound the directional derivatives of the model in areas where data is scarce, effectively adding a bounded model extrapolation capability to the simplex splines. The quality of simplex spline models can be assessed using a number of different methods, some of which are unique to the simplex splines. In Sec. 4.3 these quality assessment methods will be introduced. Finally, in Sec. 4.4 the generalized least squares estimator will be demonstrated on a real-life dataset, in this case a part of a NASA wind tunnel dataset of the F-16 fighter aircraft. The various model quality assessment methods will be demonstrated, and it will be shown that these methods indeed provide a new perspective on model quality assessment.

4.1 Model Structure Selection

A multivariate simplex spline function effectively consists of two components; a geometric component and a polynomial component. The geometric component is the triangulation, which provides a geometric basis for the second component, the polynomials that make up the actual spline functions.

Geometric model structure selection is a challenging task, as it presents a nonlinear, non-convex optimization problem for which there is no general solution at the time of this writing. However, it was found that geometric structure selection can be decomposed into a prototype phase and a subsequent refinement phase, resulting in an adequate geometric model structure. In Sec. 4.1.1 this two-phased approach towards geometric model structure

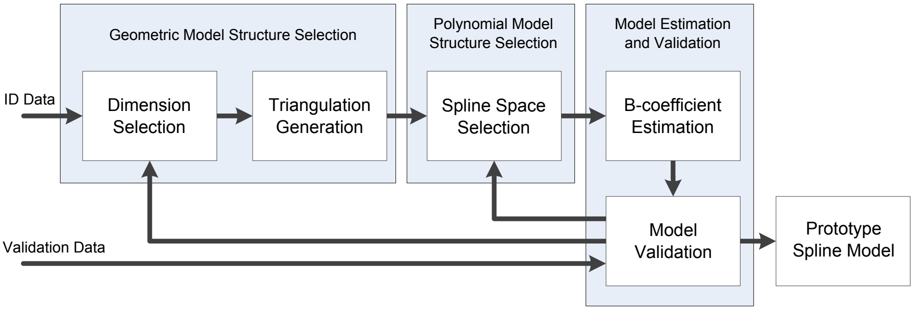


Figure 4.1: An overview of the prototype phase in simplex spline model identification.

selection is discussed. The polynomial model structure selection is more straightforward because the simplex spline polynomials form a basis for all polynomials of a given degree. Polynomial model structure selection therefore boils down to the selection of a single spline space of a certain degree and continuity order, as explained in Sec. 4.1.2. With the geometric and polynomial model structures defined, a linear regression model structure can be formulated which enables the use of well-known and much used parameter estimators like least squares, and maximum likelihood estimators. In Sec. 4.1.3 this linear regression model structure for simplex splines is introduced.

4.1.1 Geometric model structure selection

In their very essence, simplex splines are geometric entities. Their basis functions are defined on geometric structures called simplices, and their parameters have a spatial location inside the simplices that support them. Geometric model structure selection for simplex splines actually consists of two parts. The first part is the selection of the spline dimensions. The second part is the triangulation generation process during which a set of interconnected simplices is defined which in some way optimally span a given dataset. Geometric model structure selection forms a nonlinear and non-convex optimization problem for which there is no general solution at the time of this writing.

In this thesis a two-phase approach towards geometric model structure selection is taken. The first phase is called the ‘prototype phase’, see Figure 4.1. The second phase is the ‘refinement phase’ which is shown in Figure 4.2. The goal of the prototype phase is the selection of the spline model dimension from a given set of candidate dimensions. During this phase, one of the simplest possible triangulations is used in the form of the Type I triangulation of the hypercube, which was discussed in-depth in Sec. 2.3.3. An example of this triangulation for the 3-dimensional case is shown in Figure 4.3. For every dimension candidate a number of different spline spaces are used to model the dataset on the

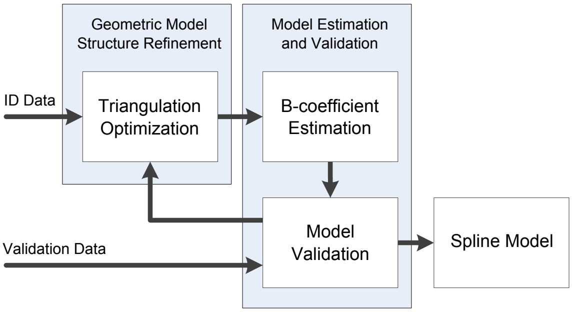


Figure 4.2: An overview of the final model refinement phase in simplex spline model identification.

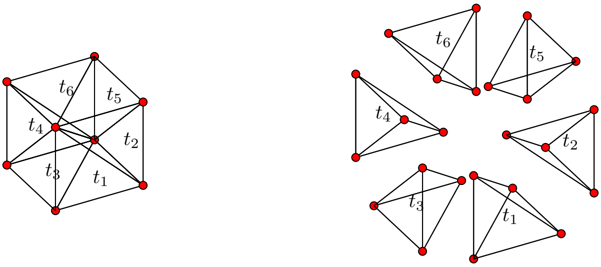


Figure 4.3: A single triangulated 3-cube (left) consisting of six tetrahedrons used during the prototype phase and its exploded view (right).

hypercube triangulation. The set of dimensions that produced the best overall performing spline function is then selected as the dimension set of choice.

The aim of the refinement phase is the creation of an optimal triangulation using the selected dimension candidate from the prototype phase. In Chpt. 2 a method for triangulation optimization was introduced which produces well-defined triangulations by filling the convex hull of the dataset with triangulated hypercubes. It was shown that this method produces high quality triangulations, which may not be globally optimal, but are nevertheless well suited for use with simplex splines. During the refinement phase, this triangulation optimization method will be used to create spline functions of the highest possible quality.

An even more powerful triangulation optimization method could be based on the formulation of the B-form polynomials in global coordinates from Sec. 3.5. Using a global optimization method, like interval analysis [137, 69, 209], this could lead to a triangulation that is guaranteed to be the optimal solution.

4.1.2 Polynomial model structure selection

While the geometric model structure selection procedure is relatively complex, the polynomial model structure selection procedure is rather trivial. The only degrees of freedom during polynomial model structure selection are the selection of the polynomial degree and the continuity order of the simplex spline polynomials. It was explained in Sec. 2.2 that a spline space of a given degree d is a basis for the complete space of polynomials of degree d. This means that selection of the degree of the spline space is equivalent to selecting all possible polynomials of the same total (and maximum) degree. The specific form of the spline polynomials is determined during parameter estimation.

One important aspect of polynomial model structure selection is B-net propagation, see Sec. 3.3.4. B-net propagation is the propagation of local disturbances through the B-nets of the multivariate simplex splines. B-net propagation was proved to be a function of the spline degree and continuity order, see **Theorem** 7 and **Theorem** 8, and is most likely to occur on Type I triangulations. The Type I triangulation is a very convenient triangulation type, as it is the simplest possible triangulation of the hypercube, see e.g. Figure 2.23 and Figure 2.24. B-net propagation is not an issue during the prototype phase, because only a single triangulated hypercube is used. During the refinement phase, however, B-net propagation is an important factor in the determination of the spline degree and continuity order.

4.1.3 A linear regression model for B-form polynomials

In this paragraph a new linear regression model structure for simplex splines will be presented. This model structure was first introduced by the author in [40]. The regression model structure presented here is essential when using simplex splines inside a framework for system identification, as it enables the use of standard parameter estimation techniques for the estimation of the B-coefficients. It is assumed at this point that the dimensions, triangulation, and spline space have already been defined during geometric and polynomial model structure selection.

First, consider the vector-scalar pair of observations $(\mathbf{x}(i), y(i))$, which are related as follows:

$$y(i) = f(\mathbf{x}(i)) + r(i), \;\; i = 1, 2, \ldots, N, \;\; \in \mathbb{R}^1, \tag{4.1}$$

with f an unknown function and with $r(i)$ a residual term. We now introduce a regression model structure for approximating f that is equivalent to a linear combination of B-form polynomials Eq. 2.59 of degree d, defined on a triangulation consisting of J simplices:

$$y(i) = \sum_{j=1}^{J} \sum_{|\kappa|=d} c_{\kappa}^{t_j} B_{\kappa}^{d}(\mathbf{b}(i)) + r(i), \tag{4.2}$$

with $\mathbf{b}(i)$ the barycentric coordinate of $\mathbf{x}(i)$ with respect to the simplex t_j as in Eq. 2.32. The model structure in Eq. 4.2 is an entirely valid linear regression structure, but it would not lead to a meaningful approximation scheme because all data points $\mathbf{x}(i)$ contribute to the approximation on a simplex t_j, regardless of whether they are inside or outside t_j. In order to obtain a per-simplex interpolation scheme, a simplex membership operator $\delta_{jk(i)}$ is introduced:

$$\delta_{jk(i)} = \left\{ \begin{array}{ll} 1, & \text{if } j = k(i) \\ 0, & \text{if } j \neq k(i) \end{array} \right. , \tag{4.3}$$

with $k(i)$ an index function that produces the index of the simplex which contains the data point $\mathbf{x}(i)$, i.e., $\mathbf{x}(i) \in t_{k(i)}$, $\forall i$. The membership operator Eq. 4.3 is now applied to the regression model Eq. 4.2, which leads to the multivariate simplex spline based linear regression model:

$$y(i) = \sum_{j=1}^{J} \left(\delta_{jk(i)} \sum_{|\kappa|=d} c_{\kappa}^{t_j} B_{\kappa}^{d}(\mathbf{b}(i)) \right) + r(i). \tag{4.4}$$

This expression can be restated in a matrix form that includes all measurements. For this purpose, a vector formulation of the B-form from Eq. 2.59 must first be defined. First, let $\mathbf{B}_{t_j}^{d}$ be the vector of lexicographically sorted basis polynomial terms for the simplex t_j:

$$\mathbf{B}_{t_j}^{d}(i) = [B_{\kappa}^{d,t_j}(\mathbf{b}(i))]_{|\kappa|=d} \in \mathbb{R}^{1\times \hat{d}}, \tag{4.5}$$

in which the simplex identifier t_j was added to the definition of the basis function for clarity. With Eq. 2.64 and Eq. 2.66 the per-simplex B-form in vector formulation is:

$$p(\mathbf{b}(i)) = \mathbf{B}_{t_j}^{d}(i) \cdot \mathbf{c}^{t_j}. \tag{4.6}$$

The per-simplex $\hat{d} \times \hat{d}$ diagonal data membership matrix for observation i is defined as follows:

$$\mathbf{D}_{t_j}(i) = \left[(\delta_{j,k(i)})_{q,q} \right]_{q=1}^{\hat{d}} \in \mathbb{R}^{\hat{d}\times \hat{d}}. \tag{4.7}$$

The full-triangulation basis function vector for a single observation is:

$$\mathbf{B}^{d}(i) = [\ \mathbf{B}_{t_1}^{d}(i) \quad \mathbf{B}_{t_2}^{d}(i) \quad \cdots \quad \mathbf{B}_{t_J}^{d}(i)\ ] \in \mathbb{R}^{1\times J\cdot\hat{d}}. \tag{4.8}$$

The block diagonal full-triangulation data membership matrix $\mathbf{D}(i)$ for a single observation is a matrix with $\mathbf{D}_{t_j}(i)$ blocks on the main diagonal:

$$\mathbf{D}(i) = \left[\left(\mathbf{D}_{t_j}(i) \right)_{j,j} \right]_{j=1}^{J} \in \mathbb{R}^{(J \cdot \hat{d}) \times (J \cdot \hat{d})}. \tag{4.9}$$

Using Eq. 3.27, Eq. 4.8 and Eq. 4.9 the B-form of the multivariate simplex spline for the complete triangulation in vector form becomes:

$$P(\mathbf{b}(i)) = \mathbf{B}^d(i) \cdot \mathbf{D}(i) \cdot \mathbf{c}. \tag{4.10}$$

Now let $\mathbf{X}(i)$ be a single row in the full-triangulation regression matrix for all observations $\mathbf{X} \in \mathbb{R}^{N \times J \cdot \hat{d}}$ as follows:

$$\mathbf{X}(i) = \mathbf{B}^d(i) \cdot \mathbf{D}(i) \ \in \mathbb{R}^{1 \times J \cdot \hat{d}}. \tag{4.11}$$

For a single observation on y we then have:

$$y(i) = \mathbf{X}(i)\mathbf{c} + r(i), \tag{4.12}$$

which, for all observations, leads to the well-known formulation:

$$\mathbf{Y} = \mathbf{X}\mathbf{c} + \mathbf{r} \ \in \mathbb{R}^{N \times 1}. \tag{4.13}$$

4.2 Spline Model Estimation

In this section, two estimators for the B-coefficients of multivariate simplex spline functions will be presented. These estimators differ in scope and complexity and together provide the user with a complete suite of solvers that are applicable to a wide range of simplex spline data modeling problems.

The first estimator, discussed in Sec. 4.2.1, is a generalized least squares estimator which was first presented in [40]. The generalized least squares estimator relies on the linear regression scheme for simplex splines from Sec. 4.1.3. The second estimator is the constrained recursive least squares estimator, which is discussed in Sec. 4.2.2. The constrained recursive least squares estimator was first introduced by the author in [41], and was designed for use in real-time adaptive modeling applications like adaptive model based controllers.

4.2.1 Generalized Least Squares with simplex splines

In Sec. 4.1.3 it was shown that the scattered data modeling problem for multivariate simplex splines can be formulated in the form of the linear regression problem from Eq. 4.13, which is repeated here:

$$\mathbf{Y} = \mathbf{X}\mathbf{c} + \mathbf{r} \ \in \mathbb{R}^{N\times 1},$$

with $\mathbf{Y}$ the vector containing all observations, with $\mathbf{X}$ the regression matrix for all observations constructed using Eq. 4.11, and with $\mathbf{c}$ the global vector of B-coefficients from Eq. 3.27.

Equation Eq. 4.13 can be solved using many different methods, depending on the assumptions made on the nature of the residual term $\mathbf{r}$. A generalized least squares (GLS) estimator for Eq. 4.13 will be introduced, which implies that the residual contains white noise of varying magnitude:

$$E(\mathbf{r}) = 0, \quad Cov(\mathbf{r}) = \mathbf{\Sigma}, \tag{4.14}$$

with $\mathbf{\Sigma} \ \in \mathbb{R}^{N\times N}$ the residual covariance matrix, which is both non-singular and positive definite. The well-known (see e.g. [87]) GLS cost function is:

$$J_{GLS}(\mathbf{c}) = \frac{1}{2}(\mathbf{Y} - \mathbf{X}\mathbf{c})^{\top}\mathbf{\Sigma}^{-1}(\mathbf{Y} - \mathbf{X}\mathbf{c}). \tag{4.15}$$

Up to this point it has not been discussed how continuity between simplices is achieved. As explained in Chpt. 3, the continuity conditions are contained in the smoothness matrix $\mathbf{H}$ from Eq. 3.102. The continuity conditions act as linear equality constraints on B-coefficients located in the continuity structure of a triangulation. Therefore, the complete optimization problem can be stated as an equality constrained GLS problem (ECGLS) as follows:

$$\min_{\mathbf{c}} \ J_{GLS}(\mathbf{c}), \text{ subject to } \mathbf{G}\mathbf{c} = \mathbf{g}. \tag{4.16}$$

with $\mathbf{G} \ \in \mathbb{R}^{G\times J\cdot\hat{d}}$ a matrix containing a total of G linear constraints, and with $\mathbf{g} \ \in \mathbb{R}^{G\times 1}$ a vector containing a total of G constraint values as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{H} & \mathbf{D} \end{bmatrix}^{\top}, \ \mathbf{g} = \begin{bmatrix} \mathbf{0} & \mathbf{d} \end{bmatrix}^{\top}, \tag{4.17}$$

with $\mathbf{H}$ the all-important smoothness matrix from Eq. 3.102, and with $\mathbf{D}$ and $\mathbf{d}$ a constraint matrix and vector, respectively.

Using Lagrange multipliers the constrained optimization problem from Eq. 4.16 can be stated as:

$$\mathcal{L}(\mathbf{c}, \nu) = J_{GLS}(\mathbf{c}) + \nu^{T}(\mathbf{G}\mathbf{c} - \mathbf{g}), \tag{4.18}$$

with ν a vector of Lagrange multipliers. Equating the partial derivatives of Eq. 4.18 to zero and writing the results in matrix form results in the Karush-Kuhn-Tucker (KKT) system:

$$\begin{bmatrix} \mathbf{X}^\top \boldsymbol{\Sigma}^{-\mathbf{1}} \mathbf{X} & \mathbf{G}^\top \\ \mathbf{G} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \nu \end{bmatrix} = \begin{bmatrix} \mathbf{X}^\top \boldsymbol{\Sigma}^{-\mathbf{1}} \mathbf{Y} \\ \mathbf{g} \end{bmatrix}, \tag{4.19}$$

Using the entities from Eq. 4.13 and the residual covariance matrix from Eq. 4.14 the *dispersion matrix* $\mathbf{Q}$ is defined as follows:

$$\mathbf{Q} := \mathbf{X}^\top \boldsymbol{\Sigma}^{-\mathbf{1}} \mathbf{X} \ \in \mathbb{R}^{J \cdot \hat{d} \times J \cdot \hat{d}}. \tag{4.20}$$

The left hand matrix in Eq. 4.19 is known in the literature as the Karush-Kuhn-Tucker matrix [9]. The complete KKT system in Eq. 4.19 can be solved using a variety of different methods, see e.g. [9] and [112]. Most methods in the literature require that that the dispersion matrix $\mathbf{Q}$ is non-singular. Awanou and Lai presented a matrix iterative solver which converges to $\mathbf{c}$ for a singular $\mathbf{Q}$ as long as $\mathbf{Q}$ is positive definitive on the kernel of $\mathbf{H}$ [7, 6]. The first iterate for $\mathbf{c}$ is:

$$\mathbf{c}^{(1)} = (2\mathbf{Q} + \frac{1}{\epsilon}\mathbf{G}^T\mathbf{G})^{-1}(2\mathbf{X}^\top \boldsymbol{\Sigma}^{-\mathbf{1}} \mathbf{Y} + \frac{1}{\epsilon}\mathbf{G}^T\mathbf{g} - \mathbf{G}^T\nu^{(0)}), \tag{4.21}$$

with $\epsilon > 0$, and with $\nu^{(0)}$ an initial guess for the Lagrange multipliers[1]. The iteration sequence is:

$$\mathbf{c}^{(k+1)} = (2\mathbf{Q} + \frac{1}{\epsilon}\mathbf{G}^T\mathbf{G})^{-1}(2\mathbf{Q}\mathbf{c}^{(k)} + \frac{1}{\epsilon}\mathbf{G}^T\mathbf{g}). \tag{4.22}$$

The iterative solver was found to be very efficient computationally when solving large scale data modeling problems in which more than 20000 B-coefficients had to be estimated.

The exact solution of the KKT system from Eq. 4.19 is:

$$\begin{bmatrix} \hat{\mathbf{c}} \\ \hat{\nu} \end{bmatrix} = \begin{bmatrix} \mathbf{C_1} & \mathbf{C_2} \\ \mathbf{C_3} & \mathbf{C_4} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}^\top \boldsymbol{\Sigma}^{-\mathbf{1}} \mathbf{Y} \\ \mathbf{g} \end{bmatrix}, \tag{4.23}$$

with $\hat{\mathbf{c}}$ and $\hat{\nu}$ estimators for $\mathbf{c}$ and ν respectively. Using Eq. 4.23 a more explicit expression for the ECGLS estimator for the B-coefficients can be derived as follows:

$$\hat{\mathbf{c}} = \mathbf{C_1} \cdot \mathbf{X}^\top \boldsymbol{\Sigma}^{-\mathbf{1}} \mathbf{Y}. \tag{4.24}$$

Rao shows in [177] that the matrix in Eq. 4.23 is equal to the pseudo inverse of the KKT matrix:

$$\begin{bmatrix} \mathbf{C_1} & \mathbf{C_2} \\ \mathbf{C_3} & \mathbf{C_4} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} & \mathbf{G}^\top \\ \mathbf{G} & 0 \end{bmatrix}^{+}. \tag{4.25}$$

---

[1] In practice, it was found that $\epsilon = 10^{-6}$ resulted in fast convergence of the iterative solver

Note that the sizes of the submatrices $\mathbf{C_1}$, $\mathbf{C_2}$ and $\mathbf{C_3}$ in Eq. 4.25 are equal to the sizes of $\mathbf{Q}$, $\mathbf{G}^\top$ and $\mathbf{G}$ respectively.

In order to obtain accurate estimates for the B-coefficient variances, it is required that the KKT matrix in Eq. 4.19 is non-singular. The KKT matrix is found to be non-singular if the dispersion matrix $\mathbf{Q}$ from Eq. 4.20 is positive definite on the kernel of the constraint matrix $\mathbf{G}$:

$$\mathbf{Gc} = 0,\ \mathbf{c} \neq 0 \implies \mathbf{c}^\top \mathbf{Qc} > 0. \tag{4.26}$$

This statement holds if $\mathbf{G}$ and $\mathbf{Q}$ are both of full rank. The proof of Eq. 4.26 for the general KKT matrix is well known in the literature, see e.g. [9]. The constraint matrix $\mathbf{G}$ from Eq. 4.17 can only be of full rank when the smoothness matrix $\mathbf{H}$ and constraint matrix $\mathbf{D}$ are both of full rank, and when no linear dependencies exist between the rows of $\mathbf{H}$ and $\mathbf{D}$. The following theorem will prove that the rank of $\mathbf{Q}$ is dependent on the volume and configuration of the data.

Theorem 15. *The dispersion matrix* $\mathbf{Q}$ *is non-singular when every simplex in a triangulation* $\mathcal{T}$ *contains a minimum of* $\hat{d}$ *non-coplanar data points, with* $\hat{d}$ *as in Eq. 2.51.*

Proof. The proof requires that the data content of every individual simplex is considered separately. We therefore first re-order the rows in $\mathbf{X}$ and $\mathbf{\Sigma}^{-1}$ such that they are in block diagonal form. This operation does not alter the rank of $\mathbf{Q}$. We denote the per-simplex blocks $\mathbf{X}_j$, and $\mathbf{\Sigma}_j^{-1}$ with $j = 0, 1, \ldots, J$. The number of data points in the simplex t_j is N_j. The rank of $\mathbf{Q}$ is now simply the sum of the ranks of the diagonal sub blocks:

$$\text{rank } \mathbf{Q} = \sum_{j=0}^{J} \text{rank } \mathbf{X}_j^\top \mathbf{\Sigma}_j^{-1} \mathbf{X}_j. \tag{4.27}$$

For $\mathbf{Q}$ to be of full rank, we must have for every set of blocks:

$$\text{rank } \mathbf{X}_j^\top \mathbf{\Sigma}_j^{-1} \mathbf{X}_j = \hat{d}. \tag{4.28}$$

In the following, use is made of a nested form of the rank statement from [74]:

$$\begin{aligned} \text{rank } \mathbf{X}_j^\top \quad + \quad & \text{rank } \mathbf{\Sigma}_j^{-1}\mathbf{X}_j - N_j \leq \text{rank } \mathbf{X}_j^\top \mathbf{\Sigma}_j^{-1} \mathbf{X}_j \leq \\ & \min\{\text{rank } \mathbf{X}_j^\top, \text{rank } \mathbf{\Sigma}_j^{-1}\mathbf{X}_j\}, \end{aligned} \tag{4.29}$$

where the rank of $\mathbf{\Sigma}_j^{-1}\mathbf{X}_j$ is given by:

$$\begin{aligned} \text{rank } \mathbf{\Sigma}_j^{-1} + \text{rank } \mathbf{X}_j - N_j \leq \text{rank } & \mathbf{\Sigma}_j^{-1}\mathbf{X}_j \leq \\ & \min\{\text{rank } \mathbf{\Sigma}_j^{-1}, \text{rank } \mathbf{X}_j\}. \end{aligned} \tag{4.30}$$

Because Σ is invertible, its rank is equal to the total number of data points N_j in simplex t_j.

The rank of $\mathbf{X}_j$ is:

$$\operatorname{rank} \mathbf{X}_j = \min\{N_j, \hat{d}\}. \tag{4.31}$$

When $N_j < \hat{d}$, i.e. when there are less than $\hat{d}$ non-coplanar data points in simplex t_j, we get rank $\mathbf{X}_j = N_j$ with which the inequalities in Eq. 4.30 reduce to the following equality:

$$\operatorname{rank} \mathbf{\Sigma}_j^{-1} \mathbf{X}_j = N_j. \tag{4.32}$$

Using this result in Eq. 4.29, and eliminating the inequalities the following results are obtained:

$$\operatorname{rank} \mathbf{X}_j^\top \mathbf{\Sigma}_j^{-1} \mathbf{X}_j = N_j < \hat{d}. \tag{4.33}$$

This result proves that $\mathbf{Q}$ is singular when there are one or more simplices with less than $\hat{d}$ non-coplanar data points. When $N_j \geq \hat{d}$ we have rank $\mathbf{X}_j = \hat{d}$ with which the inequalities in Eq. 4.30 reduce to:

$$\operatorname{rank} \mathbf{\Sigma}_j^{-1} \mathbf{X}_j = \hat{d}. \tag{4.34}$$

Substituting this result in Eq. 4.29 and eliminating the inequalities we get:

$$\operatorname{rank} \mathbf{X}_j^\top \mathbf{\Sigma}_j^{-1} \mathbf{X}_j = \hat{d}. \tag{4.35}$$

Which proves that $\mathbf{Q}$ is non-singular only if $N_j \geq \hat{d}$.

□

Theorem 16. *First let* $\mathbf{M}$ *be the Fisher information matrix of the KKT system from Eq. 4.23 as follows:*

$$\mathbf{M} = \begin{bmatrix} \mathbf{Q} & \mathbf{G}^\top \\ \mathbf{G} & 0 \end{bmatrix}^{-1} \in \mathbb{R}^{(J\cdot\hat{d}+G)\times(J\cdot\hat{d}+G)}. \tag{4.36}$$

Then $\mathbf{M}$ *is non-singular if* and only if *the dispersion matrix* $\mathbf{Q}$ *from Eq. 4.20 and the constraint matrix* $\mathbf{G}$ *from Eq. 4.17 are both of full rank.*

Proof. The information matrix from Eq. 4.36 can be factorized as follows:

$$\mathbf{M} = \begin{bmatrix} \mathbf{Q} & 0 \\ \mathbf{G} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Q}^{-1} & 0 \\ 0 & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{Q} & \mathbf{G}^T \\ 0 & \mathbf{I} \end{bmatrix}, \tag{4.37}$$

with $\mathbf{C} = \mathbf{G}\mathbf{Q}^{-1}\mathbf{G}^T$ an $G \times G$ matrix and with $\mathbf{I}$ the $G \times G$ identity matrix. The rank of the non-singular lower and upper block triangular matrices in Eq. 4.37 is at least, and can be greater than, the sum of the ranks of the diagonal blocks [74]. The ranks of the diagonal blocks follow directly from Eq. 4.27 and the fact that the constraint matrix $\mathbf{G}$ is considered to be of (full) rank G:

$$\text{rank}\begin{bmatrix} \mathbf{Q} & 0 \\ \mathbf{G} & \mathbf{I} \end{bmatrix} = \text{rank}\begin{bmatrix} \mathbf{Q} & \mathbf{G}^T \\ 0 & \mathbf{I} \end{bmatrix} = J \cdot \hat{d} + G. \tag{4.38}$$

The lower and upper block diagonal matrices in Eq. 4.37 are of full rank if $\mathbf{Q}$ is of full rank. Rank is unchanged by left and right multiplication of non-singular matrices [74], therefore we must have:

$$\begin{aligned} \text{rank}\,\mathbf{M} &= \text{rank}\begin{bmatrix} \mathbf{Q}^{-1} & 0 \\ 0 & -\mathbf{C} \end{bmatrix} \\ &= \text{rank}\,\mathbf{Q}^{-1} + \text{rank}\,\mathbf{C}. \end{aligned} \tag{4.39}$$

The rank of $\mathbf{Q}^{-1}$ is equal to $J \cdot \hat{d}$ leaving only the determination of the rank of $\mathbf{C}$. The square $G \times G$ matrix matrix $\mathbf{C}$ is formed by left and right multiplication with the non-square matrices $\mathbf{G}$ and $\mathbf{G}^T$. To prove that $\mathbf{C}$ is of full rank, we again make use of the nested form of the rank statement from [74]:

$$\text{rank}\,\mathbf{G} + \text{rank}\,\mathbf{P} - R \leq \text{rank}\,\mathbf{GP} \leq \min\{\text{rank}\,\mathbf{G}, \text{rank}\,\mathbf{P}\}, \tag{4.40}$$

with $\mathbf{P} = \mathbf{Q}^{-1}\mathbf{G}^T$. The rank of $\mathbf{P}$ follows from the same rank statement:

$$\begin{aligned} \text{rank}\,\mathbf{Q}^{-1} + \text{rank}\,\mathbf{G}^T - J \cdot \hat{d} \leq \text{rank}\,\mathbf{P} \leq \\ \min\{\text{rank}\,\mathbf{Q}^{-1}, \text{rank}\,\mathbf{G}^T\}. \end{aligned} \tag{4.41}$$

Because we have rank $\mathbf{G}^T \leq$ rank $\mathbf{Q}^{-1}$ this reduces to:

$$\text{rank}\,\mathbf{G}^T \leq \text{rank}\,\mathbf{P} \leq \text{rank}\,\mathbf{G}^T.$$

And therefore:

$$\text{rank}\,\mathbf{P} = \text{rank}\,\mathbf{Q}^{-1}\mathbf{G}^T = \text{rank}\,\mathbf{G}^T. \tag{4.42}$$

Returning to Eq. 4.40, and using the result from Eq. 4.42 we get:

$$\begin{aligned} \text{rank}\,\mathbf{G} + \text{rank}\,\mathbf{G}^T - R &\leq \text{rank}\,\mathbf{GQ}^{-1}\mathbf{G}^T \\ &\leq \min\{\text{rank}\,\mathbf{G}, \text{rank}\,\mathbf{G}^T\}, \end{aligned} \tag{4.43}$$

with which the following rank statement for $\mathbf{C}$ is found:

$$\text{rank}\,\mathbf{GQ}^{-1}\mathbf{G}^T = \text{rank}\,\mathbf{G} = \text{rank}\,\mathbf{C}, \tag{4.44}$$

proving the theorem. This proof is also valid for the OLS information matrix; in that case $\boldsymbol{\Sigma}$ is replaced with the identity matrix of the same size.

□

A complete and easy to reproduce example of the use of the generalized least squares estimator was provided in [40].

4.2.2 Constrained recursive regression with simplex splines

In this section an equality constrained Recursive Least Squares (RLS) estimator for the B-coefficients of multivariate simplex splines is presented. This recursive estimator is based on the linear regression model for simplex splines introduced in [40], and repeated in Sec. 4.1.3. The RLS estimator significantly reduces the computational efforts for real-time model modification with simplex splines, because no matrix inversions are required beyond the initial covariance matrix estimation.

First, let $J(\mathbf{c})$ be a least squares cost function in the global vector of B-coefficients $\mathbf{c} = [\mathbf{c}^{t_j}]_{j=1}^{J} \in \mathbb{R}^{J\cdot\hat{d}\times 1}$ as follows:

$$J(\mathbf{c}) = (\mathbf{y} - \mathbf{X}\mathbf{c})^\top(\mathbf{y} - \mathbf{X}\mathbf{c}), \tag{4.45}$$

with $\mathbf{X} \in \mathbb{R}^{N\times J\cdot\hat{d}}$ the matrix of B-form regressors for N observations as derived in Eq. 4.13, and with $\mathbf{y}$ the vector containing all N observations. The equality constrained least squares (ECLS) optimization problem then is:

$$\min_{\mathbf{c}} J(\mathbf{c}), \text{ subject to } \mathbf{G}\mathbf{c} = \mathbf{g}, \tag{4.46}$$

with $\mathbf{G}$ the matrix with linear constraints, and with $\mathbf{g}$ a vector of constraints as defined in Eq. 4.17. Note that $\mathbf{G}$ contains both the smoothness matrix $\mathbf{H}$ and the linear constraint matrix $\mathbf{D}$. In this case, $\mathbf{D}$ is the differential constraint matrix as defined in Eq. 3.108 in Chpt. 3.

There are many different methods available in the literature for solving the ECLS problem Eq. 4.46 such as null-space methods [112] and Lagrange multiplier methods [9]. In the literature the Lagrange multiplier method has been successfully used with simplex splines, see e.g. [7][40]. Using the null-space method from [112], the B-coefficient $\hat{\mathbf{c}}$ estimator minimizing the LS cost function $J(\mathbf{c})$ subject to the linear equality constraint $\mathbf{G}\mathbf{c} = \mathbf{g}$ is:

$$\hat{\mathbf{c}} = \mathbf{G}^+\mathbf{g} + (\mathbf{X}\mathbf{Z})^+(\mathbf{y} - \mathbf{X}\mathbf{G}^+\mathbf{g}), \tag{4.47}$$

with $\mathbf{Z}$ an orthogonal projector onto the null-space of $\mathbf{G}$:

$$\mathbf{Z} = \mathbf{I} - \mathbf{G}^+\mathbf{G} \ \in \mathbb{R}^{J\cdot\hat{d}\times J\cdot\hat{d}}. \tag{4.48}$$

Lawson and Hanson proved in [112] that a unique solution for $\hat{\mathbf{c}}$ exists if the matrix $\begin{bmatrix} X & G \end{bmatrix}$ is of full (row) rank. In this case this means the following:

$$\text{rank}\left(\begin{bmatrix} \mathbf{X} \\ \mathbf{G} \end{bmatrix}\right) = J \cdot \hat{d}. \tag{4.49}$$

The expression for $\hat{\mathbf{c}}$ from Eq. 4.47 can be restated in terms of the parameter covariance matrix $\mathbf{P}$ as follows:

$$\hat{\mathbf{c}} = \mathbf{G}^{+}\mathbf{g} + \mathbf{P}\mathbf{X}^{\top}(\mathbf{y} - \mathbf{X}\mathbf{G}^{+}\mathbf{g}), \tag{4.50}$$

where we make use of the following properties of the Moore-Penrose pseudo inverse, and the fact that $\mathbf{Z}$ is symmetric:

$$\begin{aligned} (\mathbf{X}\mathbf{Z})^{+} &= \left((\mathbf{X}\mathbf{Z})^{\top}(\mathbf{X}\mathbf{Z})\right)^{+}(\mathbf{X}\mathbf{Z})^{\top} \\ &= \left(\mathbf{Z}\mathbf{X}^{\top}\mathbf{X}\mathbf{Z}\right)^{+}\mathbf{Z}\mathbf{X}^{\top} \\ &= \mathbf{P}\mathbf{X}^{\top}, \end{aligned}$$

with $\mathbf{P}$ the parameter covariance matrix estimate for all t defined as follows:

$$\mathbf{P} = \left(\mathbf{Z}\mathbf{X}^{\top}\mathbf{X}\mathbf{Z}\right)^{+} \in \mathbb{R}^{J\cdot\hat{d}\times J\cdot\hat{d}}. \tag{4.51}$$

A theorem will now be presented for the recursive equality constrained generalized least squares estimator for the B-coefficients of multivariate simplex splines. This theorem was already presented by Zhu and Li [220], but the proof offered in this thesis differs from these authors in the sense that it is based on the matrix pseudo inversion lemma from Ogawa [163], and therefore more akin to the proof of the ordinary unconstrained recursive least squares estimator from literature.

Theorem 17. *The RECLS estimator for the B-coefficients of a constrained simplex spline function is:*

$$\begin{aligned} \mathbf{L}(t+1) &= \mathbf{P}(t)\mathbf{x}^{\top}(t+1)\left[1 + \mathbf{x}(t+1)\mathbf{P}(t)\mathbf{x}^{\top}(t+1)\right]^{-1}, \\ \mathbf{P}(t+1) &= \mathbf{P}(t) - \mathbf{L}(t+1)\mathbf{x}(t+1)\mathbf{P}(t), \\ \hat{\mathbf{c}}(t+1) &= \hat{\mathbf{c}}(t) + \mathbf{L}(t+1)\left[\mathbf{y}(t+1) - \mathbf{x}(t+1)\hat{\mathbf{c}}(t)\right], \end{aligned} \tag{4.52}$$

with $\mathbf{P}$ *parameter covariance matrix estimate from Eq. 4.51 having the initial condition:*

$$\mathbf{P}(0) = \left(\mathbf{Z}\mathbf{X}^{\top}(0)\mathbf{X}(0)\mathbf{Z}\right)^{+}, \tag{4.53}$$

and with $\hat{\mathbf{c}}$ *the estimated B-coefficients having the initial condition:*

$$\hat{\mathbf{c}}(0) \quad = \quad \mathbf{G}^{+}\mathbf{g} + \mathbf{P}(0)\mathbf{X}^{\top}(0)(\mathbf{y} - \mathbf{X}(0)\mathbf{G}^{+}\mathbf{g}). \tag{4.54}$$

Proof. We start the proof of Theorem 17 by observing that the parameter covariance matrix $\mathbf{P}$ in Eq. 4.51 can be decomposed into an 'old' part and a 'new' part:

$$\mathbf{P} \quad = \quad \left[\mathbf{Z}\mathbf{X}^{\top}(t)\mathbf{X}(t)\mathbf{Z} + \mathbf{Z}\mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{Z}\right]^{+}. \tag{4.55}$$

We now use the theorem for the matrix pseudo inversion lemma from Ogawa [163] which states that:

$$(\mathbf{A} + \mathbf{B}\mathbf{C}\mathbf{B}^{\top})^{+} \quad = \quad \mathbf{A}^{+} - \mathbf{A}^{+}\mathbf{B}\left[\mathbf{C}^{-1} + \mathbf{B}^{\top}\mathbf{A}^{+}\mathbf{B}\right]^{-1}\mathbf{B}^{\top}\mathbf{A}^{+}, \tag{4.56}$$

if and only if $R(\mathbf{A}) \supset R(\mathbf{B})$, or the range of $\mathbf{A}$ is a superset of the range of $\mathbf{B}$. In our case we have $\mathbf{A} = \mathbf{Z}\mathbf{X}^{\top}(t)\mathbf{X}(t)\mathbf{Z}$, $\mathbf{B} = \mathbf{Z}\mathbf{x}^{\top}(t+1)$ and $\mathbf{C} = \mathbf{I}$. It is now easy to see that the range statement holds for the given $\mathbf{A}$ and $\mathbf{B}$, because B is a vector in $\mathbb{R}^{J\cdot\hat{d}\times 1}$. with the matrix pseudo inversion lemma, the expression for the parameter covariance matrix estimation from Eq. 4.55 becomes:

$$\begin{aligned}
\mathbf{P}(t+1) \quad &= \quad \mathbf{P}(t) - \mathbf{P}(t)\mathbf{x}^{\top}(t+1)\cdot \\
&\qquad \left[1 + \mathbf{x}(t+1)\mathbf{P}(t)\mathbf{x}^{\top}(t+1)\right]^{-1}\cdot\mathbf{x}(t+1)\mathbf{P}(t) \\
&= \quad \mathbf{P}(t) - \mathbf{L}(t+1)\mathbf{x}(t+1)\mathbf{P}(t),
\end{aligned} \tag{4.57}$$

with $\mathbf{L}(t+1)$ a scaling vector as follows:

$$\mathbf{L}(t+1) \quad = \quad \mathbf{P}(t)\mathbf{x}^{\top}(t+1)\cdot\left[1 + \mathbf{x}(t+1)\mathbf{P}(t)\mathbf{x}^{\top}(t+1)\right]^{-1}. \tag{4.58}$$

From Eq. 4.50 it follows that the update value for the B-coefficients, i.e. $\hat{\mathbf{c}}(t+1)$, can be formulated in terms of a combination of old data and new data:

$$\begin{aligned}
\hat{\mathbf{c}}(t+1) \quad = \quad & \mathbf{G}^{+}\mathbf{g} + \mathbf{P}(t+1)\left[(\mathbf{X}^{\top}(t)\mathbf{y}(t) - \mathbf{X}^{\top}(t)\mathbf{X}(t)\mathbf{G}^{+}\mathbf{g}) + \right. \\
& \left.(\mathbf{x}^{\top}(t+1)\mathbf{y}(t+1) - \mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{G}^{+}\mathbf{g})\right].
\end{aligned} \tag{4.59}$$

Using the results from Eq. 4.57 to expand $\mathbf{P}(t+1)$ in Eq. 4.59 we get:

$$\begin{aligned}
\hat{\mathbf{c}}(t+1) \quad = \quad & \mathbf{G}^{+}\mathbf{g} + \\
& \left[\mathbf{I} - \mathbf{L}(t+1)\mathbf{x}(t+1)\right]\mathbf{P}(t)\left[(\mathbf{X}^{\top}(t)\mathbf{y}(t) - \mathbf{X}^{\top}(t)\mathbf{X}(t)\mathbf{G}^{+}\mathbf{g})\right] + \\
& \mathbf{P}(t+1)\left[(\mathbf{x}^{\top}(t+1)\mathbf{y}(t+1) - \mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{G}^{+}\mathbf{g})\right].
\end{aligned} \tag{4.60}$$

Observe that with Eq. 4.50 we have for $\hat{\mathbf{c}}(t)$:

$$\hat{\mathbf{c}}(t) \quad = \quad \mathbf{G}^{+}\mathbf{g} + \mathbf{P}(t)\left(\mathbf{X}^{\top}(t)\mathbf{y}(t) - \mathbf{X}^{\top}(t)\mathbf{X}(t)\mathbf{G}^{+}\mathbf{g}\right), \tag{4.61}$$

Using Eq. 4.61 the ex Eq. 4.60 can be simplified Eq. 4.60 to:

$$\begin{aligned}\hat{\mathbf{c}}(t+1) &= \mathbf{G}^{+}\mathbf{g} + [\mathbf{I} - \mathbf{L}(t+1)\mathbf{x}(t+1)]\,(\hat{\mathbf{c}}(t) - \mathbf{G}^{+}\mathbf{g}) + \\ &\quad \mathbf{P}(t+1)\left[\mathbf{x}^{\top}(t+1)\mathbf{y}(t+1) - \mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{G}^{+}\mathbf{g}\right] \\ &= \hat{\mathbf{c}}(t) + \mathbf{L}(t+1)\mathbf{x}(t+1)\left[\mathbf{G}^{+}\mathbf{g} - \hat{\mathbf{c}}(t)\right] + \\ &\quad \mathbf{P}(t+1)\left[\mathbf{x}^{\top}(t+1)\mathbf{y}(t+1) - \mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{G}^{+}\mathbf{g}\right].\end{aligned} \tag{4.62}$$

Lemma 1. *After initialization of the recursion the constraint terms can be dropped, which is equivalent to the following statement:*

$$0 = \mathbf{L}(t+1)\mathbf{x}(t+1)\mathbf{G}^{+}\mathbf{g} - \mathbf{P}(t+1)\mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{G}^{+}\mathbf{g}. \tag{4.63}$$

Proof. First, let the scalar $s(t+1)$ be defined as follows:

$$s(t+1) = \left(1 + \mathbf{x}(t+1)\mathbf{P}(t)\mathbf{x}^{\top}(t+1)\right)^{-1}. \tag{4.64}$$

Then, together with the expressions for $\mathbf{P}(t+1)$ from Eq. 4.57 and $\mathbf{L}(t+1)$ from Eq. 4.58 we get for Eq. 4.63:

$$\begin{aligned}0 &= \left[s(t+1)\mathbf{P}(t) - \mathbf{P}(t) + s(t+1)\mathbf{P}(t)\mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{P}(t)\right]\cdot \\ &\quad \mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{G}^{+}\mathbf{g},\end{aligned} \tag{4.65}$$

Multiplication with $s^{-1}(t+1)$ results in:

$$\begin{aligned}0 &= \left[\mathbf{P}(t) - s^{-1}(t+1)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{P}(t)\right]\cdot \\ &\quad \mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{G}^{+}\mathbf{g},\end{aligned} \tag{4.66}$$

which, after substitution of the inverse of the right hand side of Eq. 4.64 results in:

$$\begin{aligned}0 &= \left[\mathbf{P}(t) - \left(1 + \mathbf{x}(t+1)\mathbf{P}(t)\mathbf{x}^{\top}(t+1)\right)\mathbf{P}(t) + \right. \\ &\quad \left.\mathbf{P}(t)\mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{P}(t)\right]\cdot \mathbf{x}^{\top}(t+1)\mathbf{x}(t+1) \\ &= -\mathbf{x}(t+1)\mathbf{P}(t)\mathbf{x}^{\top}(t+1)\mathbf{P}(t)\mathbf{x}^{\top}(t+1)\mathbf{x}(t+1) + \\ &\quad \mathbf{P}(t)\mathbf{x}^{\top}(t+1)\mathbf{x}(t+1)\mathbf{P}(t)\mathbf{x}^{\top}(t+1)\mathbf{x}(t+1).\end{aligned} \tag{4.67}$$

It is easy to check that with $\mathbf{x}(t+1)$ a column vector, we have for the trace of

$\mathbf{P}(t)\mathbf{x}^\top(t+1)\mathbf{x}(t+1)$:

$$\mathbf{x}(t+1)\mathbf{P}(t)\mathbf{x}^\top(t+1) = tr\left(\mathbf{P}(t)\mathbf{x}^\top(t+1)\mathbf{x}(t+1)\right), \tag{4.68}$$

such that

$$\begin{aligned} 0 &= -tr\left(\mathbf{P}(t)\mathbf{x}^\top(t+1)\mathbf{x}(t+1)\right)\mathbf{P}(t)\mathbf{x}^\top(t+1)\mathbf{x}(t+1) + \\ &\quad \mathbf{P}(t)\mathbf{x}^\top(t+1)\mathbf{x}(t+1)\mathbf{P}(t)\mathbf{x}^\top(t+1)\mathbf{x}(t+1). \end{aligned} \tag{4.69}$$

Because $\mathbf{P}(t)\mathbf{x}^\top(t+1)\mathbf{x}(t+1)$ is a symmetric matrix this statement holds, proving the lemma. $\square$

Using the lemma, Eq. 4.62 can be simplified to:

$$\begin{aligned} \hat{\mathbf{c}}(t+1) &= \hat{\mathbf{c}}(t) - \mathbf{L}(t+1)\mathbf{x}(t+1)\hat{\mathbf{c}}(t) + \\ &\quad \mathbf{P}(t+1)\mathbf{x}^\top(t+1)\mathbf{y}(t+1). \end{aligned} \tag{4.70}$$

After some manipulations we find that

$$\mathbf{P}(t+1)\mathbf{x}^\top(t+1) = \mathbf{L}(t+1). \tag{4.71}$$

So finally,

$$\hat{\mathbf{c}}(t+1) = \hat{\mathbf{c}}(t) + \mathbf{L}(t+1)\left[\mathbf{y}(t+1) - \mathbf{x}(t+1)\hat{\mathbf{c}}(t)\right]. \tag{4.72}$$

which proves the theorem.

$\square$

A tutorial example of the use of the differentially constrained recursive least squares estimator was provided in [41].

4.3 Model Quality Assessment

In this section three different methods are presented for assessing the quality of multivariate simplex spline models. The first method is the model residue analysis, which is discussed in Sec. 4.3.1. The second quality assessment method is the statistical model quality assessment, which is presented in Sec. 4.3.2. The third, and final, model quality assessment method is the spline model stability analysis, which is introduced in Sec. 4.3.3. Together, these three quality assessment methods allow the rigorous and precise testing of the quality of a simplex spline model.

4.3.1 Model residue analysis

The model residue analysis provides a direct measure of the data fitting performance of a multivariate simplex spline function by analyzing the residual. The model residue analysis also plays an important role in determining whether the problem should be stated as an OLS problem or as a GLS problem, and should therefore always precede the statistical model quality assessment. When the problem is stated as a GLS problem, the estimation of the residual covariance matrix $\mathbf{\Sigma}$ is performed as a post processing step using the model residual from the OLS parameter estimation. The model residue ϵ follows from Eq. 4.13 and the estimated B-coefficient vector from Eq. 4.24:

$$\epsilon = \mathbf{Y} - \mathbf{X}\hat{\mathbf{c}}. \tag{4.73}$$

The model residue analysis consists of two parts. The first part is a direct analysis of the magnitude of the residual in the form of the residual RMS. The residual RMS provides an adequate overview of the overall data fitting quality of a spline function. The residual RMS is given by:

$$RMS(\epsilon) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\epsilon(i))}. \tag{4.74}$$

While the $RMS(\epsilon)$ is a useful model quality metric, it can in some cases lead to confusing results if the vector of validation observations contains numerical data of very low or very high magnitude. A more useful performance metric is the relative residual RMS which normalizes the RMS score to the range of validation observations as follows:

$$RMS_{rel}(\epsilon) = \frac{RMS(\epsilon)}{\max \mathbf{Y}_V - \min \mathbf{Y}_V}, \tag{4.75}$$

with $\mathbf{Y}_V$ the vector of validation observations.

When significant correlations are found to be present in the residue, the OLS assumption of uncorrelated residuals is violated, resulting in an underestimation of parameter variances. As a result, the problem may need to be restated as a GLS problem. The per-simplex estimate of the autocorrelation function $\hat{R}_t(k)$ is calculated as follows:

$$\hat{R}_t(k) = \frac{1}{N_t}\sum_{i=1}^{N_t-k}\epsilon_t(i)\epsilon_t(i+k), \quad k = 0, 1, \ldots, N_t, \tag{4.76}$$

with N_t the number of data points present in simplex t, and with ϵ_t the residual inside t.

The per-simplex residual is considered to be uncorrelated when the expression

$$\frac{|\hat{R}_t(k \neq 0)|}{\hat{R}_t(0)} \leq \frac{\mathcal{N}_\alpha}{\sqrt{N_t}}, \quad k > 0 \tag{4.77}$$

holds for at least $\alpha \cdot N_t$ of N_t values of the per-simplex autocorrelation function, with $\mathcal{N}_\alpha$ the α-level of the $\mathcal{N}(0,1)$ distribution, e.g $\mathcal{N}_{0.95} \approx= 1.96$. The expression Eq. 4.77 is a direct result of the central limit theorem, i.e. $\sqrt{N_t} \cdot R_t(k \neq 0) \xrightarrow{D} \mathcal{N}(0, \sigma^2_{\epsilon_t})$, with the assumption that the values $R_t(k \neq 0)$ are independent and identically distributed. This test for correlation was found to be more reliable than the Durbin-Watson statistic, which is calculated for only a single lag value and may miss larger scale correlations. The residual autocorrelation function also forms the basis for estimating the residual covariance matrix.

The second part of the model residue analysis is concerned with the estimation of the residual covariance matrix. Residual covariance matrix estimation is not a trivial task, and many different methods for constructing such matrices are presented in the literature, see e.g [177, 87, 92]. In this thesis the following method, presented by Klein and Morelli in [92], is used to calculate the $N_t \times N_t$ per-simplex residual covariance matrix $\mathbf{\Sigma}_t$:

$$\mathbf{\Sigma}_t = \frac{1}{N_t} \sum_{i=1}^{N_t - k} \epsilon_t(i)\epsilon_t(i+k), \quad k = 1, 2, \ldots, N_t \tag{4.78}$$

The per-simplex residual covariance matrix blocks are then assembled into the global, block diagonal $N \times N$ residual covariance matrix $\mathbf{\Sigma}_{\mathcal{T}}$:

$$\mathbf{\Sigma}_{\mathcal{T}} = \begin{bmatrix} \mathbf{\Sigma}_{t_1} & 0 & \cdots & 0 \\ 0 & \mathbf{\Sigma}^{t_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{\Sigma}^{t_J} \end{bmatrix} \tag{4.79}$$

Note that this residual covariance matrix structure does not consider inter-simplex residue correlations. Instead, the effects of inter simplex correlations on the parameter variances will be transmitted through the smoothness conditions in the ECGLS problem from Eq. 4.19.

The block diagonal structure of the global residual covariance matrix $\mathbf{\Sigma}_{\mathcal{T}}$ greatly reduces the computational efforts of determining $\mathbf{\Sigma}_{\mathcal{T}}^{-1}$, i.e:

$$\mathbf{\Sigma}_{\mathcal{T}}^{-1} = \begin{bmatrix} \mathbf{\Sigma}_{t_1}^{-1} & 0 & \cdots & 0 \\ 0 & \mathbf{\Sigma}_{t_2}^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{\Sigma}_{t_J}^{-1} \end{bmatrix} \tag{4.80}$$

where it is assumed that every per-simplex estimated covariance matrix block $\mathbf{\Sigma}_{t_j}$ is nonsingular and positive definite.

The third part of the model residue analysis is concerned with the creation of empirical confidence intervals for the B-coefficients of the multivariate spline. The empirical confidence intervals are based on the local (per-data point) model error and Chebychev's inequality:

$$Pr(|\epsilon - \mu(\epsilon)| \geq a\sigma) \leq \frac{1}{a^2} \tag{4.81}$$

with σ the local standard deviation of the model residue, and $0 < a \leq 1$. It can be seen that σ is not constant over the triangulation $\mathcal{T}$, as the quality of the spline model varies locally. For this reason a moving average algorithm was devised that locally determines the value of σ across the triangulation. The result of the moving average algorithm is a smoothened empirical confidence interval for every data point. The confidence intervals are then interpreted as data values which are used to construct a secondary multivariate spline function of the same degree as the original spline function. The B-coefficients of the resulting secondary spline function effectively are the empirical confidence bounds for the B-coefficients of the original spline function.

4.3.2 Statistical model quality assessment

The linear regression scheme for multivariate simplex splines from [40] leads to an additional model quality measure in the form of the parameter covariance matrix for the B-coefficients. Rao shows in [177] that when the pseudo inverse in Eq. 4.25 is equal to the true inverse, the parameter covariance matrix of $\hat{\mathbf{c}}$ is equal to the $\mathbf{C_1}$ submatrix in Eq. 4.25. In section Sec. 4.2 it was proved that the information matrices for ECGLS and ECOLS are invertible when sufficient data is present in every simplex, and when the linear constraint matrix $\mathbf{G}$ from Eq. 4.17 is of full rank; in the following it will be assumed that both these conditions are met. In that case the GLS parameter covariance matrix is given by:

$$Cov(\hat{\mathbf{c}}) = \mathbf{C_1} \tag{4.82}$$

with parameter variances equal to the main diagonal of $Cov(\hat{\mathbf{c}})$:

$$Var(\hat{\mathbf{c}}_q) = Cov(\hat{\mathbf{c}})_{q,q}, \;\; q = 1, 2, \ldots J \cdot \hat{d} \tag{4.83}$$

The OLS parameter covariance matrix is:

$$Cov(\hat{\mathbf{c}}_{\text{OLS}}) = \sigma^2 \mathbf{C_1} \tag{4.84}$$

with $\mathbf{C_1}$ the submatrix in Eq. 4.25 and with the global (estimated) variance of the residue σ^2. OLS parameter variances are determined as in Eq. 4.83 but with the GLS covariance matrix replaced with the OLS covariance matrix from Eq. 4.84.

The GLS Eq. 4.82 and OLS Eq. 4.84 parameter covariance matrices are used to compute statistical confidence intervals on the estimated B-coefficients of the multivariate spline. Kyriakides and Heydt present a conservative, simultaneous method for determining confidence intervals for parameter estimates called the Simultaneous Bonferroni method [93]:

$$|\mathbf{c}_q - \hat{\mathbf{c}}_q| \leq t_{\frac{\alpha}{2p}, N-p} \sqrt{Var(\hat{\mathbf{c}})_q} \tag{4.85}$$

with $t_{\frac{\alpha}{2p}, N-p}$ the Student's t distribution with $N - p$ degrees of freedom and significance level $\frac{\alpha}{2p}$ and with $p = J \cdot \hat{d}$. The B-coefficient confidence intervals lead to the definition of a secondary, or companion, spline function accompanying a primary modeling spline function. This companion spline function adds statistical upper and lower bounds on the function values of the modeling spline. The companion, statistical confidence spline function of the spline function $s_c \in S_d^r(\mathcal{T})$ is defined as follows:

$$s_c = \sum_{|\kappa|=d} \tau_\kappa^{t_j} B_\kappa^d(\mathbf{b}), \; j = 1, 2, \ldots, J, \tag{4.86}$$

with $\tau_\kappa^{t_j} = t_{\frac{\alpha}{2p}, N-p} \sqrt{Var(\hat{c}_\kappa^{t_j})}$ the B-coefficients of the statistical confidence spline.

4.3.3 Stability analysis

A final measure of model quality is its guaranteed stability within the model domain. This measure is hard, if not impossible, to determine for other nonlinear function approximators like neural networks. For the simplex splines, however, this measure is almost trivially simple, because it depends completely on the values of the estimated B-coefficients. In Chpt. 3 an expression for bounds on the values of B-form polynomials was presented. These bounds guarantee the well-behavedness of the spline function inside its domain. The expression for the bounds of B-form polynomials was given in Eq. 3.58, but is repeated here for clarity:

$$\frac{\|\mathbf{c}\|}{K} \leq \|p\| \leq \|\mathbf{c}\|,$$

with $||p||$ the maximum norm of the spline polynomial p and with $||\mathbf{c}||$ the maximum absolute value in the vector of per-simplex B-coefficients as follows:

$$||\mathbf{c}|| := \max_{|\kappa|=d} |c_\kappa|.$$

For the stability analysis, only the upper bound of the spline polynomial p is considered. This bound, follows from the maximum value in the vector of B-coefficients as follows:

$$||p|| \leq ||\mathbf{c}||, \tag{4.87}$$

that is, the maximum value in the vector of B-coefficients bounds the maximum value of the spline polynomial.

4.4 Scattered Data Modeling with Simplex Splines

In this section the new methodology for system identification with multivariate simplex splines is demonstrated with a numerical experiment. In the experiment, bivariate simplex spline functions of varying polynomial degree and continuity order were used to fit a highly nonlinear, bivariate dataset describing the aerodynamic pitching moment coefficient of the F-16 fighter aircraft. This dataset was generated by querying a 5-dimensional NASA wind tunnel model [155] of the F-16 at a set of scattered angle of attack and angle of sideslip coordinates. The generalized least squares estimator from Sec. 4.2.1 was then used to estimate the B-coefficients of the simplex splines. The main focus of this experiment will be on the polynomial model structure selection part of the complete spline model structure selection procedure as presented in Sec. 4.1. More accurately, this experiment can be considered as the inner loop of the prototype phase in simplex spline model identification shown earlier in Figure 4.1.

A number of different spline spaces were used in the demonstration experiment, with polynomial degrees ranging from 1 to 6. White noise of four different magnitudes was added to the dataset in order to test the noise sensitivity of the various spline spaces. The quality of the spline models was assessed using the residual analysis and the statistical analysis methods from 4.3.

The demonstration starts with Sec. 4.4.1 in which the dataset, the tested spline spaces and the triangulation are discussed. The results from the model residue analysis are then presented in Sec. 4.4.2. Following this, the results from the statistical model quality analysis are presented in Sec. 4.4.3. The results from the final model quality analysis method, the stability analysis, are presented in Sec. 4.4.4. Based on the results from the model residual analysis, empirical confidence bounds can be defined. The resulting empirical confidence bounds are discussed in Sec. 4.4.5. Finally, in Sec. 4.4.6 the statistical Bonferroni confidence bounds for two spline functions are presented.

4.4.1 Experiment setup

The following simplified equation of motion for the pitch rate acceleration of the F-16 is used:

$$\dot{q} \;=\; \frac{1}{2}\rho V^2 S_{ref} b_{ref} \frac{1}{I_{yy}} \left(C_m(\alpha, \beta) + \delta_e \right) + \frac{I_{xz}}{I_{yy}}(r^2 - p^2) - \frac{I_{zz} - I_{xx}}{I_{yy}} rp, \tag{4.88}$$

with $C_m(\alpha, \beta)$ an unknown nonlinear function of angle of attack α and angle of sideslip β. In this experiment, bivariate simplex spline functions $s(\alpha, \beta)$ of varying degree and continuity order will be used to approximate the nonlinear function $C_m(\alpha, \beta)$:

$$s(\alpha, \beta) \approx C_m(\alpha, \beta), \;\; s(\alpha, \beta) \in \mathcal{S}_d^r. \tag{4.89}$$

The parameters of the multivariate simplex spline function will be estimated using the ECGLS estimator from Eq. 4.23. For this, the problem is first stated in a slightly modified form of the linear regression problem from Eq. 4.13:

$$\mathbf{Y} = \mathbf{X}\mathbf{c} + \epsilon + \nu(k), \tag{4.90}$$

with $\mathbf{Y}$ the observation vector, $\mathbf{X}$ the matrix with B-form regressors evaluated at the data points $(\alpha(i), \beta(i))$, and with $\mathbf{c}$ the global vector of B-coefficients. The new function $\nu(k)$ in Eq. 4.90 is a variable magnitude noise generating function given by:

$$\nu(k) = 0.01k \cdot (\max\{\mathbf{Y}\} - \min\{\mathbf{Y}\}) \cdot \nu, \;\; k \geq 0, \tag{4.91}$$

with ν a uniformly distributed, zero-mean white noise sequence in the interval $[-0.5 \;\; 0.5]$. For the factor k the following values are taken: $k \in \{0, 1, 10, 100\}$. The noise function $\nu(k)$ enables the investigation of the effects of measurement noise on the quality of the spline approximations.

The data values used for estimating the spline parameters are generated by querying the 5-dimensional NASA wind tunnel model of the F-16 pitching moment coefficient at 15000 (α, β) data points which are randomly distributed over the flight envelope. The flight envelope is the rectangular region $-10° \leq \alpha \leq 45°, -30° \leq \alpha \leq 30°$. The set of 15000 data points is the identification dataset which is defined as $\mathcal{X}_I$. In Figure 4.4 the identification dataset $\mathcal{X}_I$ is drawn together with the triangulation $\mathcal{T}_{98}$. A single observation of $y(i)$ is:

$$\begin{aligned} y(i) \;&=\; \mathcal{C}_m^{wt}\left(\alpha(i), \beta(i), \delta_e = 0, \delta_{lef} = 0, q = 0\right) + \nu(k) \\ &=\; \mathcal{C}_m^{wt}\left(\alpha(i), \beta(i)\right) + \nu(k), \end{aligned} \tag{4.92}$$

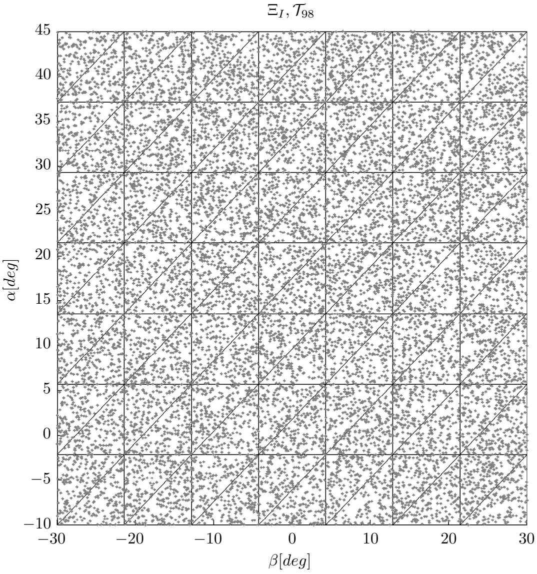


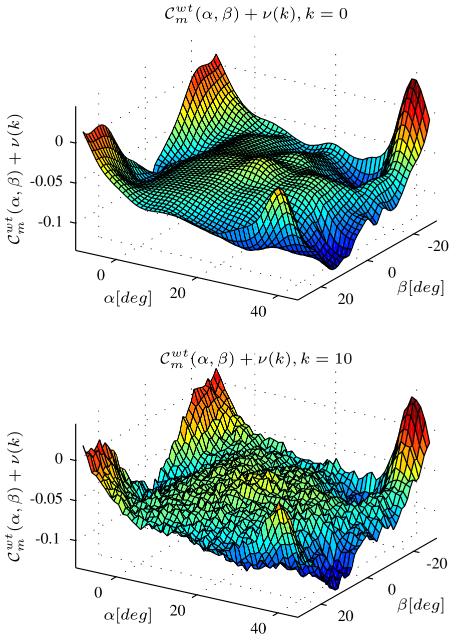
Figure 4.4: The identification dataset $\mathcal{X}_I$ and the triangulation $\mathcal{T}_{98}$

Table 4.1: Spline spaces

Continuity order	spline spaces	
-1	S_d^{-1}	$S_1^{-1}, S_2^{-1}, S_3^{-1}, S_4^{-1}, S_5^{-1}, S_6^{-1}$
0	S_d^0	$S_1^0, S_2^0, S_3^0, S_4^0, S_5^0, S_6^0$
1	S_d^1	$S_2^1, S_3^1, S_4^1, S_5^1, S_6^1$
2	S_d^2	$S_3^2, S_4^2, S_5^2, S_6^2$
3	S_d^3	S_4^3, S_5^3, S_6^3
4	S_d^4	S_5^4, S_6^4

with $C_m^{wt}(\alpha, \beta)$ the reduced (i.e. from 5-variate to bivariate) wind tunnel model, including the linear interpolation algorithm. The vector $\mathbf{Y}$ containing all measurements is then constructed row by row. Two surface plots of the bivariate wind tunnel model $C_m^{wt}(\alpha, \beta)$ together with the noise function $\nu(k)$ are drawn in Figure 4.5. This figure clearly illustrate the nonlinear nature of $C_m^{wt}(\alpha, \beta)$, as well as the effect of measurement noise on the smoothness of the dataset. For the spline model validation a new dataset $\mathcal{X}_V$ is created. This validation dataset consists of 50000 (α, β) points which are distributed randomly, and independently from $\mathcal{X}_I$, over the flight envelope.

A total of 26 different bivariate spline spaces are used to approximate $\mathbf{Y}$, see Table 4.1. The triangulation used in the experiment was a Type I triangulation consisting of 98


$\mathcal{C}_m^{wt}(\alpha,\beta) + \nu(k), k = 0$

$\mathcal{C}_m^{wt}(\alpha,\beta) + \nu(k), k = 10$

Figure 4.5: The data generating function $\mathcal{C}_m^{wt}(\alpha,\beta) + \nu(k)$ for $k = 0$ (top) and $k = 10$ (bottom)

Table 4.2: Triangulation

Triangulation	simplex count	min data	max data
$\mathcal{T}_{98}$	98	123	188

simplices, see Table 4.2. The B-coefficients of the in total 208 spline functions are estimated using the ECGLS estimator from Sec. 4.2.1. In all cases, the ECGLS estimator was employed as a post-processing step after ECOLS parameter estimation, as explained in Sec. 4.3.

In the remainder of this section 2 of these 208 spline functions were put through the complete model validation process from Sec. 4.3. These two selected spline functions are $s \in \mathcal{S}_5^0$ and $s \in \mathcal{S}_5^2$. These two spline functions were selected, as they produced very different model validation results, even though they were both 5^{th} degree spline functions. As it will turn out, the continuity order of a spline space is a very important factor for the quality of a spline function.

4.4.2 Model residue analysis

The model residual analysis is the analysis of the difference between the validation observations and the spline model output at the validation locations. The model residual analysis is performed using the theory from Sec. 4.3.1.

In Figure 4.6 the relative residual RMS of the spline functions is plotted for four different values of the noise factor k. When no measurement noise is present (i.e. $k = 0$), the lowest RMS values are found for spline functions of high degree, and low continuity order. When noise is present, however, spline functions with low continuity orders produce significantly lower quality results than splines with a high continuity order. This can be explained as follows. Splines with a low continuity order have a higher approximation power than splines of higher continuity order. On noise-free datasets, low continuity splines will always outperform high continuity splines, see for example the top left figure in Figure 4.6. When noise is present in the identification dataset, the low continuity order splines have the approximation power to fit the measurement noise. Because the noise is random, this will lead to larger than normal errors during validation. The low continuity spline functions will in that case be outperformed by high continuity order splines. In Figure 4.6 this can be seen in the bottom right plot, where the $s \in \mathcal{S}_4^2$ (left) clearly outperforms the $s \in \mathcal{S}_5^{-1}$ spline, completely reversing the situation in the top left plot ofFigure 4.6.

The relative residual RMS is a global statistic; it does not provide any information about the local modeling quality of the spline spaces. For this reason four individual spline functions are plotted for the two noise intensities $k = 0$ and $k = 10$. In Figure 4.7 the spline functions $s \in S_5^0$ and $s \in S_5^2$ respectively are plotted for the zero noise case ($k = 0$). According to Figure 4.6, the C^0 continuity order spline function has a residual RMS that is more than half an order of magnitude smaller than the residual RMS of the C^2 continuity order spline function. In Figure 4.8 the same spline spaces are again plotted, but now for the $k = 10$ noise case. The left hand figure shows that the spline space S_5^0 is actually modeling the noise term $\nu(k)$, which is apparent from the noisy residual plot. The spline space S_5^2 on the other hand is relatively insensitive to noise, its residual is quite similar to the residual for the zero noise case.

4.4.3 Statistical model quality analysis

The B-coefficient variances are estimated based on the methodology presented in Sec. 4.3. A useful statistical measure for assessing the global quality of a spline model is the logarithm of the mean of the variances of all B-coefficients in a single spline function as follows:

$$\log(\overline{Var(\hat{\mathbf{c}})}) = \log\left(\frac{1}{(T \cdot \hat{d})}\sum_q Var(\hat{\mathbf{c}})_q\right), \tag{4.93}$$

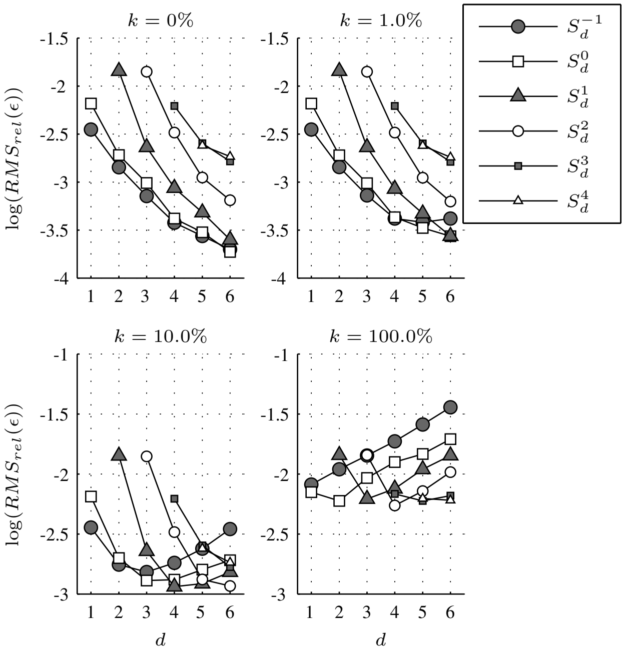
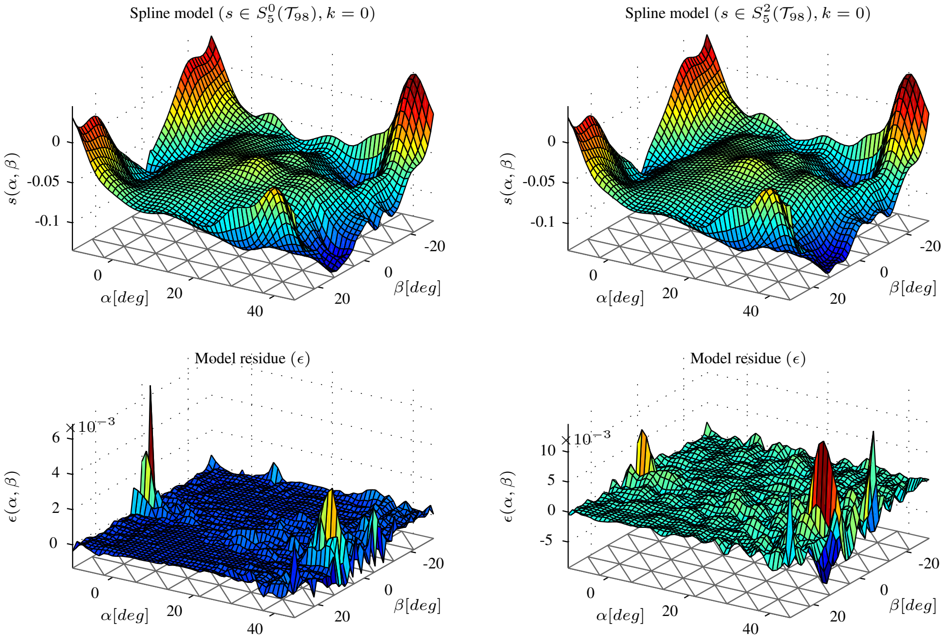


Figure 4.6: The relative residual RMS resulting from model validation, as a function of spline space and noise level on the triangulation $\mathcal{T}_{98}$

with $Var(\hat{\mathbf{c}})$ as in Eq. 4.83, T the number of simplices, and $\hat{d}$ the number of B-coefficients in a single simplex. In Figure 4.9 this statistical measure is plotted as a function of spline degree and continuity for the four different noise levels. A clear pattern emerges in this figures; for a given spline degree, the mean of the B-coefficient variances for low continuity order splines is higher than for high continuity order splines, especially when noise is present. The spline space that for a given degree d best fits the data according to Figure 4.9 is the S_d^{-1} spline space. This spline space, however, has the highest mean B-coefficient variance according to Figure 4.9.

The individual values of estimated B-coefficient variances for the two 5^{th} degree spline functions $s \in \mathcal{S}_5^0$ and $s \in \mathcal{S}_5^2$ are plotted in Figure 4.10 and Figure 4.11 for two different measurement noise values. The zero-noise case is shown in Figure 4.10 while the variances for the 10% noise case are shown in Figure 4.11. These two figures confirm the global statistics from Figure 4.9, that is, the $s \in \mathcal{S}_5^0$ spline function is much more susceptible to measurement noise than the $s \in \mathcal{S}_5^2$ spline function.

The spatial location of the B-coefficients allows the variances of individual B-coefficients to be visualized. In the left half plots in Figure 4.12 the B-coefficient variance surfaces are drawn for the S_5^0 and S_5^0 spline spaces. These plots show that the global statistical measure from Eq. 4.93 is just that; a global measure. The variance surfaces in Figure 4.12 clearly


Spline model ($s \in S_5^0(\mathcal{T}_{98}), k = 0$)

Spline model ($s \in S_5^2(\mathcal{T}_{98}), k = 0$)

Model residue (ϵ)

Model residue (ϵ)

Figure 4.7: Two 5^{th} degree spline functions with continuity order C^0 (left) and C^2 (right) and their validation residuals for the zero-noise case $(k = 0)$

show local fluctuations. The variances of the B-coefficients in the neighborhood of the edges of the flight envelope show much larger values than in the inner regions of the flight envelope. Note that the B-coefficient variance surface of the S_5^0 spline resembles the model residue from Figure 4.7, while the S_5^0 spline has a more flattened variance surface.

In the right hand plots of Figure 4.12 the B-coefficient variance surfaces of both spline functions are drawn for the $k = 10$ noise case. This time there is a striking difference in the structure of the variance surfaces of the two spline spaces. The variance surface of the S_5^0 spline is highly erratic, while the variance surface of the S_5^2 spline is very smooth. Effectively, the continuity equations cause a sharing of information between simplices which reduces B-coefficient variances. In fact, the B-coefficients of the S_5^2 spline with the highest variance are B-coefficients that are located on the edge of the triangulation. These B-coefficients are subject to less continuity constraints than B-coefficients that are located away from the triangulation edges. The variance surfaces in Figure 4.12 also help to explain the noisy nature of the S_5^0 spline in Figure 4.8; high parameter variances lead to high noise sensitivity.

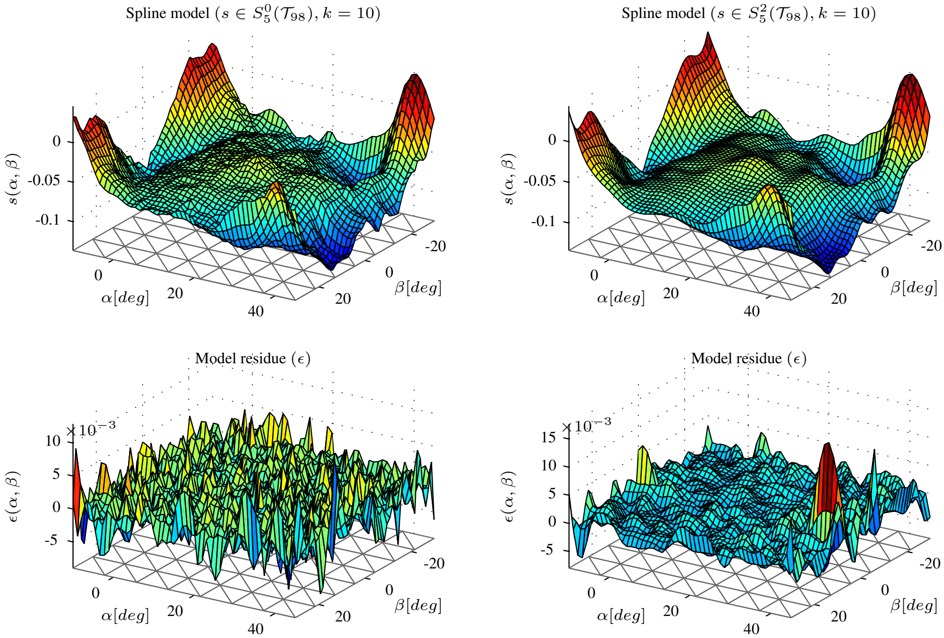


Figure 4.8: Two 5^{th} degree spline functions with continuity order C^0 (left) and C^2 (right) and their validation residuals for the 10%-noise case $(k = 10)$

4.4.4 Stability analysis

Using the method from Sec. 4.3.3, the stability of the two 5^{th} degree spline models was tested. The results from these tests are shown in Figure 4.13 and Figure 4.14. In Figure 4.13 the B-coefficient values for the zero-noise case are shown. From this figure, it can be seen that both the $s \in \mathcal{S}_5^0$ and the $s \in \mathcal{S}_5^2$ spline functions have B-coefficient values that are relatively close. The black lines in the figures are the bounds of the identification dataset (i.e. $\max Y / \min Y$) which are, as they should be, contained within the hull of the B-coefficients.

For the 10% measurement noise case, things start to change, see Figure 4.14. This time, the B-coefficients of the spline function with C^0 continuity behave much more erratic, while the B-coefficients of the spline function with continuity order C^2 show hardly any difference with the zero-noise case. An explanation for this result will have to wait until the statistical model quality analysis.

From the results Figure 4.13 and Figure 4.14 it can safely be concluded that both the $s \in \mathcal{S}_5^0$ as well as the $s \in \mathcal{S}_5^2$ spline function are stable and well defined within the spline domain.

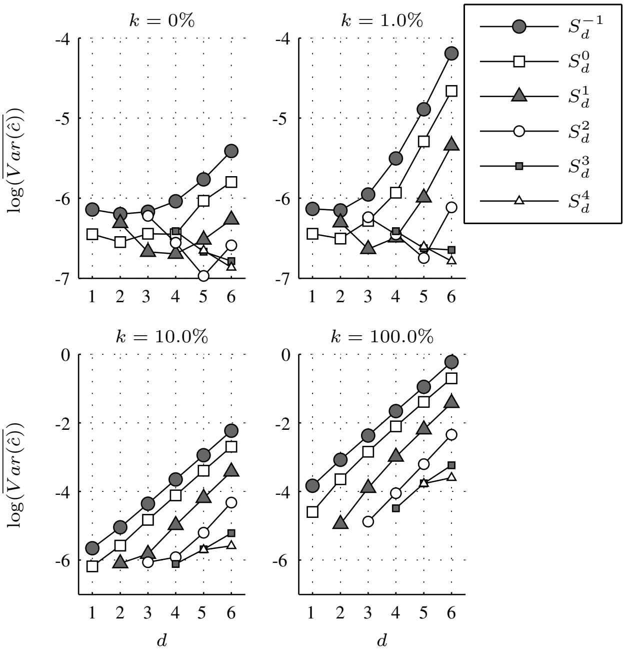


Figure 4.9: The mean B-coefficient variances as a function of the spline space and noise level.

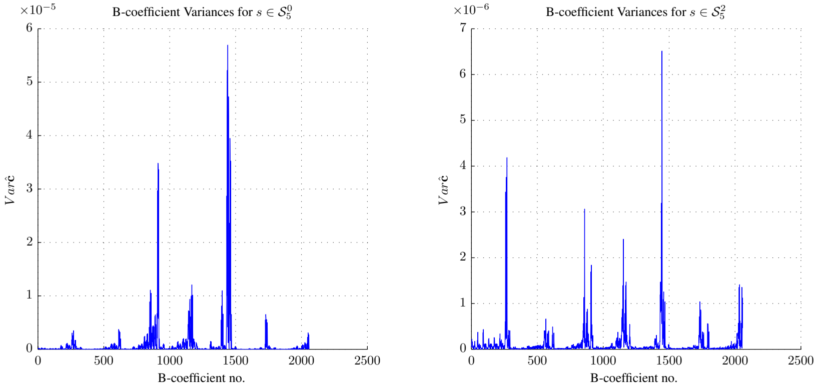


Figure 4.10: Individual B-coefficient variances of the $s \in \mathcal{S}_5^0$ spline (left) and the $s \in \mathcal{S}_5^2$ spline (right) for the zero-noise case ($k = 0$).

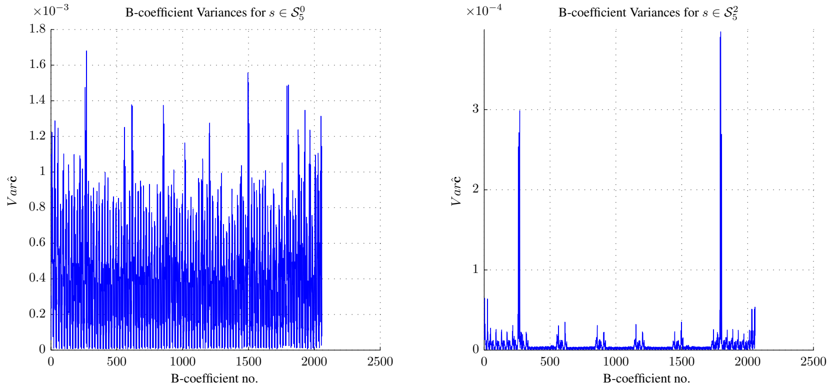


Figure 4.11: Individual B-coefficient variances of the $s \in \mathcal{S}_5^0$ spline (left) and the $s \in \mathcal{S}_5^2$ spline (right) for the 10%-noise case ($k = 10$).

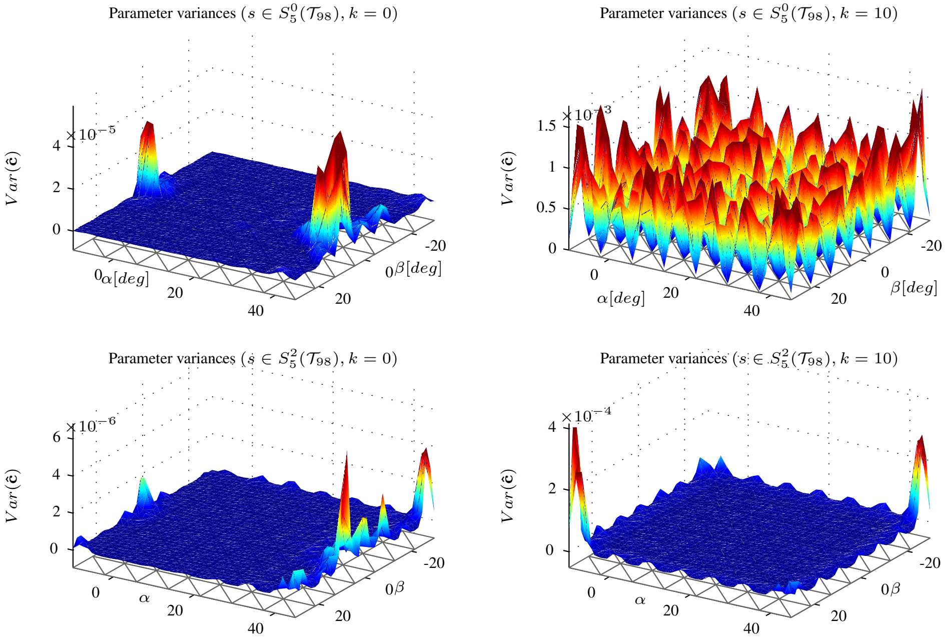


Figure 4.12: B-coefficient variance surfaces for the S_5^0 (top left) and S_5^2 (bottom left) spline functions for the $k = 0$ noise case, and S_5^0 (top right) and S_5^2 (bottom right) for the $k = 10$ noise case.

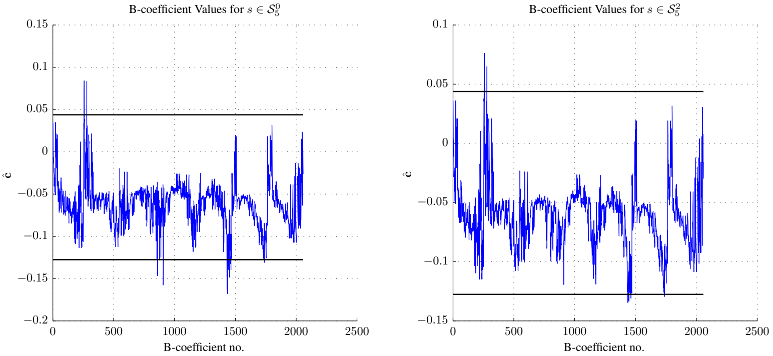


Figure 4.13: B-coefficient values for $s \in \mathcal{S}_5^0$ (left) and $s \in \mathcal{S}_5^2$ (right) for the zero-noise case. The black lines are the $\max Y / \min Y$ bounds.

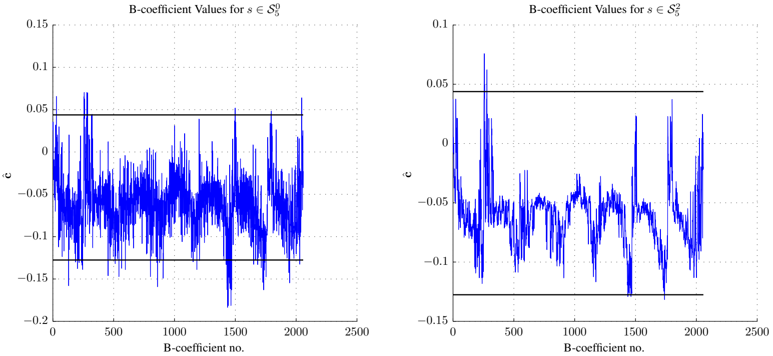
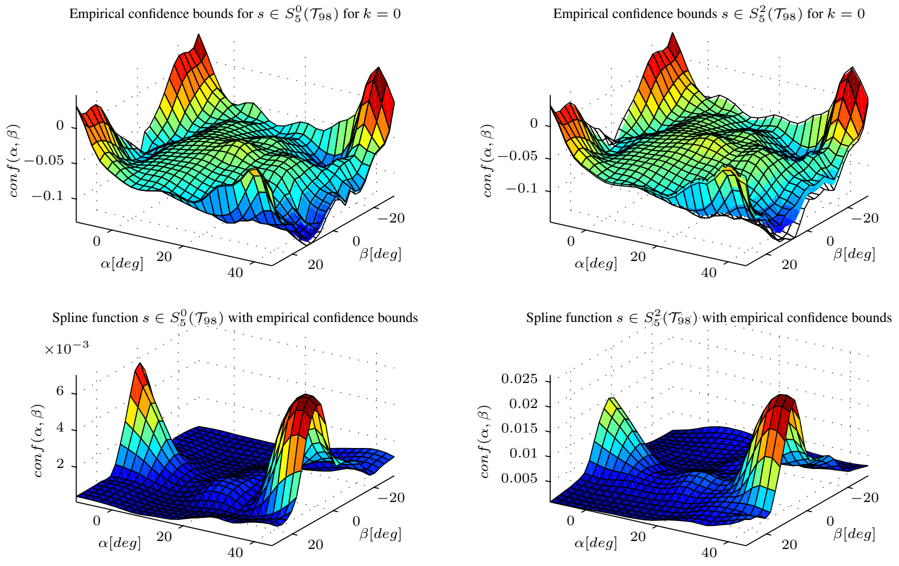


Figure 4.14: B-coefficient values for $s \in \mathcal{S}_5^0$ (left) and $s \in \mathcal{S}_5^2$ (right) for the 10%-noise case. The black lines are the $\max Y / \min Y$ bounds.

4.4.5 Empirical confidence bounds

The method for creating empirical confidence bounds was presented in Sec. 4.3.1. In this paragraph this method will be used to define empirical confidence bounds for the two spline functions $s \in \mathcal{S}_5^0$ and $s \in \mathcal{S}_5^2$ which should contain at least 97% of all identification data points. Using Chebyshev's inequality from Eq. 4.81 the 97% inclusion level requires 6σ confidence bounds.

In Figure 4.15 the 6σ empirical confidence bounds of both 5^{th} degree spline functions for the zero-noise case are shown. From this figure it can be seen that the empirical confidence model for $s \in \mathcal{S}_5^0$ is an order of magnitude tighter than the empirical confidence model for $s \in \mathcal{S}_5^2$.


Empirical confidence bounds for $s \in S_5^0(\mathcal{T}_{98})$ for $k = 0$

Empirical confidence bounds $s \in S_5^2(\mathcal{T}_{98})$ for $k = 0$

Spline function $s \in S_5^0(\mathcal{T}_{98})$ with empirical confidence bounds

Spline function $s \in S_5^2(\mathcal{T}_{98})$ with empirical confidence bounds

Figure 4.15: 6σ Empirical confidence model for $s \in \mathcal{S}_5^0$ (left) and the empirical confidence model for $s \in \mathcal{S}_5^2$ (right) for the zero noise case ($k = 0$). The top row shows the spline models together with the confidence bounds.

If the noise level is increased to the 10% level, the 6σ confidence bounds widen such that 97% of the identification dataset remains within the confidence bounds. In Figure 4.16 the confidence bounds for both spline functions $s \in \mathcal{S}_5^0$ and $s \in \mathcal{S}_5^2$ are shown. From this figure it is clear that both confidence intervals are of equally wide magnitudes. The confidence interval for $s \in \mathcal{S}_5^0$, however, is much flatter than that of $s \in \mathcal{S}_5^2$. This means that the white noise does not significantly impact the ability of $s \in \mathcal{S}_5^0$ to accurately fit the data, but does lead to small scale fluctuations over the complete spline domain.

4.4.6 Statistical confidence bounds

The B-coefficient variances are used to construct Bonferroni confidence intervals for the B-coefficients of the multivariate spline functions using the theory in Sec. 4.3.2. In Figure 4.17 the 99% Bonferroni confidence models of the primary spline functions S_5^0 and S_5^2 are shown. This figure shows a completely different picture than that for the empirical confidence bounds in Sec. 4.4.5. More specifically, it can be seen in the top right plot of Figure 4.17 that the statistical confidence bounds for the $s \in \mathcal{S}_5^0$ spline function are very sensitive to the increased (10%) noise level. Not only are the bounds twice as wide as the bounds for $s \in \mathcal{S}_5^2$, but their overall value is more or less constant across the spline domain. The Bonferroni confidence bounds for $s \in \mathcal{S}_5^2$ on the other hand are much smoother, and attain their highest magnitude at the edges of the spline domain.

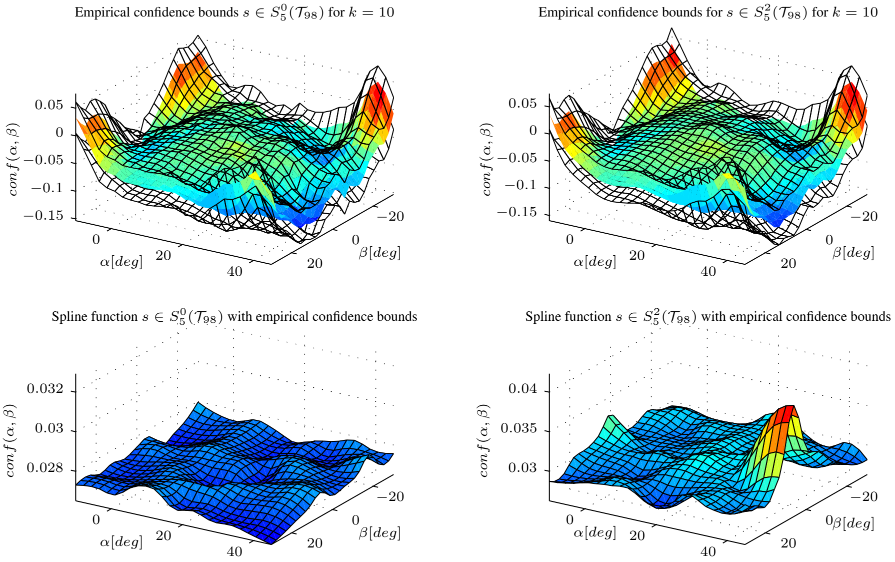
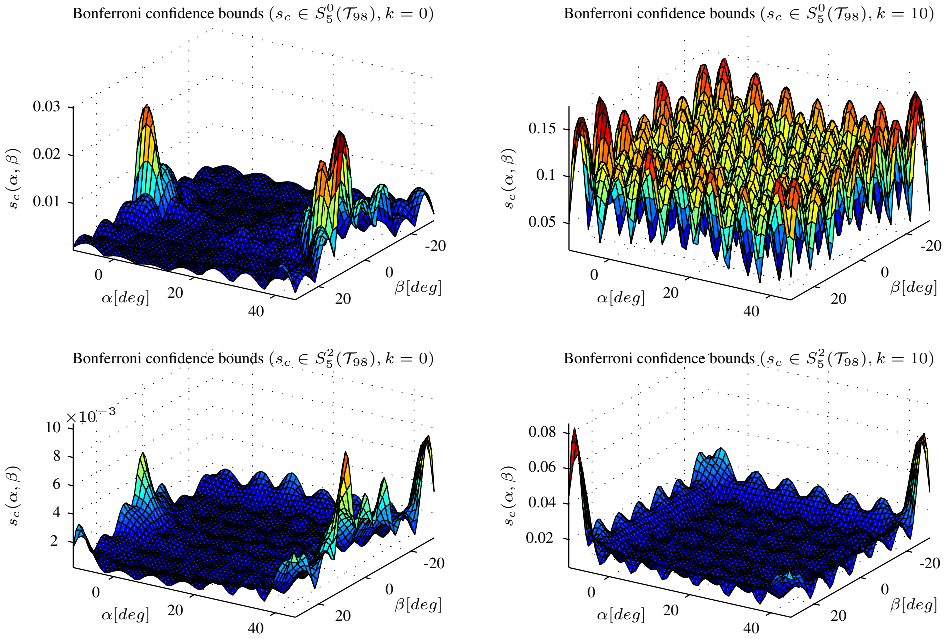


Figure 4.16: 6σ Empirical confidence model for $s \in \mathcal{S}_5^0$ (left) and the empirical confidence model for $s \in \mathcal{S}_5^2$ (right) for the 10% noise case ($k = 10$). The top row shows the spline models together with the confidence bounds.


Bonferroni confidence bounds ($s_c \in S_5^0(\mathcal{T}_{98}), k = 0$)

Bonferroni confidence bounds ($s_c \in S_5^0(\mathcal{T}_{98}), k = 10$)

Bonferroni confidence bounds ($s_c \in S_5^2(\mathcal{T}_{98}), k = 0$)

Bonferroni confidence bounds ($s_c \in S_5^2(\mathcal{T}_{98}), k = 10$)

Figure 4.17: Four different statistical confidence models for the zero-noise case (left) and the 10% noise case (right) for the two spline functions $s \in \mathcal{S}_5^0$ spline (top) and the $s \in \mathcal{S}_5^2$ (bottom).

Chapter 5

Aerodynamic Modeling of the Cessna Citation II

Detailed aerodynamic models play a crucial role in the design of flight simulators as well as flight control systems. Identifying and validating aerodynamic models involves the synthesis of large databases containing wind tunnel data and data obtained during flight testing [159, 162, 161]. Currently, the most widely used method for identifying aerodynamic models uses a parameter estimation method like least squares or maximum likelihood to estimate the parameters of a polynomial regression model, see e.g. [81, 90, 147, 88, 138, 144, 142, 92, 84, 125]. It is well known that polynomials have a limited approximation power, which is directly proportional with their degree. This in turn limits the application of polynomial based aerodynamic models to aircraft with small flight envelopes and limited nonlinear aerodynamics. More complex models can be created, however, by creating local polynomial models on partitions of the flight envelope. The set of local polynomial models can then be blended into a single smooth structure with for example fuzzy blending techniques [8, 122] or neural networks [115, 72, 44, 84]. However, these blending methods suffer from a number of important drawbacks such as loss of transparency, loss of the linear-in-the-parameter property of the polynomial models, and the human intervention required in the blending operation[8].

Many authors have therefore suggested the use of polynomial spline functions for fitting flight data [198, 89, 10, 92]. Spline functions are piecewise defined polynomials with a predefined continuity order between polynomial pieces. The approximation power of spline functions is proportional with the degree of the polynomial but also with the number and density of the polynomial pieces. Polynomial spline functions are therefore capable of fitting highly nonlinear datasets over large domains.

While one-dimensional (i.e. univariate) spline theory is well known and developed, multi-dimensional (i.e. multivariate) spline theory is still an active research field. Many different multivariate spline types exist, such as the thin plate splines [216, 217], polyhedral splines [25], and the well-known tensor product B-spline [16]. Most of these multivariate spline types have been proved to be difficult to work with in practice [55, 152]. In the past, however, multivariate tensor product splines have been used successfully for the modeling of aircraft aerodynamics using gridded wind tunnel data [198, 89, 88, 10]. The great downside of tensor product splines, however, is that they are, for fundamental mathematical reasons, incapable of fitting scattered data [16, 3]. This means that tensor product splines cannot be used to model flight test data, which is inherently scattered. More recently, a new type of multivariate spline called the multivariate simplex spline was introduced [7, 99, 40]. The multivariate simplex splines are capable of fitting nonlinear, multi-dimensional scattered data and have an arbitrarily high approximation power, which makes them especially suited for the modeling of aircraft aerodynamics based on flight data.

In this chapter, the new methodology for global model identification based on multivariate simplex splines introduced in Chpt. 4 is applied inside the Two-Step method for aerodynamic model identification [64, 146, 147, 149] to identify global models for the aerodynamic force and moment coefficients of the Cessna Citation II laboratory aircraft. The Two-Step method decomposes the output-error problem of combined state and parameter estimation into a separate state estimation and aerodynamic model parameter estimation problem. This in turn allows the use of the linear regression method from [40] for estimating the parameters of the simplex spline based aerodynamic models. The data for the model identification process was real flight data, which was obtained during seven test flights with the Cessna Citation II.

A phenomenological modeling approach will be taken in the aerodynamic model identification process. The phenomenological approach differs from the physical modeling approach only during the model structure selection phase. The physical model approach uses model structures that can be directly related to well established methods from the literature. The phenomenological model approach, on the other hand, does not exclude any model structure beforehand. Instead, it simply tries to find the model structure that results in the highest quality model. It will be shown in this chapter that the phenomenological model approach leads to sometimes surprising model structures that can nevertheless be proven to be of excellent quality.

This chapter is organized as follows. First, in Sec. 5.1 the subject of the identification experiment, the Cessna Citation II and its advanced flight test instrumentation system will be introduced. Then, in Sec. 5.2 the design of the flight test maneuvers is presented. The results from the flight tests then is presented in Sec. 5.3. The first step in the Two-Step method for aerodynamic model identification is flight path reconstruction which is presented in Sec. 5.4. Finally, in Sec. 5.5 the new simplex spline based system identification method is applied to the reconstructed flight data for the estimation of multivariate simplex spline based aerodynamic models.

5.1 Introduction

In this section, a high level overview of the procedure for aerodynamic model identification is presented in Sec. 5.1.1. Following this, the main subject of the identification experiment, the Cessna Citation II laboratory aircraft is introduced in Sec. 5.1.2.

5.1.1 A procedure for AMI with simplex splines

In Figure 5.1 a high level overview of the complete aerodynamic model identification (AMI) procedure is given. The bracketed numbers in the figure refer to the sections of this chapter in which that particular element receives an in-depth discussion.

The complete procedure for aerodynamic model identification consists of three distinct parts, the first of which is the ‘Data Acquisition’ part. The goal of data acquisition is the creation of a set of flight data which is suitable for aerodynamic model identification. This implies that the subject aircraft must be sufficiently excited with specially designed input sequences [147, 84]. In the ideal case, the data should be acquired from hundreds of special flight test maneuvers scattered through the complete flight envelope. Operating an aircraft is expensive, however, and in the framework of this thesis only three flights that were at least partially dedicated to aerodynamic model identification took place. Despite this, much more flight data was available from earlier flights with the laboratory aircraft. In most cases, however, this data was logged using different formats and standards. Data mining resulted in four additional usable flights of which one was completely dedicated to aerodynamic model identification. The total dataset after flight testing and data mining consisted of 247 maneuvers. This raw flight data forms the input to the next part of the aerodynamic model identification procedure, which is the data pre-processing part.

The data pre-processing part is actually the first step in the Two-Step method for aerodynamic model identification [147, 149]. The first element in this step is the pre-processing of raw data, which is aimed at removing data corruptions and sensor glitches, resulting in coherent datasets. The next element is flight path reconstruction (FPR), which is aimed at estimating the true states of the aircraft from the biased and noise contaminated observations [149]. The results from the flight path reconstruction are the true aircraft states, which can now be used in the actual aerodynamic model identification part in Figure 5.1.

The final part of the procedure is the aerodynamic model identification part, which is the second step in the Two-Step method, see Figure 5.1. This part of the procedure is aimed at the identification of a high fidelity aerodynamic model based on multivariate simplex splines. The first element in this part is the simplex spline model estimator, which contains

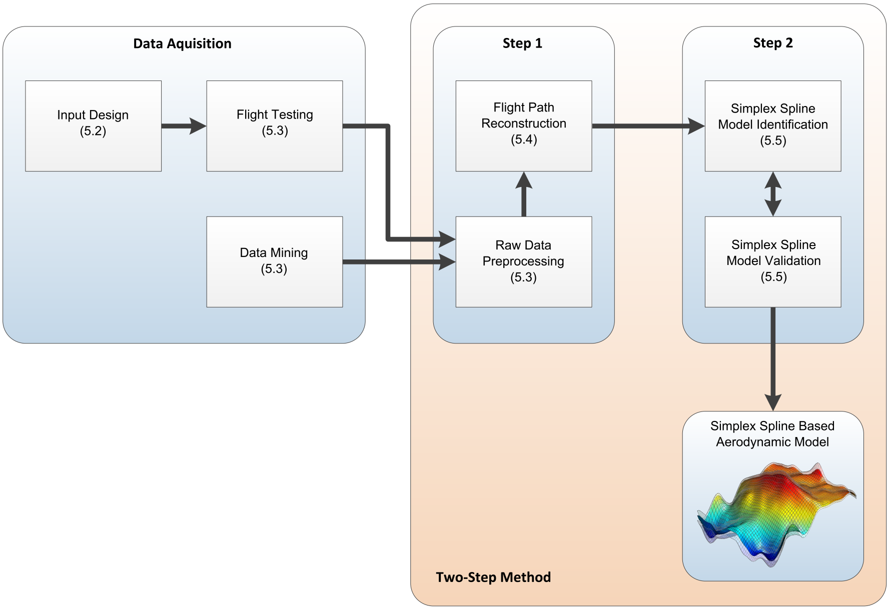

Data Aquisition
Input Design (5.2)
Flight Testing (5.3)
Data Mining (5.3)
Step 1
Flight Path Reconstruction (5.4)
Raw Data Preprocessing (5.3)
Step 2
Simplex Spline Model Identification (5.5)
Simplex Spline Model Validation (5.5)
Simplex Spline Based Aerodynamic Model
Two-Step Method

Figure 5.1: A high level overview of the complete aerodynamic model identification procedure.

of the model structure selection and parameter estimation procedures. The model structure selection and parameter estimator procedures were discussed in-depth in Chpt. 4. Both procedures are based on the theory presented in Chpt. 2 and Chpt. 3. The identified spline model is then subjected to the various different model validation techniques discussed in Chpt. 4. Depending on the results from the model validation, either a new model with a different model structure is created or the resulting spline model is returned, ending the procedure.

Now that the procedure for aerodynamic model identification is clear, a more exact analysis on the details of the Cessna Citation II laboratory aircraft and its flight test instrumentation system is required. These details for a large part determine the implementation of the various elements of the procedure for aerodynamic model identification in Figure 5.1.

5.1.2 The Cessna Citation II and its FTIS

The subject of the aerodynamic model identification procedure presented in the previous paragraph is the Cessna Citation II laboratory aircraft operated by the Delft University of Technology and the Netherlands National Aerospace Laboratory (NLR), see Figure 5.2. The call sign of the aircraft is PH-LAB, and in the following this call sign will frequently be used as an abbreviation.

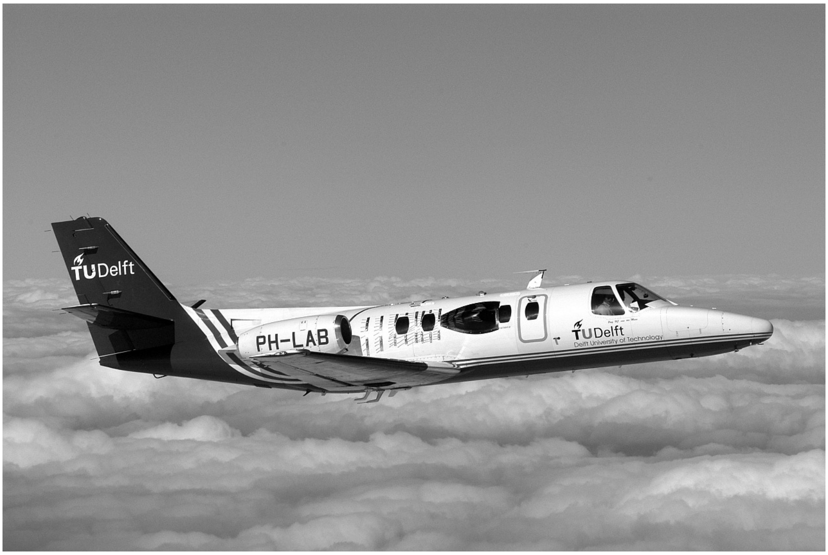


Figure 5.2: The Cessna Citation II laboratory aircraft with call sign PH-LAB.

The PH-LAB is a twin-jet business aircraft designed and constructed by the aircraft manufacturer Cessna. It is equipped with two Pratt & Whitney JT15D-4 turbofan engines each having a static thrust of 11.12 KN. The maximum operating altitude of the aircraft is 13100 meters (43000 ft) and the maximum cruising speed is 207 m/s (403 kt). The aircraft is equipped with a modern flight test instrumentation system (FTIS) developed by the Delft University of Technology. At the heart of the FTIS system is a *dSPACE* computer which is connected to an advanced suite of inertial, GPS, control input, and air data sensors Figure 5.3. Additionally, the laboratory aircraft was recently equipped with a fly-by-wire (FBW) control system, allowing fully automatic, precise execution of preprogrammed maneuvers [219].

In Figure 5.3, the various components of the FTIS are highlighted. The heart of the FTIS is the data acquisition computer (DAC) which consists of a ruggedized Pentium PC. The DAC hosts a dSPACE real-time computer in one of its ISA slots. The dSPACE computer in turn interfaces with the various sensors and transducers. In Table 5.1 the various sensors and transducer components are listed, together with their outputs to the dSPACE computer.

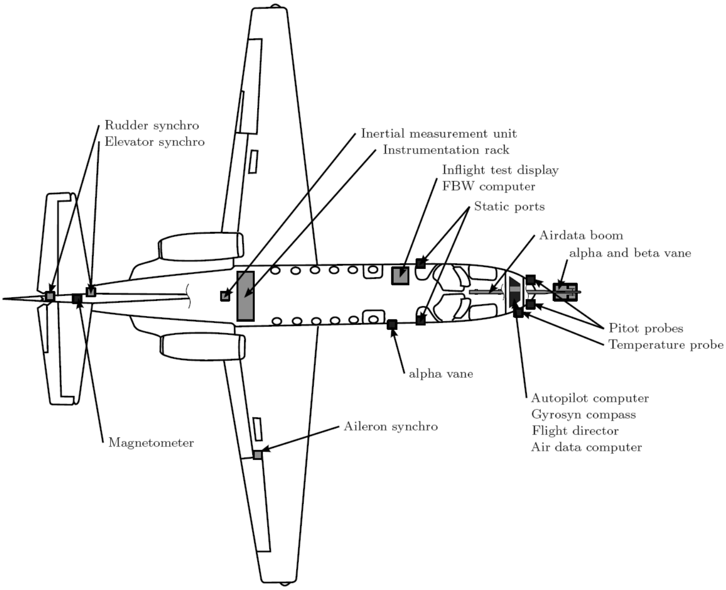


Figure 5.3: Cessna Citation II laboratory aircraft flight test instrumentation system

Table 5.1: Available

FTIS Component	Description	Output
Noseboom	Intrusive nose boom	α_v, β_v, $A_{y_{boom}}$, $A_{z_{boom}}$
α-vane	Stall warning device	α_w at the wing root
IMU	Inertial Measurement Unit	A_x, A_y, A_z, p, q, r
TARSYN	Three-Axis Reference SYNchro	θ, ϕ
DADC	Digital Air Data Computer	Mach, P$_{alt}$, IAS, CAS, TAS, TAT
FMS	Flight Management System	Navigational and Engine parameters
GPS	Global Positioning System	ECEF positions and velocities
Synchros	Measure control surface deflections	δ_a, δ_e, δ_r, δ_{t_e}

5.2 Flight Test Design

The ultimate goal of flight testing for aerodynamic model identification is the creation and validation of accurate aerodynamic models for aircraft. In order to create an aerodynamic model that is valid across the full flight envelope of an aircraft, flight testing would in the ideal case cover the complete flight envelope with test points. At every test point, a number of different maneuvers would be performed to investigate the longitudinal, lateral and cross-coupled dynamics of the aircraft. In the real world, however, such a strategy would be prohibitively expensive as it would require many thousands of test points and hundreds of flying hours. It is the goal of flight test design to maximize the yield of significant data while minimizing the required number of test points.

In this section the process of flight test design in the framework of aerodynamic model identification will be elaborated. First, in Sec. 5.2.1 the equations of motion for aircraft are presented. Using the equations of motion, the aerodynamic force and moment coefficients can be calculated, as is shown in Sec. 5.2.2. Following this, in section Sec. 5.2.4 an initial assumption on the model structure of the complete aerodynamic model will be presented. A list of the required parameters for defining the model can then be defined based on this initial assumption. In Sec. 5.2.4 the list of required parameters is then translated into a set of test points at which special flight test maneuvers need to be performed. Finally, in Sec. 5.2.5 the specific design of the control surface inputs resulting in these maneuvers is discussed.

5.2.1 Aircraft Equations of motion

The well-known equations of motion of an aircraft come in the form of three sets of first order differential equations [149, 203]. The first set of equations of motion are the equations for the total aerodynamic forces X, Y and Z working in the direction of the axes of the body fixed reference frame F_B (see Appendix A):

$$\begin{aligned} X &:= m\,(\dot{u} + qw - rv) + mg\sin(\theta), \\ Y &:= m\,(\dot{v} + ru - pw) - mg\cos(\theta)\sin(\phi), \\ Z &:= m\,(\dot{w} + pv - qu) - mg\cos(\theta)\cos(\phi), \end{aligned} \tag{5.1}$$

with ϕ, θ and ψ the Euler angles of rotation about F_B, with p, q and r denoting the rates of rotation about F_B, with u, v and w the translational velocities in the direction of F_B, and with $\dot{u}$, $\dot{v}$ and $\dot{w}$ the translational accelerations in the direction of F_B. The forces X, Y and Z include the aerodynamic effects of propulsion systems.

The second set of equations of motion are the equations for the total moments L, M, and N about the axes of F_B which for a symmetric aircraft configuration are defined as follows:

$$\begin{aligned}
L &:= J_x\dot{p} - (J_y - J_z)qr - J_{xz}(\dot{r} + pq), \\
M &:= J_y\dot{q} - (J_z - J_x)rp - J_{xz}(r^2 - p^2) + M_e, \\
N &:= J_z\dot{r} - (J_x - J_y)pq - J_{xz}(\dot{p} - qr),
\end{aligned} \tag{5.2}$$

with J_x, J_y, J_z and J_{xz} elements from the inertia matrix, which for an aircraft that is symmetric about the $X_b - Z_b$ plane is given by:

$$\mathbf{J} := \begin{bmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{bmatrix} \tag{5.3}$$

The term M_e in Eq. 5.2 is the gyroscopic effects of spinning turbines or propeller blades as follows:

$$M_e := J_e \omega_e r, \tag{5.4}$$

with J_e the moment of inertia in the $Y_B - Z_B$ plane and with ω_e the rotational velocity of the blades. It is assumed in the following that engine thrust induced moments are negligible. This means that both the differential thrust effects on the yawing moment, and engine induced pitching moments are not modeled in this thesis.

Finally, the third set of equations are the kinematic equations for the time rate of change of the Euler angles ϕ, θ and ψ:

$$\begin{aligned}
\dot{\phi} &= p + q\sin(\phi)\tan(\theta) + r\cos(\phi)\tan(\theta), \\
\dot{\theta} &= q\cos(\phi) - r\sin(\phi), \\
\dot{\psi} &= q\sin(\phi)\sec(\theta) + r\cos(\phi)\sec(\theta).
\end{aligned} \tag{5.5}$$

Now let A_x, A_y and A_z be the accelerations of an aircraft in the direction of the axes of F_B as follows:

$$\begin{aligned}
A_x &:= \frac{X}{m}, \\
A_y &:= \frac{Y}{m}, \\
A_z &:= \frac{Z}{m}.
\end{aligned} \tag{5.6}$$

These accelerations would be the accelerations measured by a set of perfect accelerometers located in the center of gravity of an aircraft. In the literature, the accelerations A_x, A_y and A_z are sometimes referred to as 'specific forces' as they are the total force per unit of mass [149].

Substitution of Eq. 5.6 in Eq. 5.1 then leads to the following set of equations:

$$
\begin{aligned}
\dot{u} &= A_x - qw + rv - g\sin(\theta), \\
\dot{v} &= A_y - ru + pw + g\cos(\theta)\sin(\phi), \\
\dot{w} &= A_z - pv + qu + g\cos(\theta)\cos(\phi).
\end{aligned}
\tag{5.7}
$$

5.2.2 Determination of the force and moment coefficients

The dimensionless aerodynamic force coefficients can be introduced into the force equations from Eq. 5.1 as follows:

$$
\begin{aligned}
\frac{1}{2}\rho V^2 S \cdot C_X &= m\dot{u} + m(qw - rv) + mg\sin(\theta), \\
\frac{1}{2}\rho V^2 S \cdot C_Y &= m\dot{v} + m(ru - pw) - mg\cos(\theta)\sin(\phi), \\
\frac{1}{2}\rho V^2 S \cdot C_Z &= m\dot{w} + m(pv - qu) - mg\cos(\theta)\cos(\phi),
\end{aligned}
\tag{5.8}
$$

with ρ the air density, with V the true airspeed, and with S the wing surface of the aircraft. Klein and Morelli also present an alternative representation for the aerodynamic force coefficients that directly relates the specific forces from Eq. 5.6, with the force coefficients [92]:

$$
\begin{aligned}
C_X &= \frac{mA_x}{\frac{1}{2}\rho V^2 S}, \\
C_Y &= \frac{mA_y}{\frac{1}{2}\rho V^2 S}, \\
C_Z &= \frac{mA_z}{\frac{1}{2}\rho V^2 S}.
\end{aligned}
\tag{5.9}
$$

In general, the accelerations measured by the IMU are not the true accelerations of the aircraft. Therefore, Eq. 5.9 will result in biased values for these coefficients. As will be explained in Sec. 5.4, however, the two-step method circumvents this problem by using state estimation techniques to estimate the true accelerations of the aircraft from the biased and noise contaminated accelerometer measurements. In that case it is actually valid to use Eq. 5.9.

The derivation of the aerodynamic moment coefficients is somewhat more complex than the derivation of the force coefficients because the equations for C_l and C_n are coupled in $\dot{p}$ and $\dot{r}$. In this thesis, the formulation from Klein and Morelli in [92] is used to derive the aerodynamic moment coefficients:

$$\begin{aligned}
\frac{\frac{1}{2}\rho V^2 Sb}{J_x} C_l &= \dot{p} - \frac{J_{xz}}{J_x}\dot{r} + \frac{J_z - J_y}{J_x} qr - \frac{J_{xz}}{J_x} qp, \\
\frac{\frac{1}{2}\rho V^2 S\bar{c}}{J_y} C_m &= \dot{q} + \frac{J_x - J_z}{J_y} pr + \frac{J_{xz}}{J_y}(p^2 - r^2) - \frac{M_e}{J_y}, \\
\frac{\frac{1}{2}\rho V^2 Sb}{J_z} C_n &= \dot{r} - \frac{J_{xz}}{J_z}\dot{p} + \frac{J_y - J_x}{J_z} pq + \frac{J_{xz}}{J_z} qr,
\end{aligned} \tag{5.10}$$

with ρ the air density, with V the true airspeed, with S the wing surface of the aircraft, $\bar{c}$ the wing chord, and with b the wing span of the aircraft.

5.2.3 Initial assumptions on model structure

The purpose of an aerodynamic model is to predict the aerodynamic forces and moments acting on an aircraft based on the current aircraft state. Because an aircraft is a six degree of freedom system, with three translational and three rotational degrees of freedom, the complete aerodynamic model consists of six sub-models for each of the individual degrees of freedom. The most general formulation of the complete aerodynamic model then is the following vector function:

$$F_{ae}(\mathbf{x}) = (C_X(\mathbf{x}), C_Y(\mathbf{x}), C_Z(\mathbf{x}), C_l(\mathbf{x}), C_m(\mathbf{x}), C_n(\mathbf{x})) \in \mathbb{R}^n. \tag{5.11}$$

Each of the individual components $F_{ae}(\mathbf{x})$ is itself a function of the aircraft state, and a specific aerodynamic force or moment as follows:

$$\begin{aligned}
C_X(\mathbf{x}) &= \frac{X(\mathbf{x})}{\frac{1}{2}\rho V^2 S}, \\
C_Y(\mathbf{x}) &= \frac{Y(\mathbf{x})}{\frac{1}{2}\rho V^2 S}, \\
C_Z(\mathbf{x}) &= \frac{Z(\mathbf{x})}{\frac{1}{2}\rho V^2 S}, \\
C_l(\mathbf{x}) &= \frac{L(\mathbf{x})}{\frac{1}{2}\rho V^2 S b}, \\
C_m(\mathbf{x}) &= \frac{M(\mathbf{x})}{\frac{1}{2}\rho V^2 S \bar{c}} \\
C_n(\mathbf{x}) &= \frac{N(\mathbf{x})}{\frac{1}{2}\rho V^2 S b}.
\end{aligned} \tag{5.12}$$

In Eq. 5.12, $X(\mathbf{x})$, $Y(\mathbf{x})$, and $Z(\mathbf{x})$ are the aerodynamic forces acting along the aircraft body reference frame axes X_B, Y_B, and Z_B, respectively (see Appendix A).

The corresponding models for these aerodynamic forces are $C_X(\mathbf{x})$, $C_Y(\mathbf{x})$, and $C_Z(\mathbf{x})$, which respectively are the longitudinal, lateral, and vertical force models. The aerodynamic moments $L(\mathbf{x})$, $M(\mathbf{x})$, and $N(\mathbf{x})$ are the moments about the aircraft body reference frame axes X_B, Y_B, and Z_B, respectively. The models for these moments then are $C_l(\mathbf{x})$, $C_m(\mathbf{x})$, and $C_n(\mathbf{x})$, which are the models for the rolling, pitching, and yawing moments. It is important to note that the models in Eq. 5.12 are dimensionless, which allows for the direct comparison between wind tunnel models and models created from flight test data [203].

From the general form Eq. 5.12, a more specific formulation can be derived by defining the contents of the state vector $\mathbf{x}$. This definition is in no way trivial, but most authors [147, 94, 203, 92, 84] agree that $\mathbf{x}$ is a vector with the following minimum structure:

$$\mathbf{x} = \Big(\alpha, \beta, \dot{\alpha}, \dot{\beta}, p, q, r, h, \text{Mach}, T_e, \delta\Big) \supseteq \mathbb{R}^{13}, \tag{5.13}$$

with h the (pressure) altitude, with T_e the combined engine thrust. The vector δ in Eq. 5.13 is a vector of control surface deflections, which for a conventional aircraft configuration consists of at least 3 components:

$$\delta = (\delta_a, \delta_e, \delta_r) \in \mathbb{R}^3 \tag{5.14}$$

The definition of the state vector from Eq. 5.13 implies that any of the models in Eq. 5.12 is a model of dimension 13 or higher. In Sec. 2.2 the complexity and counterintuitiveness of high dimensional spaces was discussed. For example, a 1-dimensional polynomial of degree 3 consists of 4 terms and 4 coefficients: $ax^3 + bx^2 + cx + d$. An equivalent 13-dimensional polynomial of degree 3, on the other hand, consists of 560 terms and coefficients. Clearly, a 13-dimensional model is not 13 times more complex[1] than a 1-dimensional model, but 13^k times more complex. The cause of this is that the full 13-dimensional polynomial not only matches the approximation power of the 1-dimensional polynomial along each of the 13 coordinate axes, but also at every possible cross-coupling, or combination, between these axes.

It is now very important to observe that the approximation power of the 13-dimensional model in the example above is exactly equal to that of the 1-dimensional model, because both models are 3^{rd} degree models. The only difference between these two models is that the 13-dimensional model offers this approximation power in $\mathbb{R}^{13}$, while the 1-dimensional model is limited to $\mathbb{R}^1$.

Models of dimension 10 or higher are rarely used in practice because of the extreme complexity of their structure, and the very high required data volumes required for their successful estimation, see Sec. 4.2. Luckily, there is an escape to this curse of dimensionality. The 13-dimensional model in the example achieves the full approximation power of the polynomial at every location in $\mathbb{R}^{13}$. However, in most cases this full approximation power is not required at every point. In general, this implies that not all dimensional cross-couplings have to be present in a model for it to be a member of $\mathbb{R}^n$. The trick is to split up the complete n-dimensional model into a number of sub-models of lower dimension, where *multilinear* combinations of the lower dimensional models together provide a membership in $\mathbb{R}^n$. In mathematical terms, this means the following:

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &\supseteq f(x_1, x_2, \dots, x_{n-1}) \cdot x_n \supseteq \\ &\supseteq f(x_1, x_2, \dots, x_{n-2}) \cdot x_{n-1} x_n \supseteq \cdots \\ &\supseteq x_1 x_2 \cdots x_{n-1} x_n \quad \in \mathbb{R}^n \end{aligned} \tag{5.15}$$

In practical term, this means that models of very high dimensionality can be produced by forming multilinear combinations of models of lower dimensions.

Example 22 (Full dimensionality vs. multilinear combinations)**.** *In this example the difference between a model that fully spans $\mathbb{R}^5$ and a model that spans $\mathbb{R}^5$ through multilinear combinations.*

First, let $\mathbf{x}$ be a vector in $\mathbb{R}^5$ as follows:

$$\mathbf{x} = (\alpha, \beta, p, q, r) \in \mathbb{R}^5$$

---

[1] Complexity in this context is considered to be the total number of coefficients and polynomial terms.

Then define two sub-vectors of $\mathbf{x}$ *as follows:*

$$\mathbf{x}_1 = (\alpha, \beta) \in \mathbb{R}^2, \;\; \mathbf{x}_2 = (p, q, r) \in \mathbb{R}^3$$

Now any function that forms a linear combination of $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *is a function in* $\mathbb{R}^5$*:*

$$\begin{aligned} f(\mathbf{x}_1, \mathbf{x}_2) &= g(\mathbf{x}_1) + h(\mathbf{x}_2) \in \mathbb{R}^5 \\ &= \alpha^2\beta + pqr \end{aligned} \tag{5.16}$$

Now consider the difference with the following function:

$$\begin{aligned} f(\mathbf{x}_1, \mathbf{x}_2) &= k(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}^5 \\ &= (\alpha + \beta + p + q + r)^3 \end{aligned} \tag{5.17}$$

While Eq. 5.16 spans only a small subset of $\mathbb{R}^5$*, Eq. 5.17 actually forms a polynomial basis for it. This means that Eq. 5.16 is a (very) small subset of Eq. 5.17 as the complete expansion of* $(\alpha + \beta + p + q + r)^3$ *consists of* 56 *polynomial terms. But both models are true* 5*-dimensional models!*

Returning to the general form of the six aerodynamic models in Eq. 5.12, it is clear that measurements on the aerodynamic forces and moments need to be performed at different values of the state vector from Eq. 5.13 in order to obtain a dataset using which an aerodynamic model can be identified. At this point it is not clear at what locations these measurements should be taken or how many should be made. In order to aid this process, a new concept is introduced in the form of the *state-plane*. A state plane has the following definition:

Definition 4. ***The state plane*** *A state-plane is an* m*-plane embedded in* n*-dimensional state space, with* $m \leq n$*, in which the* m *dimensions are fully cross-coupled.*

□

The total number of measurements required for aerodynamic model identification depends on the nonlinearity of the aerodynamics and the size of the flight envelope of an aircraft. For an aircraft with limited nonlinear aerodynamics, multilinear combinations of sub-models of lower dimension suffice. In that case, the complete state space can be decomposed into a number of state-planes of lower dimension, where measurements are only required at certain points within these planes. For an aircraft with significant nonlinear behavior, the state-planes need to be of higher dimension, which means that a much larger number of measurements are required for model identification.

Returning to the case of the Cessna Citation II, it is expected that significant nonlinear (coupled) aerodynamics are present in the $(\alpha, \beta, \dot{\alpha}, \dot{\beta}, p, q, r)$ plane. Moderate nonlinearities are expected in the (Mach, $h, \delta_a, \delta_e, \delta_r, T_e$) plane. The decoupling of aircraft dynamics

Table 5.2: State-planes for coupled, longitudinal, and lateral dynamics.

Dynamics	Significant Nonlinearities	Moderate/limited Nonlinearities
Coupled	$(\alpha, \beta, \dot{\alpha}, \dot{\beta}, p, q, r)$	$(\text{Mach}, h, \delta_a, \delta_e, \delta_r, T_e)$
Symmetric	$(\alpha, \dot{\alpha}, q)$	$(\text{Mach}, h, \delta_e, T_e)$
Asymmetric	$(\beta, \dot{\beta}, p, r)$	$(\text{Mach}, h, \delta_a, \delta_r)$

into symmetric and asymmetric dynamics is a method of further reducing the dimensionality of the state planes. This method of dimensional reduction is frequently used in the aerospace community to reduce the complexity of the aerodynamic sub-models.

In Table 5.2 the various state planes for the coupled and decoupled dynamics are listed. It is the aim of the flight test maneuvers to cover the nonlinear state-planes as completely as possible. This is achieved through the execution of maneuvers specifically designed to excite the variables that form a basis for this state-plane. The remaining state-planes are expected to contain only minor nonlinearities, and therefore require fewer measurements. This effectively means that only a small number of so-called *test points* along these dimensions suffice to cover the state-planes of limited nonlinearity. The result in practical terms is that no high density coverage of for example the Mach[2] dimension or the altitude dimensions is required. This is convenient because it removes the requirement for flight test maneuvers with very large velocity and altitude amplitudes.

The final topic to be treated in this paragraph is the actual availability of measurements on the components of the state vector from Eq. 5.13. The FTIS system of the Cessna Citation was already discussed in-depth in Sec. 5.1.2. From this discussion it is clear that all necessary variables are indeed available for aerodynamic model identification, see Table 5.3.

5.2.4 Test point selection

With a general definition of the aerodynamic models, the dimensionality of the state vector, and the dimensions of the state-planes in the previous paragraph, a choice of the location of the test points can be made. It was argued in the previous paragraph that the test points should be points along the dimensions of the full state space in which only limited nonlinearities are expected to be present. In the case of the aerodynamic model identification of the Cessna Citation II these dimensions were the Mach and altitude dimensions.

As stated before operating an aircraft is expensive. In the context of this thesis, three flights were performed which were only partially dedicated to flight tests in the framework of aerodynamic model identification. In the past, however, more flight tests have been performed with the Cessna Citation II. While the first of these flight tests were performed with a different FTIS system, a careful reconstruction of the raw data resulted in four additional usable flights, which added more than 100 maneuvers to the maneuver database.

---

[2] This changes when the Mach number increases and effects of compressibility can no longer be neglected.

Table 5.3: Available parameters for aerodynamic model identification.

Parameter	Symbol	Unit	Std Error	Sensor/Source
Angle of Attack	α	[rad]	$3.5 \cdot 10^{-3}$ rad	alpha vane
Angle of Sideslip	β	[rad]	$3.5 \cdot 10^{-3}$ rad	beta vane
True airspeed	V	$[ms]$	n/a	Pitot-static probe (AADC)
X_B-axis acceleration	A_x	$[ms^{-2}]$	$2 \cdot 10^{-2}$	Q-Flex 3100 accelerometer (IMU)
Y_B-axis acceleration	A_y	$[ms^{-2}]$	$2 \cdot 10^{-2}$	Q-Flex 3100 accelerometer (IMU)
Z_B-axis acceleration	A_z	$[ms^{-2}]$	$3 \cdot 10^{-2}$	Q-Flex 3100 accelerometer (IMU)
X_B-axis velocity	u	$[ms^{-1}]$	-	Not measurable
Y_B-axis velocity	v	$[ms^{-1}]$	-	Not measurable
Z_B-axis velocity	w	$[ms^{-1}]$	-	Not measurable
X_B-axis rotational rate	p	[rads^{-1}]	$2 \cdot 10^{-3}$	LITEF μFORS rate gyro (IMU)
Y_B-axis rotational rate	q	[rads^{-1}]	$2 \cdot 10^{-3}$	LITEF μFORS rate gyro (IMU)
Z_B-axis rotational rate	r	[rads^{-1}]	$5 \cdot 10^{-3}$	LITEF μFORS rate gyro (IMU)
X_B-axis rotation	ϕ	[rad]	$8.7 \cdot 10^{-3}$	Sperry VG-14H vertical gyro (TARSYN)
Y_B-axis rotation	θ	[rad]	$8.7 \cdot 10^{-3}$	Sperry VG-14H vertical gyro (TARSYN)
Z_E-axis rotation	ψ	[rad]	n/a	gyrosyn compass (FMS)
Altitude	h	[m]	n/a	static probe (AADC)
Elevator deflection	δ_e	[rad]	$1.7 \cdot 10^{-3}$	Elevator synchro
Aileron deflection	δ_a	[rad]	$1.7 \cdot 10^{-3}$	Aileron synchro
Rudder deflection	δ_r	[rad]	$1.7 \cdot 10^{-3}$	Rudder synchro
Engine thrust	T_e	$[N]$	n/a	Engine model
Used fuel weight	W_{fuel}	n/a $[Kg]$		FMS
Aircraft mass	m	$[Kg]$	n/a	Mass model
Aircraft inertia matrix	$\mathbf{J}$	$[Kg \cdot m^2]$	n/a	Mass model

One of the more interesting of these flights was a test flight executed in the summer of 2008, in which a prototype fly by wire system was put through its paces. The goal of this test flight was testing worst case scenarios for autopilot failures in which simultaneous maximum control surface deflections were commanded. These hardover maneuvers are very interesting as they fully excite the nonlinear coupled state-plane, allowing for the identification of coupled aerodynamic models.

In Table 5.4 a complete listing of all test flights and maneuver data used in this thesis is given. The values between brackets in Table 5.4 are the total number of maneuvers used for model identification. The remaining maneuvers are used to validate the identified models. The complete set of flight test maneuvers are plotted inside the flight envelope of the Cessna Citation II in Figure 5.4. The test points are roughly located in three altitude planes and provide a relatively dense coverage of the Mach dimension.

In the next paragraph the design of maneuvers that should be flown to best cover the nonlinear state-planes will be discussed. These maneuvers will be executed at the test points specified in this paragraph.

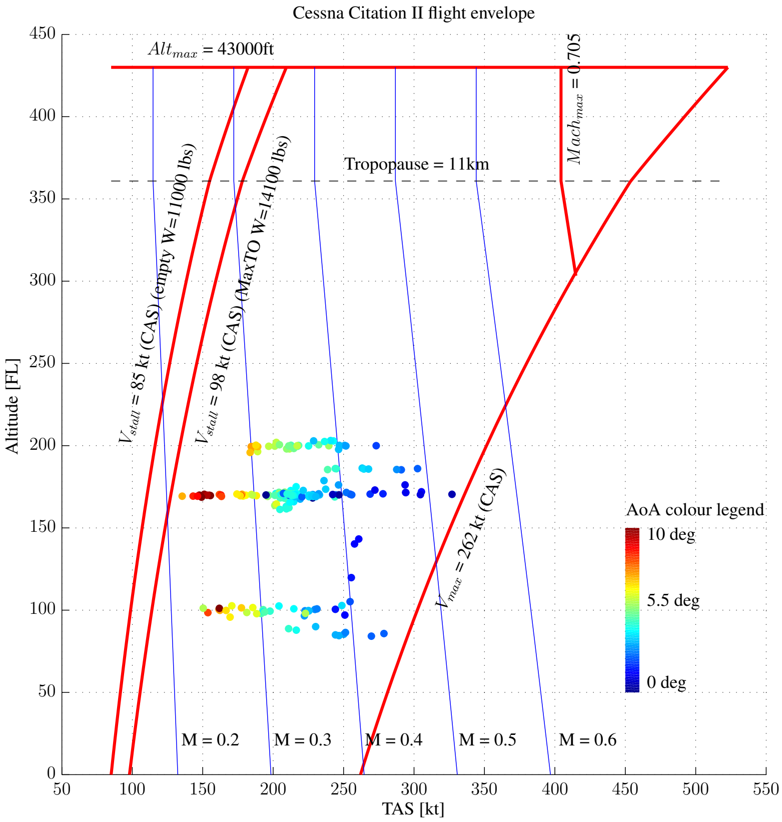


Figure 5.4: Flight envelope of the Cessna Citation II, with the 247 test points.

Table 5.4: List of test flights.

Flight No.	Flight date	Total	Coupled	Longitudinal	Lateral	Remarks
1	06-09-2006	81	25	30	26	Many FTIS glitches
2	28-09-2008	30	30	0	0	
3	30-10-2008	35	8	13	14	No GPS/engine readings
4	01-11-2008	28	9	12	7	No GPS/engine readings
5	07-10-2010	25	0	15	10	No GPS/engine readings
6	09-10-2010	17	0	4	13	No GPS/engine readings
7	10-10-2010	29	12	9	8	No GPS/engine readings
Total		247	106	79	62	

5.2.5 Maneuver input design

Since the early days of flight, scientists and engineers have tried to obtain an understanding of the dynamic behavior of the aircraft they designed. One of the first well documented flight test campaigns was that performed by Norton in the middle of the 1920's [156, 157]. Norton investigated the lateral and longitudinal dynamic stability of the Vought VE-7 'Bluebird' and the Curtiss JN-4H 'Jenny' through systematic observations on the responses to aileron and elevator step inputs.

The step inputs were the first type of input designed to excite the dynamic modes of aircraft. It was later found that the doublet input produced better results. A doublet consists of a step input followed by another step input in the opposite direction. The timing between the two step inputs can be varied to excite different frequency modes of the aircraft. This input type is widely used because it is relatively simple to produce manually, while still resulting in interesting datasets, especially for high performance unstable aircraft like Eurofighter Typhoon [162].

In 1976 a new type of control input was first introduced. Called the 3211 control input signal, or alternatively the Koehler input, it was designed to optimally excite specific frequency modes of an aircraft. The 3211 input consists of four contiguous steps with alternating signs lasting for 3, 2, 1 and 1 time units, where the time units can be adjusted to center the frequency band around the dynamic eigenmodes of an aircraft.

A more difficult input for pilots to fly is the frequency sweep input. The frequency sweep input is formed by a set of harmonic inputs with specific frequencies. Each of the individual harmonic inputs is flown for some time after which the frequency of the input is changed. This type of input is designed to completely cover all dynamic eigenmodes of an aircraft. The disadvantage of the frequency sweep input is that there can be a coupling between motions or off-axis responses which invalidates the results [141].

A more advanced approach to input design is the field of optimal input design. In the past many authors have suggested different methods of maneuver design. J.A. Mulder was arguably the first to present a formal method for the design of optimal inputs. Mulder's method was based on orthogonal input functions which were optimized based on norms of the Fisher information matrix [147, 148, 68]. Morelli used a different method for input optimization which was based on dynamic programming [143, 139, 141, 85].

Mulder's method was used during the first flight test performed in the framework of this thesis. This test flight took place on the 6^{th} of September 2006 (see Table 5.4), and consisted of more than 100 individual, manually flown maneuvers, see Figure 5.5. The second flight test was performed on the 28^{th} of September 2008 and was aimed at testing worst case scenarios for the failure of a prototype autopilot, see Figure 5.6. The hardover maneuvers were not designed specifically for aerodynamic model identification, but they did lead to very large excitations of the aircraft in the cross-coupled state-plane. The two flights on the 30^{th} of October and the 1^{st} of November 2008 were aimed at estimating the aerodynamic

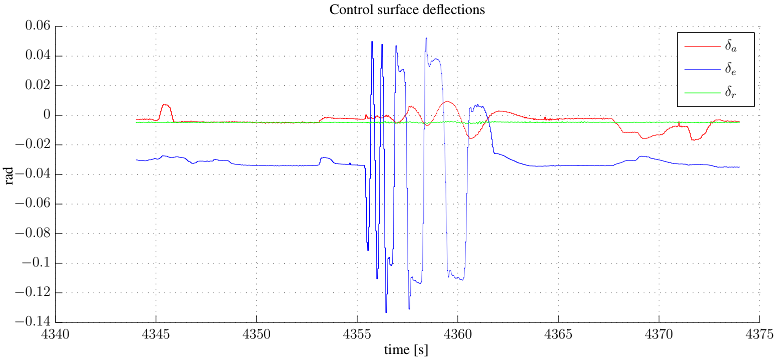


Figure 5.5: Manually flown optimal input sequence on the elevator.

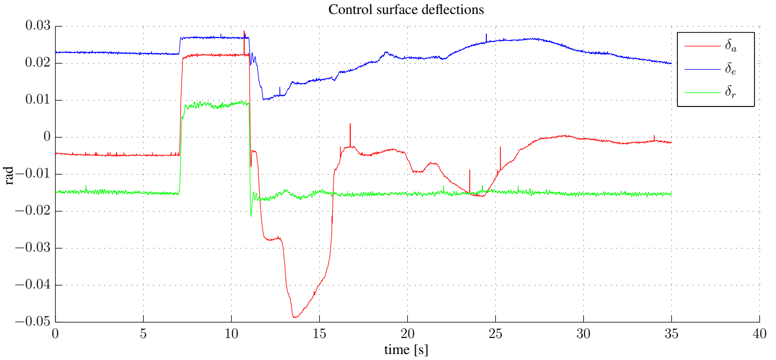


Figure 5.6: Autopilot executed hardover input sequence on the ailerons, the elevator, and the rudder.

hinge moments and consisted of single and double 3211 maneuvers which were executed by the autopilot [126, 150], see Figure 5.7. The goal of the final three flights was to produce high amplitude, coupled dynamic maneuvers which were manually executed, see Figure 5.8. With these maneuvers, parts of the nonlinear state-plane were covered that are otherwise unreachable. The final input type is a pseudo random noise (PRN) input, which is not used very often for aerodynamic model identification purposes Figure 5.9. In the general system identification community, however, the PRN is widely used because it provides a complete coverage of the frequency spectrum [117]. In this case the PRN actually was the residual of a multi-sine forcing function and a disturbance rejecting pilot input sequence [219].

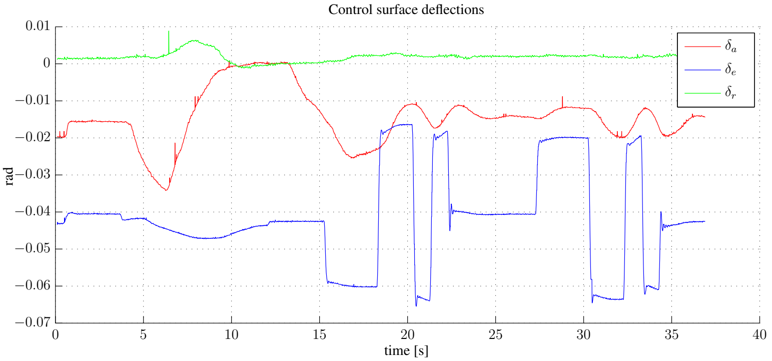


Figure 5.7: Autopilot executed double 3211 input on the elevator.

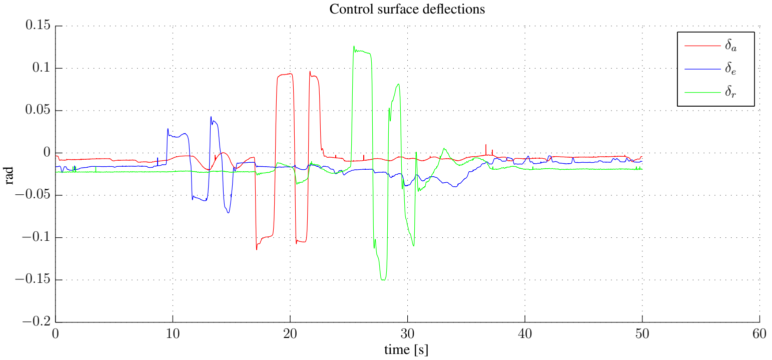


Figure 5.8: Manually flown high amplitude input sequence on the ailerons, the elevator, and the rudder.

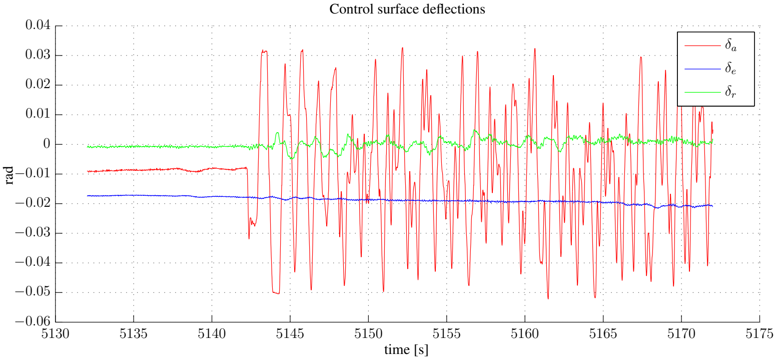


Figure 5.9: Manually flown pseudo random noise input sequence on the ailerons.

5.3 Flight Test Results

In this section the results from seven flight tests performed with the Cessna Citation II laboratory aircraft (PH-LAB) between 2006 and 2010 are presented. It will be shown that the four different input types presented in Sec. 5.2 produce very different aircraft responses, and provide a dense coverage of the flight envelope in which significant nonlinearities are expected.

In Sec. 5.3.1 the raw results from the flight testing are presented. It will then be explained in Sec. 5.3.2 that a valid spline model can only be estimated within the convex hull of the identification dataset. In Sec. 5.3.3 an analysis on the output of the linear accelerometers during ground testing is presented. In this analysis it is shown that noise levels in the accelerometers are more than an order of magnitude higher than those specified by the manufacturer. This analysis serves as a motivation for the choice of the cutoff frequency of the low-pass filter that will be applied to the raw flight data. Finally, in Sec. 5.3.4 a procedure for preprocessing the raw results from the flight testing is discussed.

5.3.1 Flight test results

The four different control surface input methods, discussed in Sec. 5.2.5, were used during the flight tests. Each of these different inputs resulted in a different response of the aircraft. First, in Figure 5.10 and Figure 5.11 the responses and aircraft states resulting from an optimal input based on Mulder's method are plotted. Noticeable in these responses are the continuously increasing angular accelerations, which reach a very significant level of $15\ deg/s^2$ for the pitch rate. Notice also that the angle of attack measured by the α-vane on the boom in Figure 5.11 which reaches a maximum of 13 degrees, which is very close to the edge of the flight envelope of the Cessna Citation II.

In Figure 5.12 and Figure 5.13 the responses and states are shown that result from a hardover input consisting of a simultaneous step input on the ailerons, the elevator, and the rudder. This time, a complex, coupled response of the aircraft is produced. A positive elevator deflection causes a nose down moment, while simultaneous positive aileron and rudder deflections cause the aircraft to roll and yaw towards the left.

In Figure 5.14 and Figure 5.15 an autopilot induced 3211 input on the elevators is executed. Interesting to note here is the high temporal and spatial precision of the maneuver. The angular rates and linear accelerations do not show the large peaks produced by the manually flown optimal input. This is both a result of the autopilot having less control authority than the pilots, and the sub-optimality of the inputs.

In Figure 5.16 and Figure 5.17 the inputs and states are shown of a manually flown high amplitude input consisting of elevator, aileron, and rudder 3211 inputs. Notice that a short period of weightlessness is experienced 14 seconds into the maneuver. This time, the angle

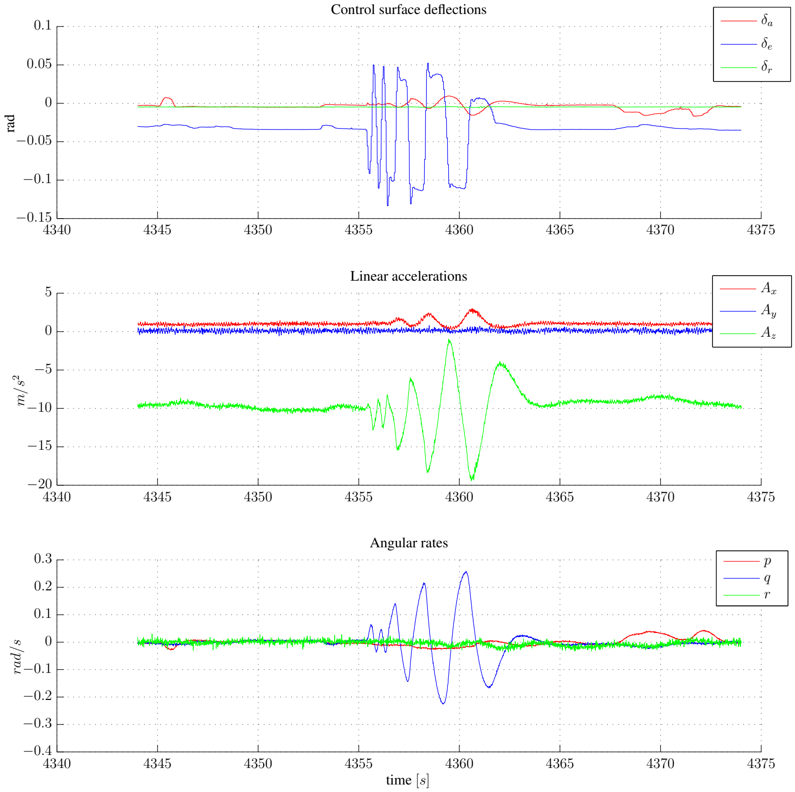


Figure 5.10: Optimal input on the elevator with dynamic responses of the aircraft.

of sideslip measured at the boom reaches a very significant value of 13 degrees, which is actually outside the envelope specified by the manufacturer.

In Figure 5.18 and Figure 5.19 the dynamic responses and states are shown that result from a manually flown pseudo-random noise (PRN) sequence on the ailerons. In this case, the PRN was the residual from a sine-forcing function and a disturbance rejecting pilot input sequence. The dynamic responses in this case are rather limited in magnitude, but the time derivatives of the responses, in this case that of the roll rate p, reach very high magnitudes.

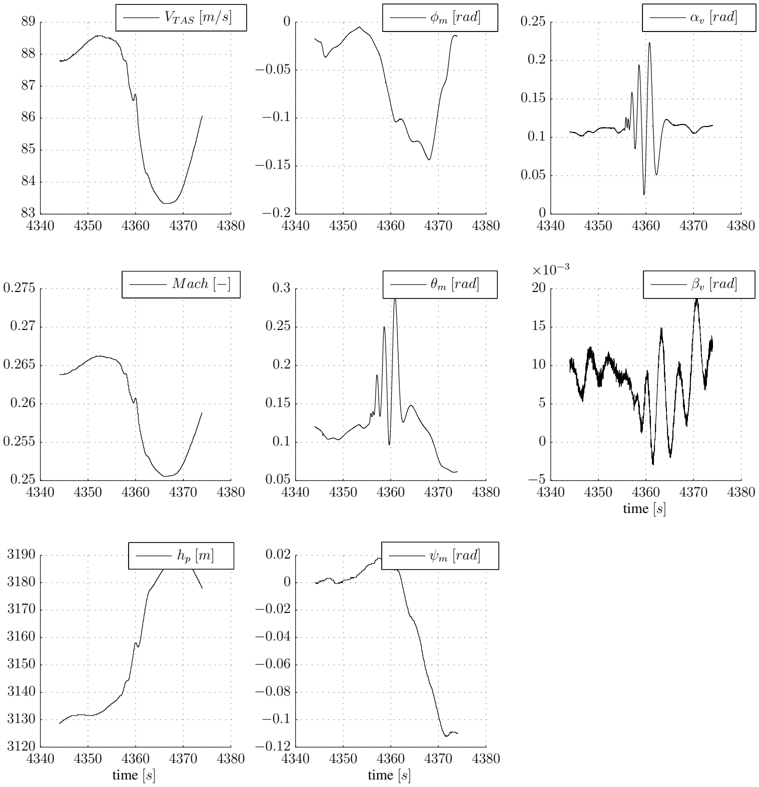


Figure 5.11: Aircraft states after an optimal input on the elevator with α_v and β_v the angle of attack and angle of sideslip measured by the vanes on the air data boom.

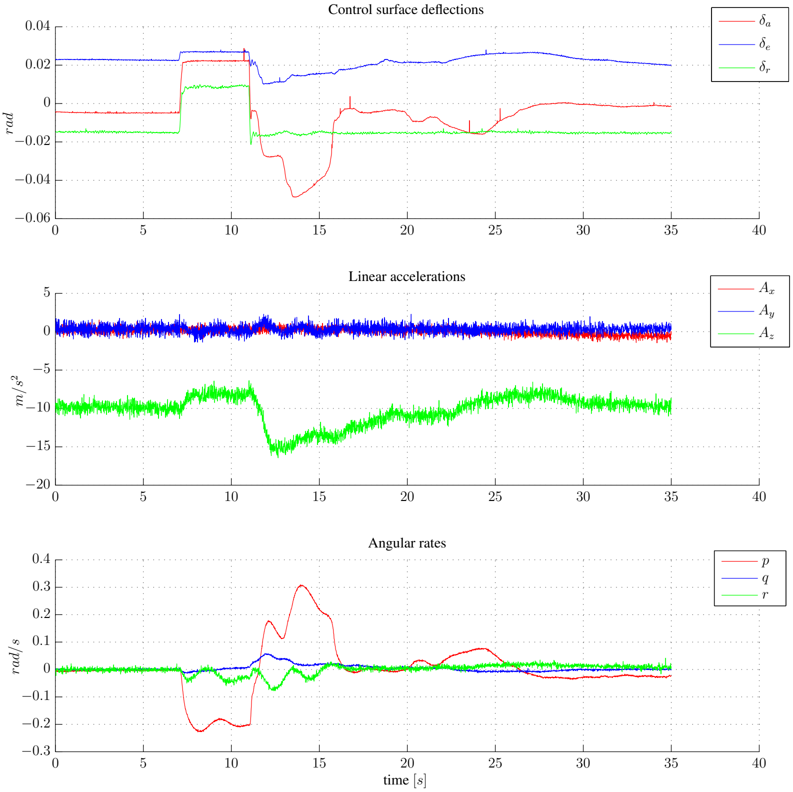


Figure 5.12: Hardover inputs on the elevator, ailerons, and rudder with the dynamic responses of the aircraft.

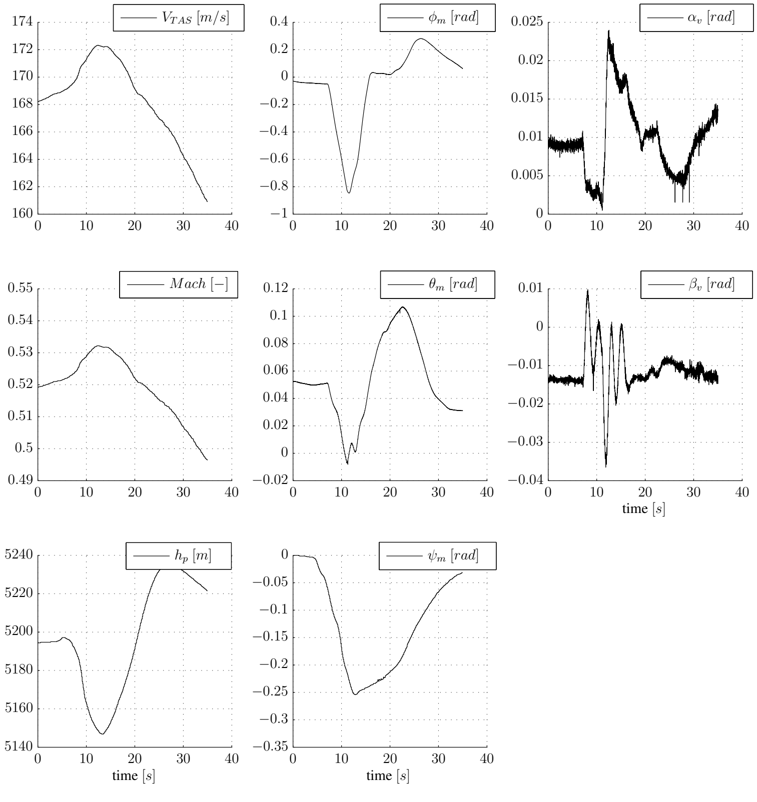


Figure 5.13: Aircraft states after a hardover input sequence, with α_v and β_v the angle of attack and angle of sideslip measured by the vanes on the boom

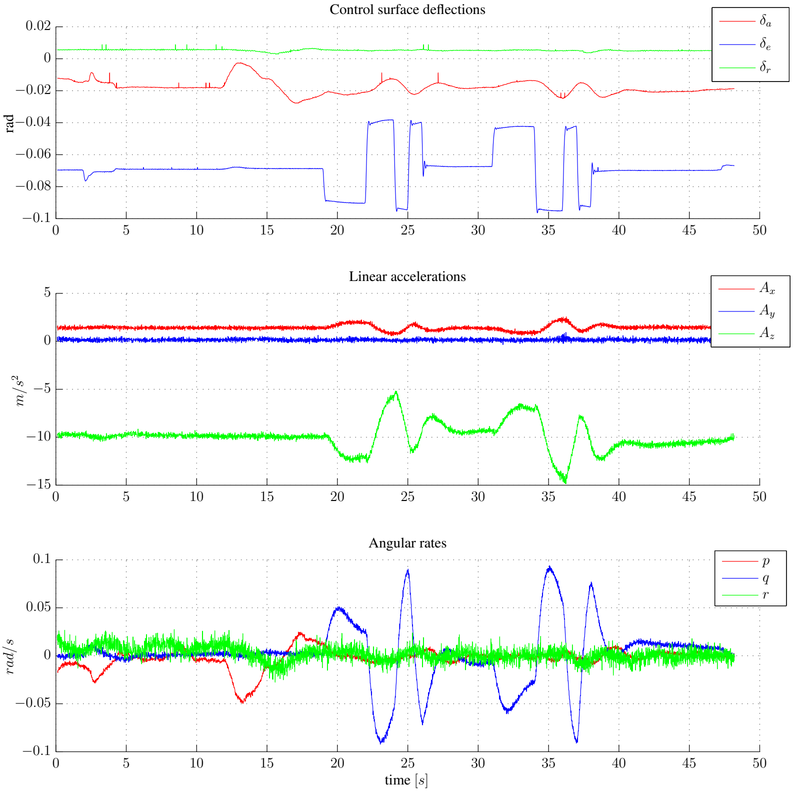


Figure 5.14: Autopilot executed 3211 input sequence on the elevator with dynamic responses of the aircraft.

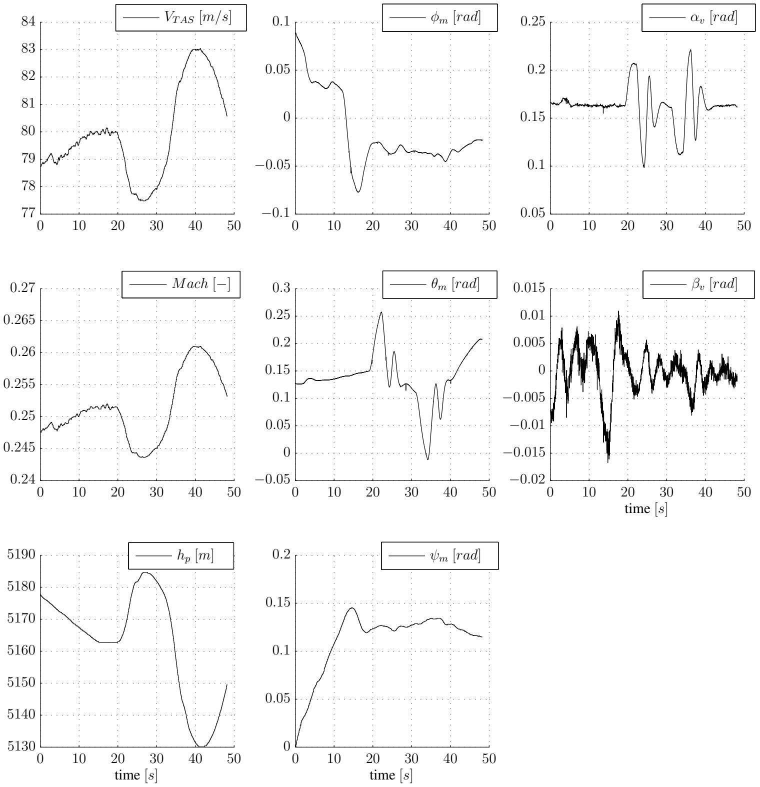


Figure 5.15: Aircraft states after an autopilot executed 3211 input sequence, with α_v and β_v the angle of attack and angle of sideslip measured by the vanes on the boom.

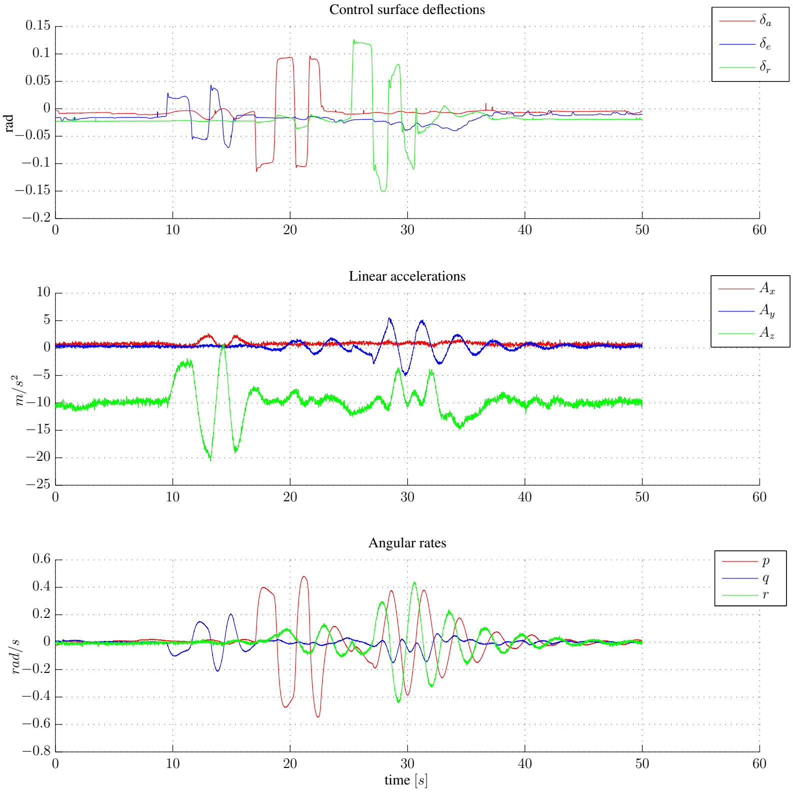


Figure 5.16: High amplitude input sequence with aircraft dynamic responses.

5.3.2 Flight envelope coverage

During the 247 flight test maneuvers a total of almost 8 million measurements were made on the state parameters. These measurement points spanned a significant part of the flight envelope of the Cessna Citation II, see Figure 5.20. Only the altitude dimension of the flight envelope received a limited covering. An important entity related to the set of measurement points is the convex hull of the measurement points. In Figure 5.20 the convex hull of the measurement points is shown as the shaded area.

The convex hull of the measurement points turns out to be a key structure for aerodynamic model identification because it bounds the area in which linear interpolation is guaranteed to be possible. This can be seen as follows. Every ridge on the edge of the

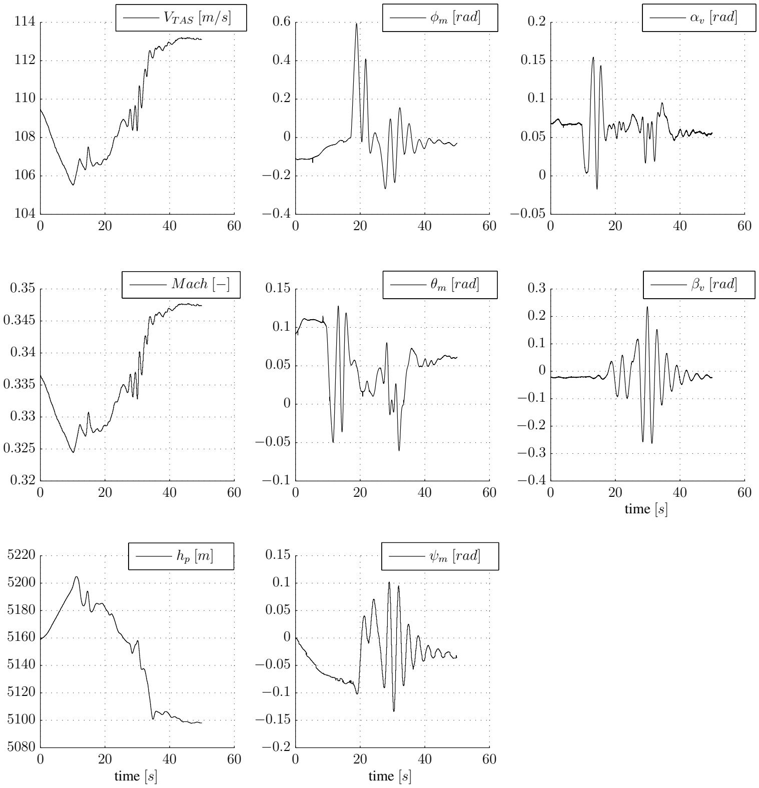


Figure 5.17: Aircraft states after a high amplitude input sequence, with α_v and β_v the angle of attack and angle of sideslip measured by the vanes on the boom.

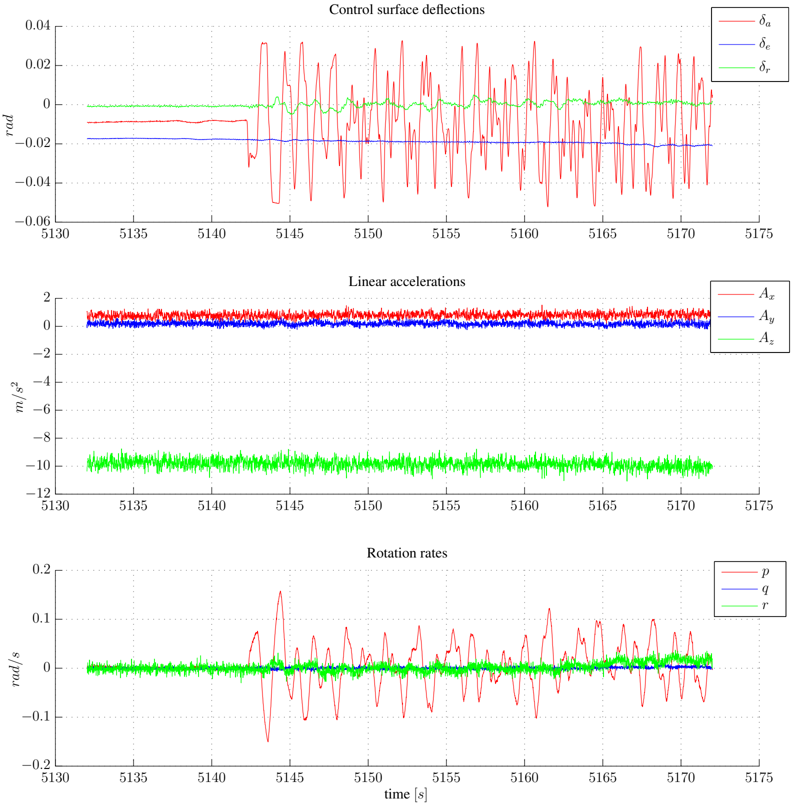


Figure 5.18: High amplitude input sequence with aircraft dynamic responses.

convex hull is formed by two vertices on the hull. The maximum degree polynomial that can be defined between each of these two vertices therefore is a linear polynomial.

In the following, the convex hull of the measurement points is called the '*region of validity*' as the spline model can only be validated within this region. The formal definition of the region of validity is the following:

$$E_X = \langle \mathcal{X}_{ident} \rangle \,, \tag{5.18}$$

with $\mathcal{X}_{ident}$ the identification dataset. The operator $\langle \bullet \rangle$ in Eq. 5.18 is the convex hull operator.

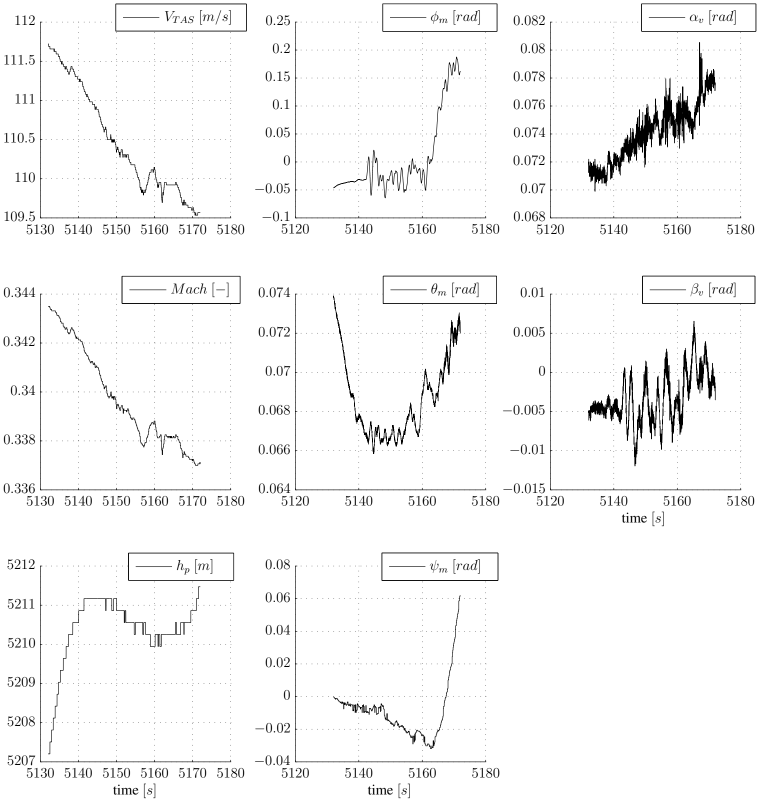


Figure 5.19: Aircraft states after a high amplitude input sequence, with α_v and β_v the angle of attack and angle of sideslip measured by the vanes on the boom.

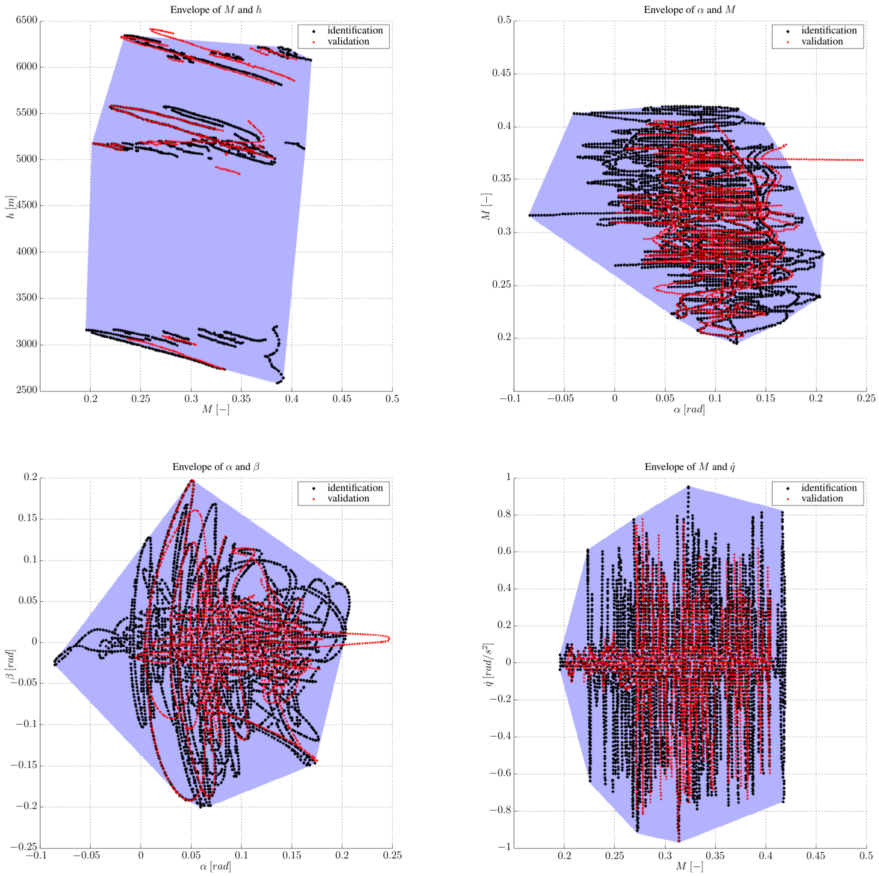


Figure 5.20: Flight envelope coverage for the 247 flight test maneuvers.

Any evaluation of the aerodynamic model outside of E_X effectively extrapolates the model, with results that can not be guaranteed to be accurate.

5.3.3 IMU sensor noise analysis

The output from the Honeywell Q-Flex 3100 accelerometers in the IMU of the Cessna Citation II laboratory aircraft contains noise of significantly higher magnitude than specified by the manufacturer[3]. Closer investigation of the noise signal shows that it is caused by an engine induced resonance. This can be concluded from the fact that the noise signal is absent in engine-off ground tests, while present in engine-idle ground tests. The spectrum

---

[3] See http://www.inertialsensor.com/docs/qa3000.pdf for full specifications.

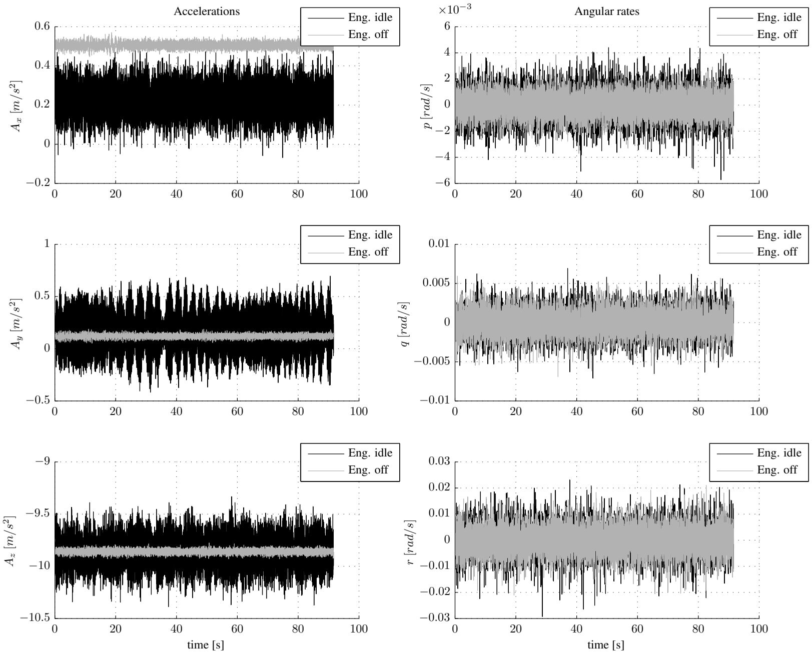


Figure 5.21: Engine-off and engine-idle accelerations and angular rates measured by the IMU during ground testing.

of the engine-idle accelerometer output shows some strong unexplained peaks around 47 Hz. Peaks of lower magnitude are present up to 7 Hz. This finding motivates the decision to pre-filter the raw IMU data with a low-pass filter with a threshold at 7 Hz. The LITEF μ-FORS fiber optic rate gyro's do not suffer from this engine induced noise.

In Figure 5.21 and Table 5.5 the raw output from the IMU sensors is plotted and compared. From the figure, it can be seen that the accelerometers produce noise of a significantly higher magnitude when the engines are running then when the engines are off. In fact, the standard deviation of the noise is more than an order of magnitude larger than that specified by the manufacturer.

Frequency domain analysis shows that there are some clear peaks in the spectra of A_x, A_y, and A_z for the engine-idle ground tests, with a very noticeable peak at 47.5 Hz. A second set of peaks is present at around 23 Hz and 26 Hz, while a third strong peak can be found at around 7 Hz. All these peaks are present in every engine-idle ground test data set, although they are more pronounced in some data sets. The spectra for the engine-off data sets do not contain any of the peaks observed in the engine-idle ground test data sets,

Table 5.5: Standard deviation of accelerometer output during ground testing.

Channel	Manufacturer Spec.	Engine-off $[m/s^2]$	Engine-idle $[m/s^2]$
$std(A_x)$	0.0147	0.0161	0.0838
$std(A_y)$	0.0147	0.0186	0.2068
$std(A_z)$	0.0147	0.0213	0.1544

Table 5.6: Standard deviation of fiber optic rate gyro output during ground testing.

Channel	Manufacturer Spec.	Engine-off $[rad/s]$	Engine-idle $[rad/s]$
$std(p)$	N/A	0.00086	0.00116
$std(q)$	N/A	0.00147	0.00156
$std(r)$	N/A	0.00486	0.00533

leading to the conclusion that the engines are the main cause of these peaks. Additionally, from observations on the shape of the noise signal in the time domain in Figure 5.21 it can be concluded that the signal bears a strong resemblance with a ‘beat’-signal, which should be caused by interference of two (or more) high frequency sources.

When filtering the accelerometer output with a low-pass filter at a cutoff frequency of 7 Hz, the noise magnitude during engine-idle tests is reduced to levels comparable with engine-off tests, see Figure 5.23 and Figure 5.24, and the numerical results in Table 5.7.

5.3.4 IMU center of gravity offset correction

The IMU is not located exactly in the center of gravity, but is located at an offset vector $(x_{cg}-x_{ac}, y_{cg}-y_{ac}, z_{cg}-z_{ac})$ from it, see Appendix A. As such, the accelerations measured by the accelerometers in the IMU are not the true accelerations at the center of gravity. In [94] Laban derives the following correction for the true linear acceleration of the center of

Table 5.7: Standard deviation of filtered accelerometer output during engine-idle ground tests.

Channel	Unfiltered $[m/s^2]$	$F_c = 40\ [Hz]$	$F_c = 22\ [Hz]$	$F_c = 7\ [Hz]$
$std(A_x)$	0.0838	0.0703	0.0437	0.0131
$std(A_y)$	0.2068	0.0604	0.0407	0.0159
$std(A_z)$	0.1544	0.1170	0.0971	0.0247

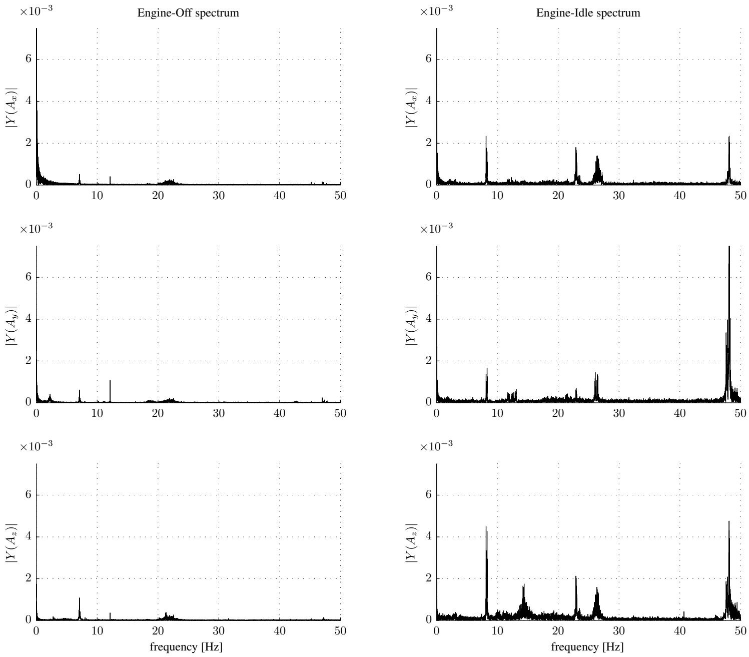


Figure 5.22: Engine-off and engine-idle spectra of the accelerometer output during ground testing.

gravity:

$$
\begin{aligned}
A_{x_{cg}} &= A_{x_{ac}} + (x_{cg} - x_{ac})(q^2 + r^2) - (y_{cg} - y_{ac})(pq - \dot{r}) - (z_{cg} - z_{ac})(pr + \dot{q}) \\
A_{y_{cg}} &= A_{y_{ac}} + (y_{cg} - y_{ac})(r^2 + p^2) - (z_{cg} - z_{ac})(qr - \dot{p}) - (x_{cg} - x_{ac})(qp + \dot{r}) \\
A_{z_{cg}} &= A_{z_{ac}} + (z_{cg} - z_{ac})(p^2 + q^2) - (x_{cg} - x_{ac})(rp - \dot{q}) - (y_{cg} - y_{ac})(rq + \dot{r})
\end{aligned}
\tag{5.19}
$$

While this correction is relatively simple to apply in theory, there is a serious implication to flight path reconstruction, the next step in the identification procedure. As it happens to be the case, the IMU sensors are subject to sensor biases and noise. The measured accelerations and angular velocities therefore are not the true accelerations and angular accelerations, but form a (presumed) linear combination with a bias term. For example, the measured pitch rate q_m is related to the true pitch rate q as follows: $q_m = q + \lambda_q$, with λ_q an unknown bias. In the next section (Sec. 5.4), a method will be presented to estimate these biases using a

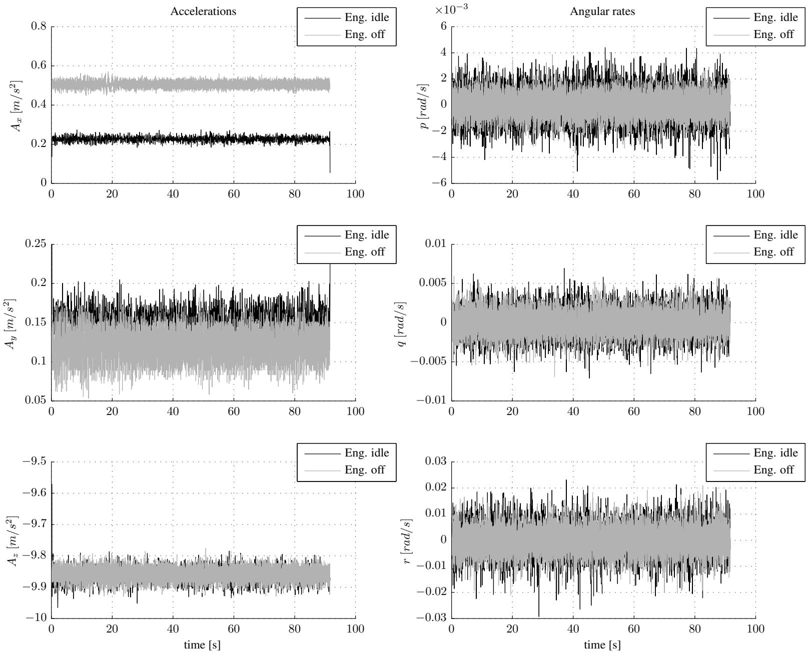


Figure 5.23: Low-pass filtered engine-idle accelerations and angular rates during ground testing, using a filter cutoff frequency at 7 Hz.

Kalman filter. However, if the correction from Eq. 5.19 is applied, the Kalman filter will become much more complex[4].

An investigation on the magnitude of the corrections from Eq. 5.19 as a function of the position of the c.g. shows that the effects are relatively minor in most cases, see Figure 5.25. In the figure, the position of the c.g. is varied along the body axes, after which the error between the true acceleration and the measured acceleration is computed. The vertical lines in the figure show a realistic arm length from the c.g. to the IMU, in this case $(x_{ac} - x_{cg}, y_{ac} - y_{cg}, z_{ac} - z_{cg}) = (1.16, 0, 0.06)$. The error between the true and measured accelerations along the Y_B and Z_B axes is in most cases smaller than $0.01\ m/s^2$, which is actually below the noise magnitude of accelerometer output after prefiltering as was shown in the previous section. The magnitude of the accelerometer measurement error along the

---

[4] More specifically, the linear accelerations would become a reconstructed state, much like the angle of attack and angle of sideslip, while the biased time derivatives $\frac{d}{dt}(p + \lambda_p)$, $\frac{d}{dt}(q + \lambda_q)$ and $\frac{d}{dt}(r + \lambda_r)$ would have to be added as additional inputs.

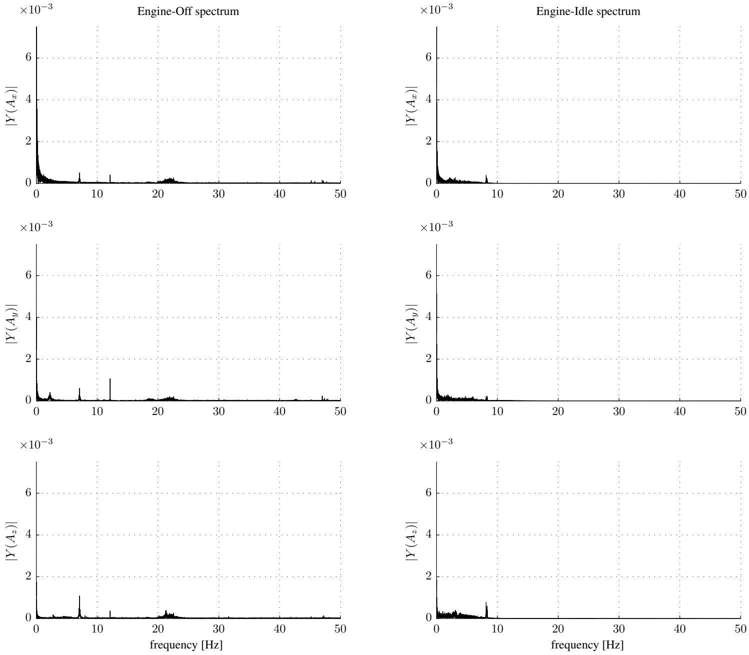


Figure 5.24: Engine-off and engine-idle spectra of the accelerometer output during ground testing after filtering the data with a low-pass filter with a cutoff frequency of 7 Hz.

X_B axis, however, will in some exceptional cases exceed 0.05 m/s^2, which means that in those cases a correction for $A_{x_{cg}}$ is warranted.

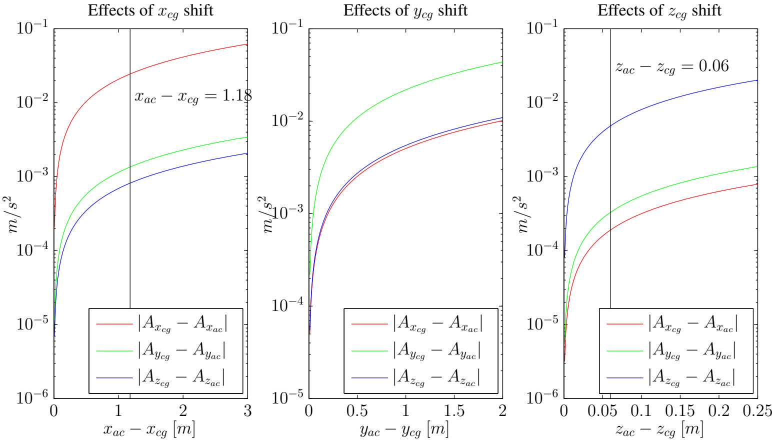


Figure 5.25: Effects of shifting the position of the c.g. on accelerometer measurements for a representative 3211 maneuver. The vertical lines in the x_{cg} and z_{cg} shift plots indicate a realistic c.g. position.

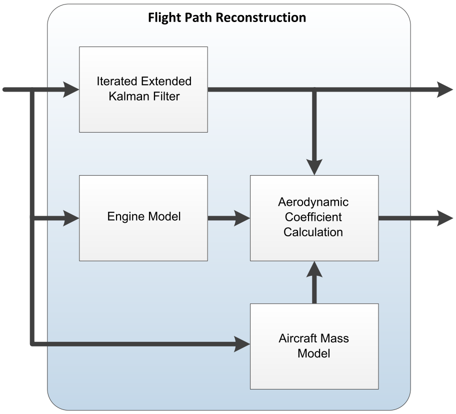

Flight Path Reconstruction
Iterated Extended Kalman Filter
Engine Model
Aerodynamic Coefficient Calculation
Aircraft Mass Model

Figure 5.26: The complete flight path reconstruction procedure.

5.4 Flight Path Reconstruction

Flight path reconstruction (FPR) is the crucial first step in the two-step method (TSM) which was developed by the Delft University of Technology during 1970's [64, 146, 147, 149]. The two step method proposes a decomposition of the nonlinear combined state and parameter estimation problem into a separate state estimation problem and an aerodynamic model parameter estimation problem. While the original combined state and parameter estimation problem constitutes a difficult nonlinear optimization problem, the two individual steps in the two-step method result in two linear optimization problems which are much simpler to solve.

In this section the FPR method introduced by Mulder et al. in [149] is applied for the estimation of the states of the Cessna Citation II (PH-LAB) laboratory aircraft. In Figure 5.4.1 an overview of the complete FPR procedure is given, which will be shown to consist of four distinct parts. The heart of the FPR procedure is an iterated extended Kalman filter (IEKF) which estimates the true states of the aircraft based on preprocessed sensor data and a set of kinematic models. In Sec. 5.4.2 a general introduction on the iterated extended Kalman filter is given. Then, in Sec. 5.4.3 the specific implementation of the IEKF used for the estimation of the states of the PH-LAB is presented. The crisp states form the input to the engine model as well as the aircraft mass model. Using the crisp states, the output from the engine model and the output from the aircraft mass model, the aerodynamic force and moment coefficients can be calculated. In Sec. 5.4.7 the results from the flight path reconstruction are presented.

5.4.1 Flight path reconstruction procedure

The complete flight path reconstruction (FPR) procedure consists of four separate parts, see Figure 5.26. The input data for the FPR procedure is the preprocessed set of flight data resulting from the preprocessing operations discussed in Sec. 5.3. The most complex part of the FPR procedure is the iterated extended Kalman filter (IEKF). The IEKF is used to estimate the true aircraft states from the biased and noise contaminated states, as will be explained in detail in the next sections.

Another block that takes the preprocessed flight data as input is the model for the Pratt & Whitney JT15D-4 turbofan engine. This model was supplied by the manufacturer of the engine and is a data table based model that predicts engine thrust as a function of a number of input parameters. One of these inputs is the revolutions per minute of the low pressure turbine ($N1$). Unfortunately, this parameter was only logged during test flights 1 and 2, but the data of test flight 1 was severely compromised by sensor glitches. The fuel flow per engine was a logged parameter, however, and using the data from flight 2 a black box simplex spline model is created that related the fuel flow with the revolutions per minute of the low pressure turbine, see Figure 5.28. Using the spline model from Eq. 5.65, $N1$ is estimated for the remaining test flights, which then forms the input to Pratt & Whitney's JT15D-4 engine model.

The final block taking preprocessed flight data as input was the mass model of the Citation II. This mass model was created at the Delft University of Technology and can be considered to be very accurate. The mass model calculates the aircraft mass, moments of inertia and the location of the c.g. as a function of the fuel load, passenger number and seating locations and baggage.

The estimated states from the Kalman filter, the estimated engine thrust from the engine model and the estimated moments of inertia from the mass model are the inputs for the aerodynamic force and moment coefficient calculation block.

In the next sections the various parts of the FPR procedure will be discussed in-depth.

5.4.2 The iterated extended Kalman filter

The Kalman filter is a recursive filter that can be used to reconstruct system states from noisy and biased sensor measurements. The Kalman filter was purportedly invented by Rudolf Kalman in 1958 although Thorvald Thiele and Peter Swerling developed had earlier developed a similar algorithm. Kalman was the first, however, to identify its use for the estimation of trajectories which lead to its implementation in the Apollo navigation computer. Since then, the Kalman filter has seen many uses, with applications ranging from cruise missile navigation computers to the phase locked loop present in most FM-radio systems.

The crux of the Kalman filter is the calculation of a weighted average between the measured and the predicted state, where the weight depends on the uncertainty in the measurement; the higher the uncertainty, the lower the weight. The number of internal states of a system may be much larger than the number of measured states. The Kalman filter allows the reconstruction of the full internal system states using only the measured states in combination with the physical relationships that bind them.

In essence, the Kalman filter is a linear filter. In the form of the extended Kalman filter (EKF) it can be used for state estimation of nonlinear systems through the local linearization of the nonlinear transition and observation models. In that case, however, the Kalman filter is no longer an optimal estimator which means that the estimated state can no longer be guaranteed to converge to the true system state. The solution to this problem is the iterated extended Kalman filter (IEKF) which uses an internal iteration which compensates for inaccuracies resulting from the linearization.

The IEKF is at the heart of the flight path reconstruction algorithm used in this thesis. Its specific implementation was discussed in detail in [149], but will be repeated here for completeness.

First, let $\mathbf{x} \in \mathbb{R}^n$ be the internal state of a general dynamic system. Then, let $\mathbf{u} \in \mathbb{R}^{N_u}$ be the vector of system inputs. Finally, let $\mathbf{y} \in \mathbb{R}^{N_y}$ be the vector of observations on the system. Now let the general nonlinear dynamic system, which relates the state $\mathbf{x}$, the input $\mathbf{u}$, and the observation $\mathbf{y}$, be defined as follows:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), t), \\ \mathbf{y}(t) &= h(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t), t), \end{aligned} \tag{5.20}$$

with $f(\bullet)$ a nonlinear state transition model, and with $h(\bullet)$ a nonlinear observation model. The terms $\mathbf{w}(t)$ and $\mathbf{v}(t_k)$ are the process noise and the observation noise, respectively. The process or input noise covariance matrix $\mathbf{Q}$ and the observation noise covariance matrix $\mathbf{R}$ are:

$$\begin{aligned} \mathbf{Q} &:= \text{Cov}(\mathbf{w}(t)) \ \in \mathbb{R}^{N_u \times N_u}, \\ \mathbf{R} &:= \text{Cov}(\mathbf{v}(t)) \ \in \mathbb{R}^{N_y \times N_y}. \end{aligned} \tag{5.21}$$

It is assumed in the following that both the input noise covariance matrix $\mathbf{Q}$ as well as the observation noise covariance matrix $\mathbf{R}$ are time invariant and known a-priori.

The Kalman filter is a discrete time filter, so consequently Eq. 5.20 should be transformed into a discrete time form. The continuous to discrete time transformation[5] is based on a Taylor series expansion of $f(\bullet)$ and $h(\bullet)$ about the current state $\mathbf{x}(k)$ at time t_k.

---

[5] Matlab provides a very useful built-in function (C2D) for the transformation of continuous time systems into discrete time systems.

For this, the Jacobian's $\mathbf{F}_\mathbf{x}(t)$ and $\mathbf{F}_\mathbf{u}(t)$ of the state transition model $f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), t)$ in the directions of respectively $\mathbf{x}$ and $\mathbf{u}$ are required:

$$\begin{aligned}\mathbf{F}_\mathbf{x}(t) &= \delta_x f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), t) \ \in \mathbb{R}^{n\times n},\\ \mathbf{F}_\mathbf{u}(t) &= \delta_u f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), t) \ \in \mathbb{R}^{n\times N_u}.\end{aligned} \tag{5.22}$$

The Jacobian $\mathbf{H}_\mathbf{x}(t)$ of the output model $h(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t), t)$ in the direction of $\mathbf{x}$ is

$$\mathbf{H}_\mathbf{x}(t) = \delta_x h(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t), t) \ \in \mathbb{R}^{n\times N_u} \tag{5.23}$$

Using the Jacobian's $\mathbf{F}_\mathbf{x}(t)$ and $\mathbf{F}_\mathbf{u}(t)$ from Eq. 5.22 the discrete time state transition and input matrices can be derived:

$$\begin{aligned}\mathbf{\Phi}(k+1|k) &= \sum_{n=0}^{\infty} \frac{\mathbf{F}_\mathbf{x}{}^n (t_{k+1}-t_k)^n}{n!} \ \in \mathbb{R}^{n\times n},\\ \mathbf{\Gamma}(k+1|k) &= \left(\sum_{n=0}^{\infty} \frac{\mathbf{F}_\mathbf{x}{}^n (t_{k+1}-t_k)^{n+1}}{(n+1)!}\right) \cdot \mathbf{F}_\mathbf{u}(t) \ \in \mathbb{R}^{n\times N_u}.\end{aligned} \tag{5.24}$$

All entities that are required for the actual IEKF are now defined. The first step in the IEKF is the one-step-ahead prediction of the discretized system state by integration of $f(\bullet)$ over the time interval $[t_k, t_{k+1}]$:

$$\hat{\mathbf{x}}(k+1|k) = \hat{\mathbf{x}}(k|k) + \int_{t_k}^{t_{k+1}} f(\hat{\mathbf{x}}(\tau), \tilde{\mathbf{u}}, \mathbf{w}(\tau), \tau) d\tau, \ \ \mathbf{x}(0|0) = \mathbf{x}_0, \tag{5.25}$$

with $\mathbf{x}_0$ the initial state, and with $\tilde{\mathbf{u}}$ the average input between step k and $k+1$:

$$\tilde{\mathbf{u}} = \frac{\mathbf{u}(k+1) + \mathbf{u}(k)}{2}. \tag{5.26}$$

The covariance matrix of the predicted state $\mathbf{x}(k+1|k)$ is:

$$\mathbf{P}(k+1|k) = \mathbf{\Phi}(k+1|k)\mathbf{P}(k|k)\mathbf{\Phi}^\top(k+1|k) + \mathbf{\Gamma}(k+1|k)\mathbf{Q}\mathbf{\Gamma}^\top(k+1|k) \ \in \mathbb{R}^{n\times n}, \tag{5.27}$$

with $\mathbf{Q} \in \mathbb{R}^{N_u\times N_u}$ the input noise covariance matrix from Eq. 5.21 and with $\mathbf{\Phi}^\top(k+1|k)$ and $\mathbf{\Gamma}(k+1|k)$ the discretized state transition and input matrices from Eq. 5.24. The initial value of the error covariance matrix is assumed to be

$$\mathbf{P}(0|0) = \mathbf{I} \ \in \mathbb{R}^{n\times n}. \tag{5.28}$$

The measurement residual or innovation $\mathbf{z}(k+1)$ is calculated as the difference between the observation $\mathbf{y}(k+1)$ and the predicted observation $h(\hat{\mathbf{x}}(k+1|k), \mathbf{u}(k+1), \mathbf{v}(k+1), t)$ as follows:

$$\mathbf{z}(k+1) = \mathbf{y}(k+1) - h(\hat{\mathbf{x}}(k+1|k), \mathbf{u}(k+1), \mathbf{v}(k+1), t). \tag{5.29}$$

The covariance matrix of the innovation is calculated as follows:

$$\mathbf{S}(k+1) = \mathbf{H_x}(k+1)\mathbf{P}(k+1|k)\mathbf{H_x}^{\top}(k+1) + \mathbf{R} \quad \in \mathbb{R}^{N_y \times N_y}, \tag{5.30}$$

with $\mathbf{H_x}(k+1)$ the Jacobian of the output model from Eq. 5.23 and with $\mathbf{R}$ the observation noise covariance matrix from Eq. 5.21. Using the covariance matrix of the predicted state $\mathbf{P}(k+1|k)$ from Eq. 5.27, the Jacobian $\mathbf{H_x}(k+1)$ from Eq. 5.23 and the covariance matrix of the innovation $\mathbf{S}(k+1)$ from Eq. 5.30, the Kalman gain matrix can be calculated:

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{H_x}^{\top}(k+1)\mathbf{S}^{-1}(k+1) \quad \in \mathbb{R}^{n \times N_y}. \tag{5.31}$$

The predicted state $\mathbf{x}(k+1|k)$ from Eq. 5.25 can now be updated using a weighted average of the innovation $\mathbf{z}(k+1)$ from Eq. 5.29 in which the Kalman gain matrix from Eq. 5.31 acts as the weighting matrix:

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)\mathbf{z}(k+1). \tag{5.32}$$

The new estimate of the state thus is $\hat{\mathbf{x}}(k+1|k+1)$. In general, however, it cannot be guaranteed that $\hat{\mathbf{x}}(k+1|k+1)$ is the true system state. This is a direct consequence of the linearization of the nonlinear system. The iterated extended Kalman filter aims to solve this problem by introducing an iterative loop which is terminated once the magnitude of the innovation drops below some predefined threshold.

The iterative part of the IEKF is initialized by defining a two new variables:

$$\begin{aligned} \eta_1 &= \hat{\mathbf{x}}(k+1|k), \\ \eta_2 &= \eta_1. \end{aligned} \tag{5.33}$$

The iteration itself is started by calculating a new Jacobian matrix $\tilde{\mathbf{H}}_{\mathbf{x}}$:

$$\tilde{\mathbf{H}}_{\mathbf{x}}(t) = \frac{\delta}{\delta \mathbf{x}} h(\eta_1, \mathbf{u}(t), \mathbf{v}(t), t). \tag{5.34}$$

Using η_1, a new innovation $\tilde{\mathbf{z}}(k+1)$ is calculated:

$$\tilde{\mathbf{z}}(k+1) = \mathbf{y}(k+1) - h(\eta_1, \mathbf{u}(k+1), \mathbf{v}(k+1), t). \tag{5.35}$$

The new innovation covariance matrix $\tilde{\mathbf{S}}(k+1)$ then becomes:

$$\tilde{\mathbf{S}}(k+1) = \tilde{\mathbf{H}}_{\mathbf{x}}(k+1)\mathbf{P}(k+1|k)\tilde{\mathbf{H}}_{\mathbf{x}}^{\top}(k+1) + \mathbf{R}, \tag{5.36}$$

with $\mathbf{P}(k+1|k)$ the covariance matrix of the predicted state from Eq. 5.27.

$$\tilde{\mathbf{K}}(k+1) = \mathbf{P}(k+1|k)\tilde{\mathbf{H}}_{\mathbf{x}}^{\top}(k+1)\tilde{\mathbf{S}}^{-1}(k+1). \tag{5.37}$$

Using Eq. 5.34, Eq. 5.35, and Eq. 5.37 the variables η_1 and η_2 are then updated as follows:

$$\begin{aligned} \eta_1 &= \eta_2, \\ \eta_2 &= \hat{\mathbf{x}}(k+1|k) + \tilde{\mathbf{K}}(k+1)\Big(\tilde{\mathbf{z}}(k+1) - \tilde{\mathbf{H}}_{\mathbf{x}}(k+1)\cdot\big(\hat{\mathbf{x}}(k+1|k) - \eta_1\big)\Big). \end{aligned} \tag{5.38}$$

After which the magnitude of the error is calculated as follows:

$$\epsilon(k+1) = \frac{\|\eta_2 - \eta_1\|\infty}{\|\eta_1\|_\infty}. \tag{5.39}$$

If $\epsilon(k+1) > \epsilon_{max}$, then the iteration continues by returning to Eq. 5.34 and calculating a new Jacobian after which the whole process is repeated. If $\epsilon(k+1) \leq \epsilon_{max}$, the iteration terminates, and the new estimated state becomes:

$$\hat{\mathbf{x}}(k+1|k+1) = \eta_2. \tag{5.40}$$

Finally, the updated error covariance matrix $\mathbf{P}(k+1|k+1)$ can be calculated as follows:

$$\mathbf{P}(k+1|k+1) = \Big(\mathbf{I} - \tilde{\mathbf{K}}(k+1)\tilde{\mathbf{H}}_{\mathbf{x}}(k+1)\Big)\mathbf{P}(k+1|k). \tag{5.41}$$

5.4.3 Kalman filter model structure

The iterated extended Kalman filter (IEKF) introduced in the previous section is applied to reconstruct the flight path of the Cessna Citation II laboratory aircraft (PH-LAB) during flight test maneuvers. In order to use the IEKF, the structure of the filter must first be determined. For this, the state vector $\mathbf{x}$, the input vector $\mathbf{u}$ and the observation vector $\mathbf{y}$ need to be defined, as well as the nonlinear transition model $f(\bullet)$ and the observation model $h(\bullet)$ from Eq. 5.20.

These definitions depend not only on the available parameters measured by the sensors, but also on the equations of motion of the aircraft. First, let $\mathcal{Y}$ be the complete set of *measurable* parameters relevant for aerodynamic model identification during the flight testing as follows:

$$\mathcal{Y} = \{V, z, A_x, A_y, A_z, p, q, r, \phi, \theta, \psi, \alpha_v, \beta_v\} \tag{5.42}$$

with α_v and β_v respectively the angle of attack and angle of sideslip as measured by the air data boom. The remaining parameters in Eq. 5.42 were already introduced in Table 5.3. The observation vector $\mathbf{y} \in \mathbb{R}^7$ can now be defined by selecting terms from Eq. 5.42 as follows:

$$\mathbf{y} = \begin{bmatrix} \phi & \theta & \psi & V & \alpha_v & \beta_v & z \end{bmatrix}^\top, \tag{5.43}$$

Then, the input vector $\mathbf{u}_m \in \mathbb{R}^6$ is defined to be:

$$\begin{aligned} \mathbf{u}_m &= \mathbf{u} + \lambda + \mathbf{w} \\ &= \begin{bmatrix} A_x \\ A_y \\ A_z \\ p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \lambda_x \\ \lambda_y \\ \lambda_z \\ \lambda_p \\ \lambda_q \\ \lambda_r \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_z \\ w_p \\ w_q \\ w_r \end{bmatrix}, \end{aligned} \tag{5.44}$$

with $\mathbf{u}$ the true accelerations, with λ a vector of sensor biases and with $\mathbf{w}$ a vector of white process noise. Note that $\mathbf{u}_m$ in Eq. 5.44 is the vector of parameters that is measured by the IMU. The sources of bias in these measurements can be explained as follows. For the linear accelerometers, biases occur when the location of the IMU does not correspond with the location of the center of gravity of the aircraft, as was demonstrated in Figure 5.10 in Sec. 5.3.4. For the rate gyros, biases are produced if there are errors in estimations for the moments and products of inertia.

The particular choice for Eq. 5.44 as the input vector has the following motivation. The input vector must contain directly measurable parameters that are only present in their non-differential forms in the kinematic equations from Eq. 5.5 and the equations of motion from Eq. 5.7. Additionally, the parameters in $\mathbf{u}$ should be ‘simple’ parameters that are not (hidden) functions of any other parameters. This prevents α_v and β_v from being included in $\mathbf{u}_m$, as these parameters are in fact unknown functions of many other parameters.

In fact, the angle of attack α_v as measured by the α-vanes on the boom is not the true geometric angle of attack as defined in Eq. 5.46. This notion has been widely recognized in the literature, see e.g. [149] and [94]. Instead, the α-vane measures a combination of the true angle of attack, a kinematically induced angle of attack, a fuselage-upwash induced angle of attack and a vertical wind component, see Figure 5.27. In [94] a model is derived for the angle of attack α_v measured by the α-vane on the boom. In this case this model is

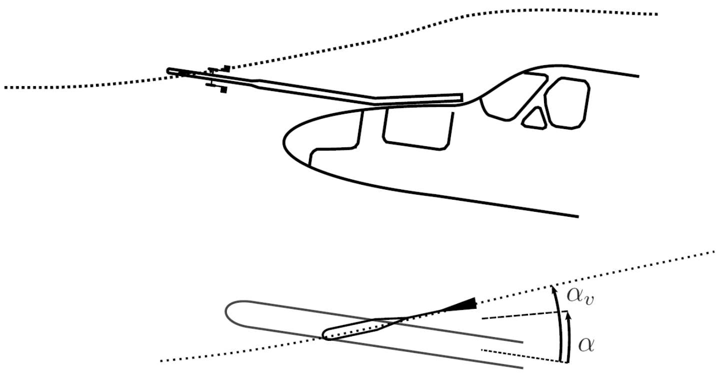


Figure 5.27: The angle of attack measured by the vane (α_v) on the boom differs from the true geometric angle of attack (α) because the fuselage induces an additional vertical flow component in the neighborhood of the vane.

extended to also include the sensor biases λ from Eq. 5.44:

$$\alpha_v = (1 + C_{\alpha_{up}})\alpha + \frac{(q + \lambda_q)x_{v_\alpha}}{\sqrt{u^2 + v^2 + w^2}} + C_{\alpha_0}, \tag{5.45}$$

with $C_{\alpha_{up}}$ the fuselage upwash coefficient, with C_{α_0} an unknown wind component and with V the true airspeed as defined in Eq. 5.49. In Eq. 5.45 the true geometric angle of attack α also occurs. The true geometric angle of attack has the following definition:

$$\alpha := \arctan\left(\frac{w}{u}\right), \tag{5.46}$$

with u and w the velocity components along the X_B, and Z_B body axes respectively.

Just like the α-vane, the β-vane on the boom does not measure the true angle of sideslip as defined in Eq. 5.48. Instead, the measured sideslip angle will be a combination of the true geometric angle of sideslip, the fuselage induced sidewash and a lateral wind component. The model for β_v given by Laban in [94] is extended to also include the sensor biases λ from Eq. 5.44 resulting in:

$$\beta_v = (1 + C_{\beta_{si}})\beta - \frac{(r + \lambda_r)x_{v_\beta}}{\sqrt{u^2 + v^2 + w^2}} + \frac{(p + \lambda_p)z_{v_\beta}}{\sqrt{u^2 + v^2 + w^2}} + C_{\beta_0}, \tag{5.47}$$

with $C_{\beta_{si}}$ the fuselage sidewash coefficient, with C_{β_0} again an unknown wind component, and with V the true airspeed as defined in Eq. 5.49. The true, geometric angle of sideslip β

used in Eq. 5.47 has the following definition:

$$\beta := \arctan\left(\frac{v}{\sqrt{u^2 + w^2}}\right), \tag{5.48}$$

with u, v, and w the velocity components along the X_B, Y_B, and Z_B respectively.

While the true geometric angle of attack and angle of sideslip are not directly measurable, they can nevertheless be inferred from Eq. 5.45 and Eq. 5.47 if the rotational rates and the velocity components of the aircraft are combined with the measured α_v and β_v, respectively. This leaves the IMU as the only direct source of input data to the IEKF, and motivates the choice for the structure of $\mathbf{u}$ as given in Eq. 5.44.

Now that the observation vector $\mathbf{y}$ and the input vector $\mathbf{u}_m$ are defined, the state vector can be constructed. The state vector should include all parameters that together allow the construction of the output vector $\mathbf{y}$. Based on the expressions for α_v from Eq. 5.45 and β_v from Eq. 5.47 this implies that the velocity components u, v, and w and the coefficients $C_{\alpha_{up}}$, C_{α_0}, $C_{\beta_{si}}$, and C_{β_0} are state variables. This would at first seem to imply that the true airspeed V is a state variable, however, on closer inspection V can be reconstructed using the velocity components u, v, and w as follows:

$$V := \sqrt{u^2 + v^2 + w^2} \tag{5.49}$$

Combining results, this leads to the following state vector:

$$\mathbf{x} = \begin{bmatrix} u & v & w & \phi & \theta & \psi & z & C_{\alpha_{up}} & C_{\alpha_0} & C_{\beta_{si}} & C_{\beta_0} \end{bmatrix}^\top, \tag{5.50}$$

The sensor biases in Eq. 5.44 should also be estimated, and therefore the state vector from Eq. 5.51 is augmented as follows:

$$\mathbf{x}_{aug} = \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} \in \mathbb{R}^{17}, \tag{5.51}$$

With the completed definitions of the observation vector $\mathbf{y}$, the input vector $\mathbf{u}_m$, and the augmented state vector $\mathbf{x}_{aug}$, construction of the state transition model $f(\bullet)$ and the observation model $h(\bullet)$ can commence. For this, the state transition equations as well as the observation equations must first be derived.

These equations are well known in the literature, see e.g. [203], but must be appended with the sensor biases λ from Eq. 5.44 as demonstrated by Mulder et al. in [149].

Starting with the state transition equations for u, v, and w, the following set of equations are found after substitution of the measured (non-stochastic) accelerations and rotational rates from Eq. 5.44 in Eq. 5.20:

$$\begin{aligned}
\dot{u} &= (A_x + \lambda_x) - (q + \lambda_q)w + (r + \lambda_r)v - g\sin(\theta), \\
\dot{v} &= (A_y + \lambda_y) - (r + \lambda_r)u + (p + \lambda_p)w + g\cos(\theta)\sin(\phi), \\
\dot{w} &= (A_z + \lambda_z) - (p + \lambda_p)v + (q + \lambda_q)u + g\cos(\theta)\cos(\phi).
\end{aligned} \tag{5.52}$$

For the state transition equations for ϕ, θ, and ψ, the substitution of the measured (non-stochastic) rotational rates from Eq. 5.44 in Eq. 5.5 leads to:

$$\begin{aligned}
\dot{\phi} &= (p + \lambda_p) + (q + \lambda_q)\sin(\phi)\tan(\theta) + (r + \lambda_r)\cos(\phi)\tan(\theta), \\
\dot{\theta} &= (q + \lambda_q)\cos(\phi) - (r + \lambda_r)\sin(\phi), \\
\dot{\psi} &= (q + \lambda_q)\sin(\phi)\sec(\theta) + (r + \lambda_r)\cos(\phi)\sec(\theta).
\end{aligned} \tag{5.53}$$

Finally, the state transition equation for the relative altitude $\mathbf{z}$ was given by [149] as follows:

$$\dot{z} = w\cos(\phi)cos(\theta) - u\sin(\theta) + vcos(\theta)sin(\phi) \tag{5.54}$$

It is important to note that the remaining states λ, $C_{\alpha_{up}}$, C_{α_0}, $C_{\beta_{si}}$, and C_{β_0} are assumed to be time invariant. Therefore, no state transition equations are required for these entities.

The complete state transition equation is therefore found by combining the expressions Eq. 5.52, Eq. 5.53, and Eq. 5.54 as follows:

$$\begin{aligned}
\dot{\mathbf{x}} &= f(\mathbf{x}_{aug}(t), \mathbf{u}_m(t), \mathbf{w}(t), t) \\
&= \begin{bmatrix}
A_x + \lambda_x - g\sin(\theta) - w(\lambda_q + q) + v(\lambda_r + r) \\
A_y + \lambda_y + w(\lambda_p + p) - u(\lambda_r + r) + g\cos(\theta)\sin(\phi) \\
A_z + \lambda_z - v(\lambda_p + p) + u(\lambda_q + q) + g\cos(\phi)\cos(\theta) \\
\lambda_p + p + (\lambda_r + r)\cos(\phi)\tan(\theta) + (\lambda_q + q)\sin(\phi)\tan(\theta) \\
(\lambda_q + q)\cos(\phi) - (\lambda_r + r)\sin(\phi) \\
(\lambda_r + r)\cos(\phi)\sec(\theta) + (\lambda_q + q)\sin(\phi)\sec(\theta) \\
w\cos(\phi)\cos(\theta) - u\sin(\theta) + v\cos(\theta)\sin(\phi) \\
\mathbf{0}_{10\times 1}
\end{bmatrix}
\end{aligned} \tag{5.55}$$

For the observation equations, the roll angle ϕ and the pitch angle θ were obtained directly from the vertical gyro, while the yaw angle ψ was obtained directly from the gyrosyn compass. The true airspeed was derived using Eq. 5.49 while the equations for α_v and β_v were already derived in Eq. 5.45 and Eq. 5.47. The only new equation is that for the relative altitude z which is given by:

$$z = h - h_0, \tag{5.56}$$

with h the time varying altitude and with h_0 the initial altitude at the start of the maneuver as measured by the AADC.

Combination of Eq. 5.45, Eq. 5.47, Eq. 5.56, and Eq. 5.49 then leads to the following observation model:

$$\begin{aligned} \mathbf{y} &= h(\mathbf{x}_{aug}(t), \mathbf{u}_m(t), \mathbf{v}(t), t) \\ &= \begin{bmatrix} \phi \\ \theta \\ \psi \\ \sqrt{u^2+v^2+w^2} \\ (1 + C_{\alpha_{up}}) \arctan\left(\frac{w}{u}\right) + \frac{(q+\lambda_q)x_{v_\alpha}}{\sqrt{u^2+v^2+w^2}} + C_{\alpha_0} \\ (1 + C_{\beta_{si}}) \arctan\left(\frac{v}{\sqrt{u^2+w^2}}\right) - \frac{(r+\lambda_r)x_{v_\beta}}{\sqrt{u^2+v^2+w^2}} + \frac{(p+\lambda_p)z_{v_\beta}}{\sqrt{u^2+v^2+w^2}} + C_{\beta_0} \\ h - h_0 \end{bmatrix} \end{aligned} \tag{5.57}$$

The Jacobians of $f(\bullet)$ from Eq. 5.55 and $h(\bullet)$ from Eq. 5.57 can now be calculated as shown in Eq. 5.22 and Eq. 5.23, respectively. Before being able to start up the IEKF, the input and observation noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ from Eq. 5.21, respectively, are initialized using the expected standard deviation of the measurement error from Table 5.3:

$$\mathbf{Q} = \begin{bmatrix} \sigma_{A_x}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{A_y}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{A_z}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_p^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_q^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_r^2 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \sigma_\phi^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\psi^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_V^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\alpha^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_\beta^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_h^2 \end{bmatrix}. \tag{5.58}$$

5.4.4 Investigation of state observability

An important consideration to be made during state estimation is the observability of the state. While it is straightforward to check the observability of linear systems, the observability of nonlinear systems is much less trivial, see e.g. [80, 127, 214]. In this thesis, the algorithm given by Walcott et al. is used [214] to analytically check the observability of the system from Eq. 5.55 and Eq. 5.57.

First, let $\mathbf{O}$ be the nonlinear observability matrix as follows:

$$\mathbf{O} = \left[\begin{array}{c} \delta_x h \\ \delta_x (L_f h) \\ \delta_x (L_f L_f h) \\ \vdots \\ \delta_x (\underbrace{L_f \cdots L_f}_{n-1} h) \end{array} \right], \tag{5.59}$$

with L_f the Lie derivative as follows:

$$\begin{array}{rcl} L_f h & = & \delta_x h \cdot f \\ L_f L_f h & = & \delta_x (L_f h) \cdot f \\ L_f L_f L_f h & = & \delta_x (L_f L_f h) \cdot f \\ & \vdots & \end{array} \tag{5.60}$$

The state is observable if the observability matrix is of full rank:

$$\operatorname{rank} \mathbf{O} = n \tag{5.61}$$

Using the Matlab symbolic toolkit, it was found that the observability matrix $\mathbf{O}$ from Eq. 5.59 was rank deficient for the state matrix given in Eq. 5.51. In fact, it was found that the observable subspace of the system did not contain the states C_{α_0} and C_{β_0}. At runtime, however, it was found that the local linearization of the system in step Eq. 5.24 and Eq. 5.34 of the IEKF further reduced the observable subspace, such that the state $C_{\beta_{si}}$ was no longer observable.

Subsequently, a new observable state $\mathbf{x}_m \in \mathbb{R}^{14}$ was defined as follows:

$$\mathbf{x}_m = \left[\begin{array}{cccccccccccccc} u & v & w & \phi & \theta & \psi & z & \lambda_u & \lambda_v & \lambda_w & \lambda_p & \lambda_q & \lambda_r & C_{\alpha_{up}} \end{array} \right]^\top, \tag{5.62}$$

Because the new observable state vector $\mathbf{x}_m$ does not contain C_{α_0}, $C_{\beta_{si}}$, and C_{β_0}, both the state transition model $f(\bullet)$ and the output model $h(\bullet)$ need to be modified to reflect this change. The new state transition model is modified from the form in Eq. 5.55 into the following:

$$
\begin{array}{rcl}
\dot{\mathbf{x}}_m & = & f(\mathbf{x}_m(t), \mathbf{u}_m(t), \mathbf{w}(t), t) \\
& = & \left[\begin{array}{l}
A_x + \lambda_x - g\sin(\theta) - w(\lambda_q + q) + v(\lambda_r + r) \\
A_y + \lambda_y + w(\lambda_p + p) - u(\lambda_r + r) + g\cos(\theta)\sin(\phi) \\
A_z + \lambda_z - v(\lambda_p + p) + u(\lambda_q + q) + g\cos(\phi)\cos(\theta) \\
\lambda_p + p + (\lambda_r + r)\cos(\phi)\tan(\theta) + (\lambda_q + q)\sin(\phi)\tan(\theta) \\
(\lambda_q + q)\cos(\phi) - (\lambda_r + r)\sin(\phi) \\
(\lambda_r + r)\cos(\phi)\sec(\theta) + (\lambda_q + q)\sin(\phi)\sec(\theta) \\
w\cos(\phi)\cos(\theta) - u\sin(\theta) + v\cos(\theta)\sin(\phi) \\
\mathbf{0}_{7\times 1}
\end{array}\right]
\end{array}
\tag{5.63}
$$

The observation model is modified from the form in Eq. 5.55 into the following:

$$
\begin{array}{rcl}
\mathbf{y}_m & = & h(\mathbf{x}_m(t), \mathbf{u}_m(t), \mathbf{v}(t), t) \\
& = & \left[\begin{array}{l}
\phi \\
\theta \\
\psi \\
\sqrt{u^2 + v^2 + w^2} \\
(1 + C_{\alpha_{up}})\arctan\left(\frac{w}{u}\right) + \frac{(q+\lambda_q)x_{v_\alpha}}{\sqrt{u^2+v^2+w^2}} + \\
\left(\arctan\left(\frac{v}{\sqrt{u^2+w^2}}\right) - \frac{(r+\lambda_r)x_{v_\beta}}{\sqrt{u^2+v^2+w^2}} + \frac{(p+\lambda_p)z_{v_\beta}}{\sqrt{u^2+v^2+w^2}}\right. \\
h - h_0
\end{array}\right]
\end{array}
. \tag{5.64}
$$

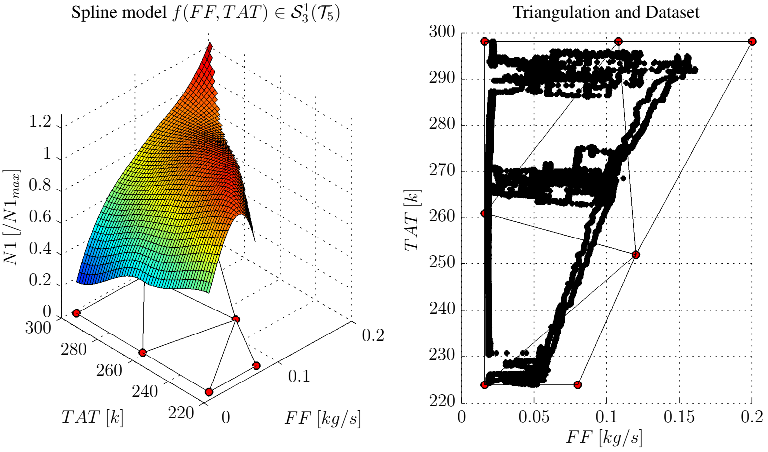
5.4.5 Engine model

The model for the Pratt & Whitney JT15D-4 turbofan engines is a complex, data table based model. The model takes various air data readings as inputs, as well as the fraction of maximum rotational velocity of the low pressure turbine stage ($N1$), which is connected to the fan. Unfortunately, during test flights 3 till 7, $N1$ was not logged; the only logged engine parameter was the fuel flow. However, this parameter is correlated to $N1$ and can be used to create a black box model that relates the fuel flow to $N1$. A bivariate simplex spline model was used to create the black box model relating the fuel flow to $N1$. The identification and validation data came from the hardover test flight in 2008. During this particular flight, both $N1$ and the fuel flow where logged. The simplest model structure that produced the best fit with the validation dataset was a 3^{rd} degree bivariate simplex spline model on 5 simplices:

$$
N1 = f(TAT, FF) \in \mathcal{S}_3^1(\mathcal{T}_5), \tag{5.65}
$$

with TAT the total air temperature in Kelvin, and with FF the fuel flow in Kg/s.

In Figure 5.28 the output from the spline function is shown, together with a plot of the


Spline model $f(FF, TAT) \in \mathcal{S}_3^1(\mathcal{T}_5)$

Triangulation and Dataset

Figure 5.28: The 3^{rd} degree simplex spline model relating the total air temperature (TAT) and the engine fuel flow (FF) to the fraction of maximum rotational velocity ($N1$) of the low pressure turbine.

triangulation and the identification dataset. This dataset consisted of more than one million data points, which were scattered through the TAT-FF plane. The output from the spline function then formed the input to the Pratt & Whitney model for the JT15D-4 turbofan engine.

5.4.6 Mass model

The mass model for the Cessna Citation II used in this thesis was developed at the faculty of Aerospace Engineering of the Delft University of Technology. The model is highly accurate and takes into account the geometry of the fuel tank, the remaining fuel weight, and the locations and weights of passengers and baggage. The mass model is initialized using estimates of the basic empty weight (BEW) of the aircraft, the locations and weights of the passengers and baggage, and the usable fuel weight. After initialization, the mass model is updated using the fuel flow parameter.

The outputs from the mass model are the moments of inertia, and the location of the center of gravity with respect to the datum of the axis system used by Cessna, a point 2.31 meter below, and 0.38 meter in front of the nose (see Appendix A). While these mass related quantities vary only little during a maneuver, there is a significant difference in their values between maneuvers. In Figure 5.29 the output from the mass model for test flight 5 is shown. This flight lasted almost three hours, during which the aircraft burned a total of 1048 Kg (2312 lbs) of fuel. During this time, the principal moments of inertia change significantly. The strongest change is found in the moment of inertia J_{xx}, the moment of

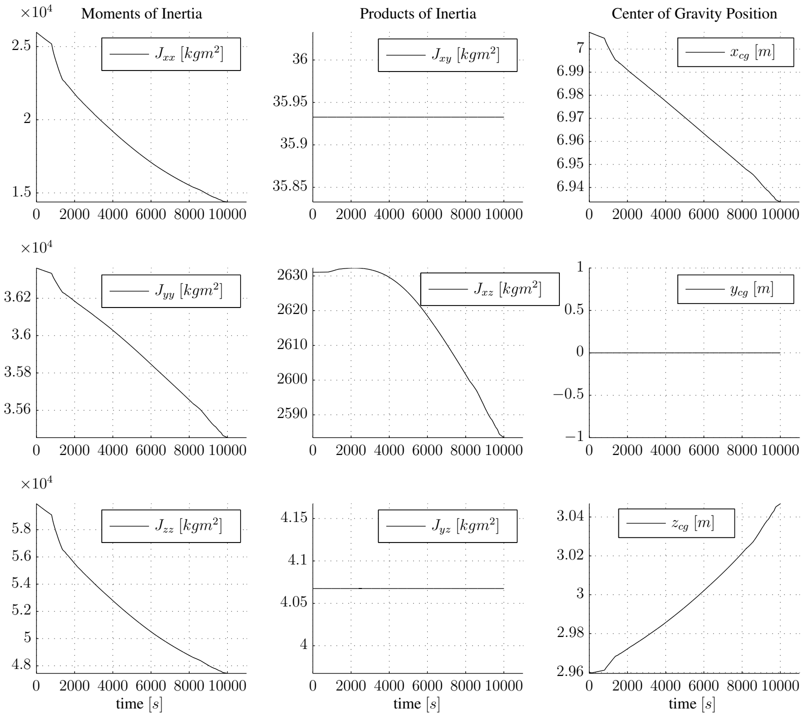


Figure 5.29: The time varying moments and products of inertia and center of gravity position for a complete test flight lasting almost three hours.

inertia about the X_B-axis, which can be explained by the aircraft losing fuel mass from the wing fuel tanks. At the same time, Figure 5.29 shows that the center of gravity position does not vary significantly during flight, as the total c.g. shift over the entire flight is in the order of centimeters.

5.4.7 Flight path reconstruction results

With the structure of the iterated extended Kalman filter defined by Eq. 5.62, Eq. 5.43, Eq. 5.44, Eq. 5.64, and Eq. 5.63, the actual flight path reconstruction operation can commence.

The only tunable values left are the expected standard deviations of the sensor measurement errors, which were already presented in Table 5.3. With these values, the input and observation noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ from Eq. 5.58 could be initialized.

The IEKF was run for all 247 flight test maneuvers. An indicative result of the state estimation for a pitch 3211 maneuver is shown in Figure 5.30. It is interesting to note that the angle of attack value measured at the boom is around 0.02 radians, or 1.14 degrees, higher than the true geometric angle of attack. This bias was present in all maneuvers, and proves that the angle of attack at the boom is indeed influenced by a fuselage induced upwash. Also interesting in Figure 5.30 is the slight bias present in the pitch angle (i.e. θ) plot, which increases over time. This is exactly as expected, because the accuracy of the vertical gyro is limited, especially during the relatively violent flight test maneuvers. These inaccuracies are integrated over time, causing a time dependent error. On the other hand, the dead reckoning sensors like the AADC supplied altitude and velocity produce values with an error which is more or less constant in time. Also interesting to note is the relative inaccuracy of the gyrosyn compass, which produces a heading angle (i.e. ψ) which differs significantly from the reconstructed heading angle.

In Figure 5.31 the sensor biases of the linear accelerometers and the rate gyroscopes are plotted for a longitudinal maneuver. A measure of the convergence of the IEKF estimated state to the true system state is obtained by initializing the IEKF with two additional initial state vectors, one with a much lower than expected initial value and one with a much higher than expected value. In Figure 5.31 these bounds are plotted as dashed lines, showing that the IEKF converges to the optimal state estimation within a few hundred samples. However, a comparison between the sensor biases of the linear accelerometers and the rate gyroscopes shows that the linear accelerometers have much larger sensor biases than the gyroscopes. In Figure 5.32 the estimated standard deviations of the sensor biases are plotted, showing a strong convergence to their respective steady state values, proving that the estimator is functioning properly.

In Figure 5.33 the $C_{\alpha_{up}}$ coefficient for a longitudinal 3211 maneuver is plotted, together with the convergence bounds. While $C_{\alpha_{up}}$ stabilizes at a value of around 0.18 the coefficient can hardly be called a constant. The comparison of the estimated values for $C_{\alpha_{up}}$ for all 79 longitudinal maneuvers showed a standard deviation of $2.43 \cdot 10^{-1}$ which is higher than any of the sensor biases. This is a strong indication that $C_{\alpha_{up}}$ is actually a function, and not a constant. In fact, it can be argued that $C_{\alpha_{up}}$ in this case is a compound function of fuselage upwash and a turbulence function. The estimated standard deviation of $C_{\alpha_{up}}$ is shown in Figure 5.34, again showing that the estimate of $C_{\alpha_{up}}$ is accurate.

In Figure 5.35 an indicative result of the state estimation for a (lateral) roll-3211 maneuver is shown. These results show that the angle of sideslip measurements are much less biased than the angle of attack measurements. The physical explanation for this is the fact that the v component of the total velocity is much smaller in magnitude than the u component of the total velocity. This means that the streamlines induced by the fuselage are effectively 2-dimensional and present mostly in the $X_b - Z_b$-plane. The remainder of the reconstructed parameters shows the same general characteristics as the reconstructed parameters resulting from the longitudinal maneuver.

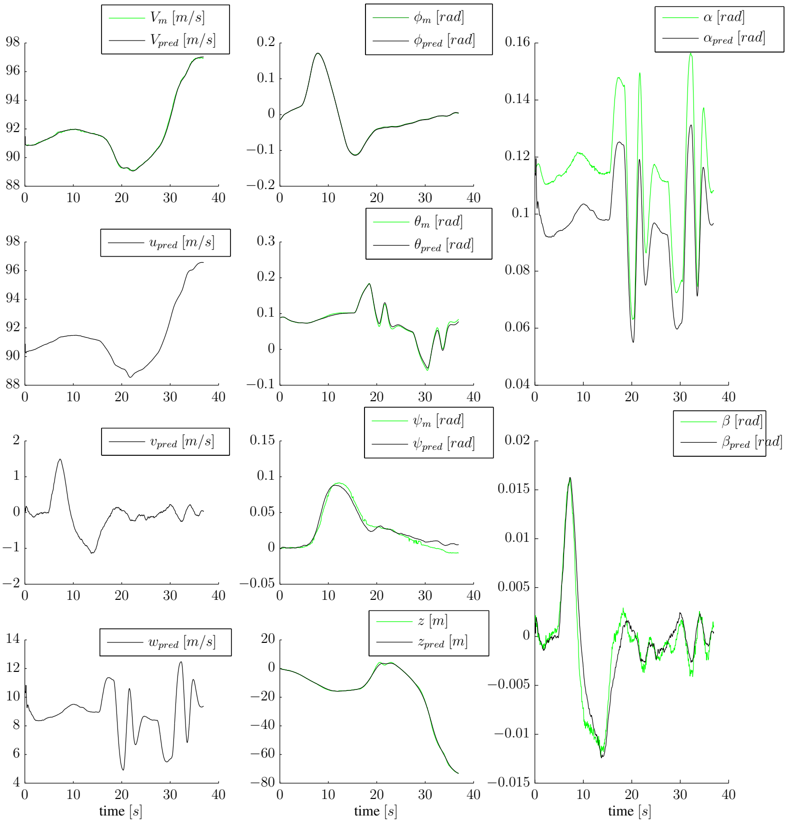


Figure 5.30: The reconstructed state parameters (u, v, w, ϕ, θ, ψ and z) together with the observed parameters (V, α and β) for a longitudinal δ_e-induced 3211 maneuver.

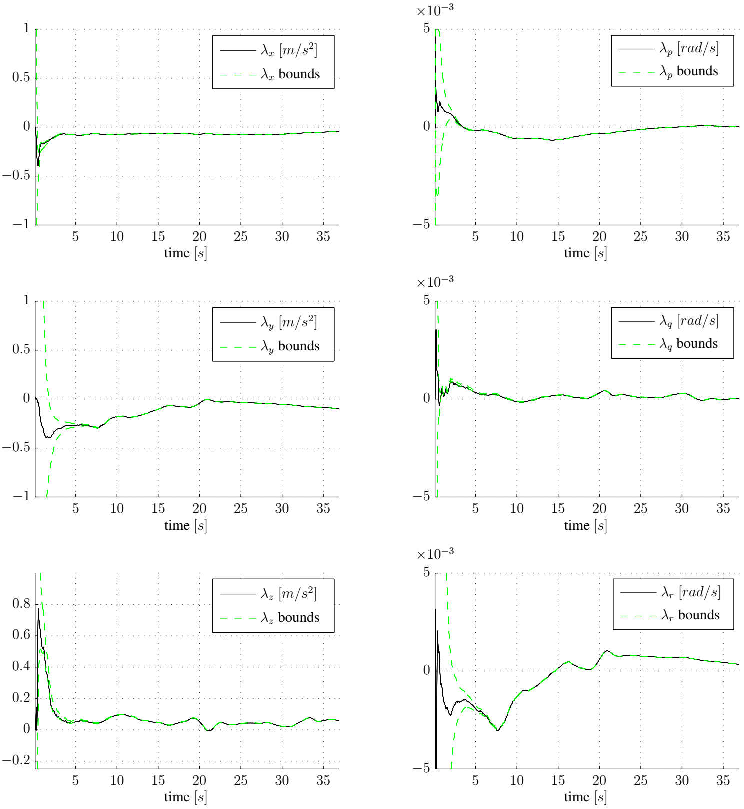


Figure 5.31: The sensor biases together with the convergence bounds obtained by initializing the IEKF with a different initial condition for a longitudinal δ_e-induced 3211 maneuver.

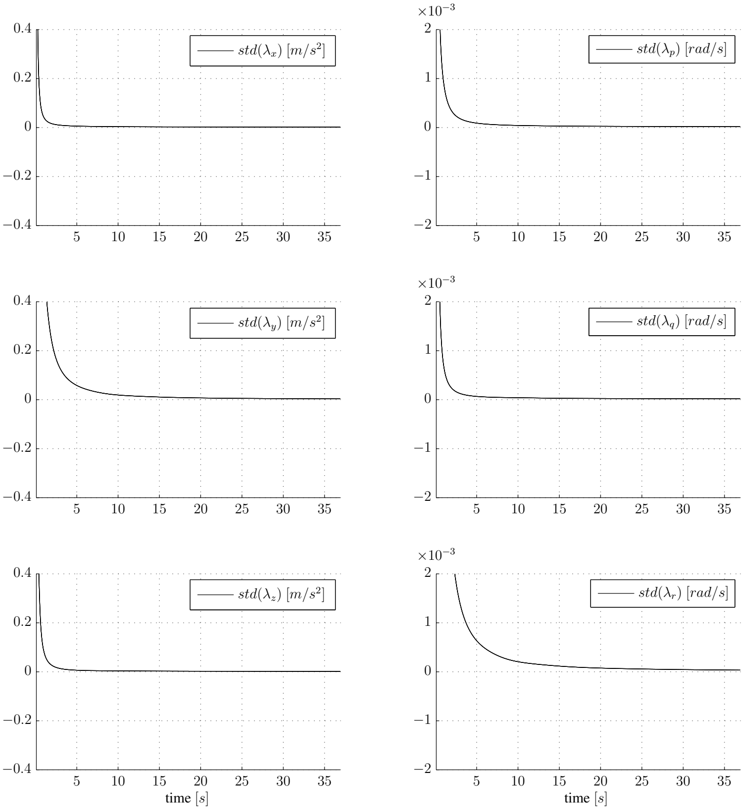


Figure 5.32: Estimated standard deviations of the sensor biases for a lateral δ_a-induced 3211 maneuver.

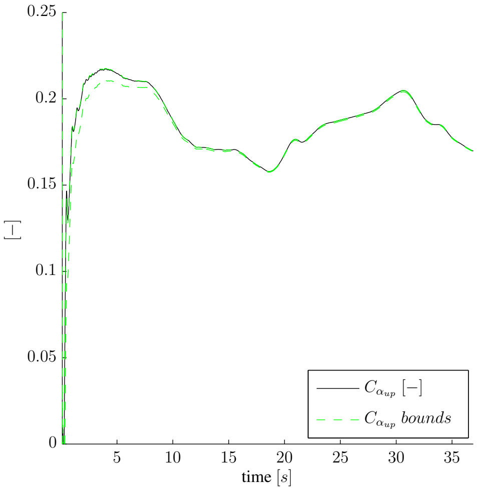


Figure 5.33: The $C_{\alpha_{up}}$ coefficient for a longitudinal δ_e-induced 3211 maneuver.

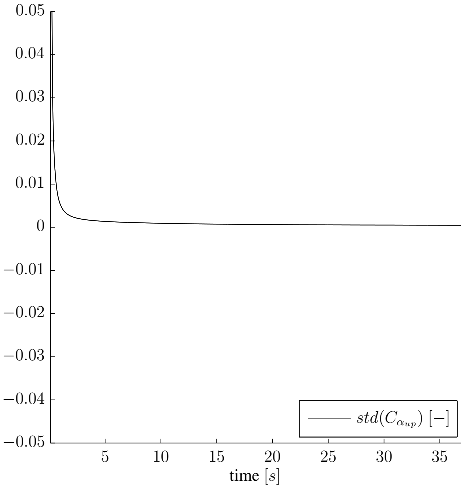


Figure 5.34: Estimated standard deviation of the $C_{\alpha_{up}}$ coefficient for a lateral δ_a-induced 3211 maneuver.

Table 5.8: Statistics of the sensor biases for all 247 maneuvers.

Bias term	Long. mean	Long. std	Lat. mean	Lat. std
λ_x	$-2.50 \cdot 10^{-2}$	$9.08 \cdot 10^{-2}$	$-3.85 \cdot 10^{-2}$	$8.70 \cdot 10^{-2}$
λ_y	$5.61 \cdot 10^{-3}$	$1.73 \cdot 10^{-1}$	$-5.34 \cdot 10^{-2}$	$1.88 \cdot 10^{-1}$
λ_z	$4.36 \cdot 10^{-2}$	$6.33 \cdot 10^{-2}$	$5.66 \cdot 10^{-2}$	$8.05 \cdot 10^{-2}$
λ_p	$-1.66 \cdot 10^{-4}$	$4.04 \cdot 10^{-4}$	$-1.88 \cdot 10^{-4}$	$3.51 \cdot 10^{-4}$
λ_q	$1.63 \cdot 10^{-4}$	$5.09 \cdot 10^{-4}$	$1.22 \cdot 10^{-4}$	$6.66 \cdot 10^{-4}$
λ_r	$4.85 \cdot 10^{-4}$	$1.37 \cdot 10^{-3}$	$-4.23 \cdot 10^{-5}$	$1.25 \cdot 10^{-3}$

In Figure 5.36 the estimated sensor biases for the lateral maneuver are shown, while Figure 5.37 shows their estimated standard deviations. A measure of the convergence of the IEKF estimated state to the true system state is obtained by initializing the IEKF with a lower and upper bound initial state vector. In Figure 5.36 these bounds are plotted, showing that the IEKF converges to the optimal state estimation within a few hundred samples, just as for the longitudinal maneuver.

In Figure 5.38 the $C_{\alpha_{up}}$ coefficient for the roll-3211 maneuver is plotted, together with the convergence bounds. In Figure 5.39 the estimated standard deviation of $C_{\alpha_{up}}$ is plotted, proving that the estimate is accurate. This time $C_{\alpha_{up}}$ stabilizes at a value of around 0.23, and does not fluctuate as much as for the longitudinal case. However, the standard deviation in the $C_{\alpha_{up}}$ values for all 62 lateral maneuvers is $2.22 \cdot 10^{-1}$ which, like for the longitudinal case, is higher than any of the sensor biases.

Finally, in Figure 5.40 and Figure 5.41 the complete set of sensor biases for respectively all 79 longitudinal and 62 lateral flight test maneuvers are plotted. These figures clearly show that the sensor biases are confined to a relatively limited range. The corresponding statistics of the bias terms are shown in Table 5.8. From the figures and the table it can be concluded that the variance in the bias of the linear accelerometers is two to three orders of magnitude higher than that of the rate gyroscopes. This again strengthens the argument that the linear accelerometers in the IMU produce substandard quality measurements.

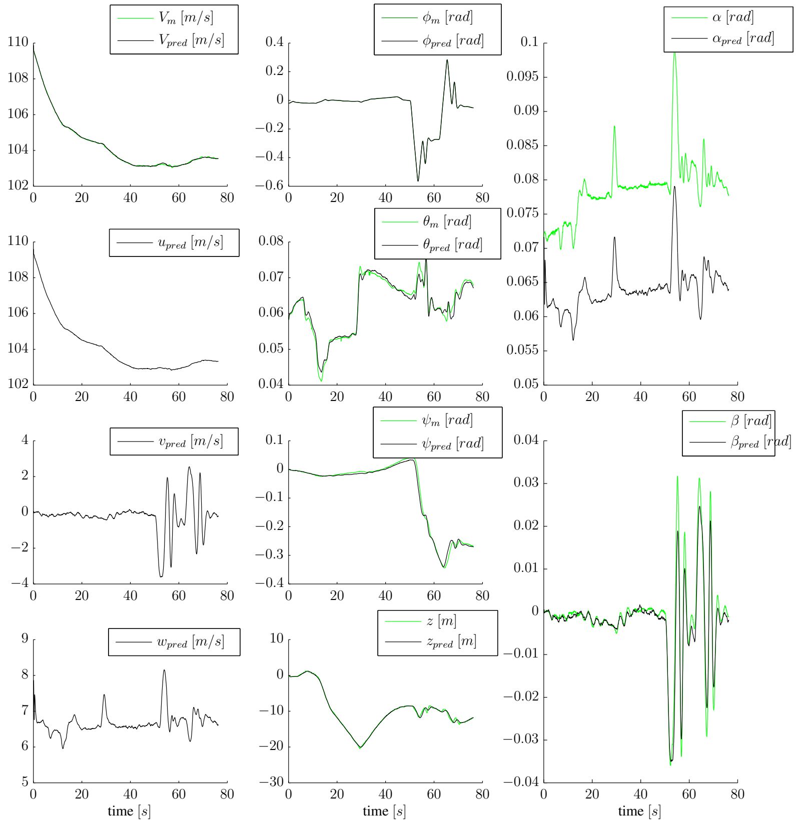


Figure 5.35: The reconstructed state parameters (u, v, w, ϕ, θ, ψ and z) together with the observed parameters (V, α and β) for a lateral δ_a-induced 3211 maneuver.

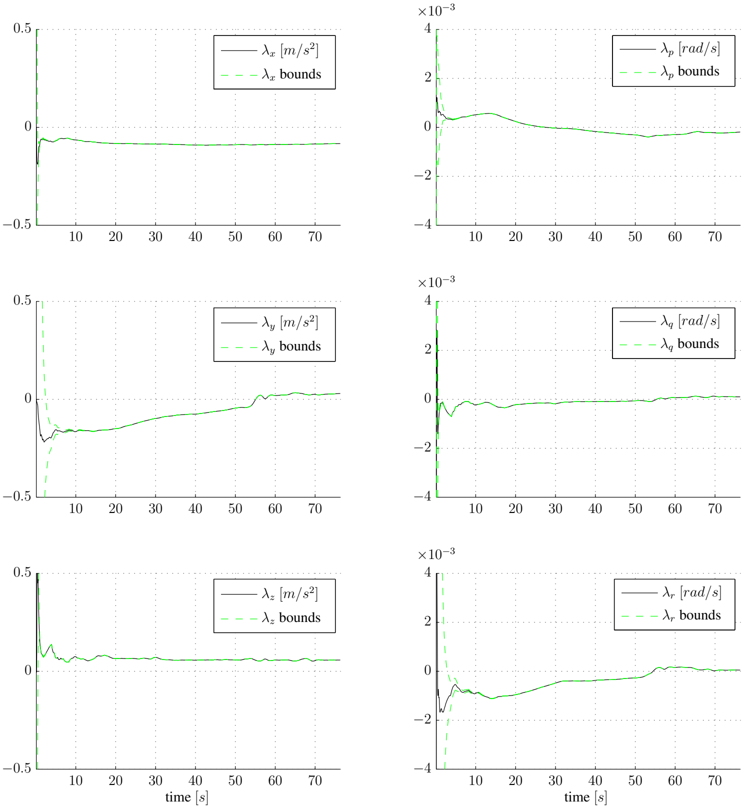


Figure 5.36: The sensor biases together with convergence bounds obtained by initializing the IEKF with a different initial condition for a lateral δ_a-induced 3211 maneuver.

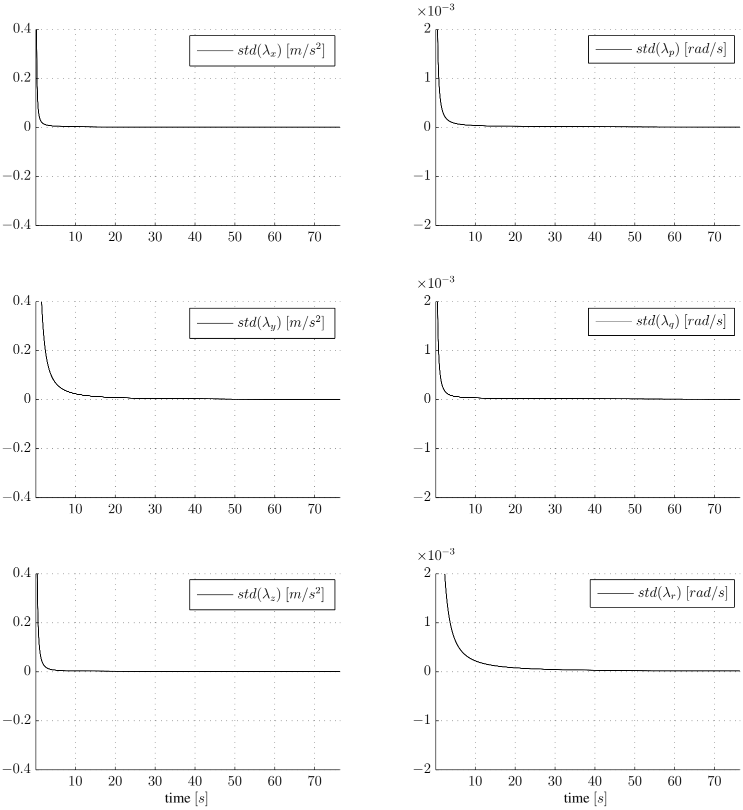


Figure 5.37: Estimated standard deviations of the sensor biases for a lateral δ_a-induced 3211 maneuver.

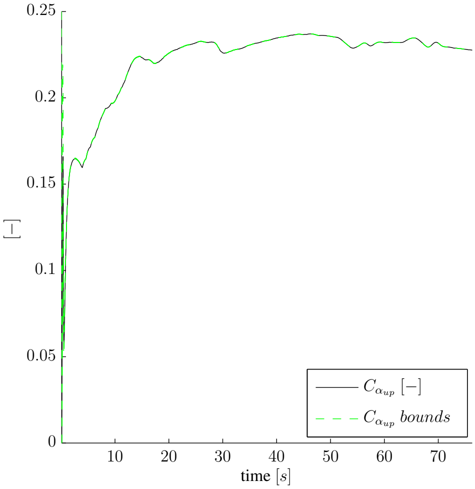


Figure 5.38: The $C_{\alpha_{up}}$ coefficient for a lateral δ_a-induced 3211 maneuver.

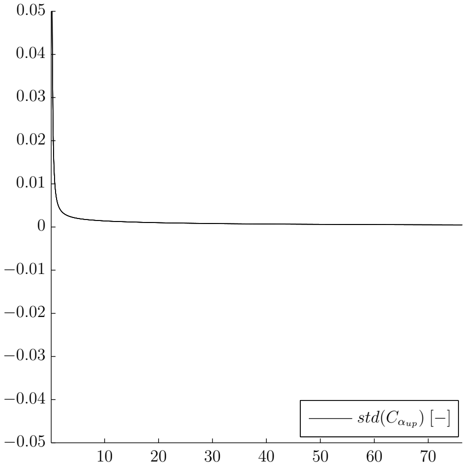


Figure 5.39: Estimated standard deviation of the $C_{\alpha_{up}}$ coefficient for a lateral δ_a-induced 3211 maneuver.

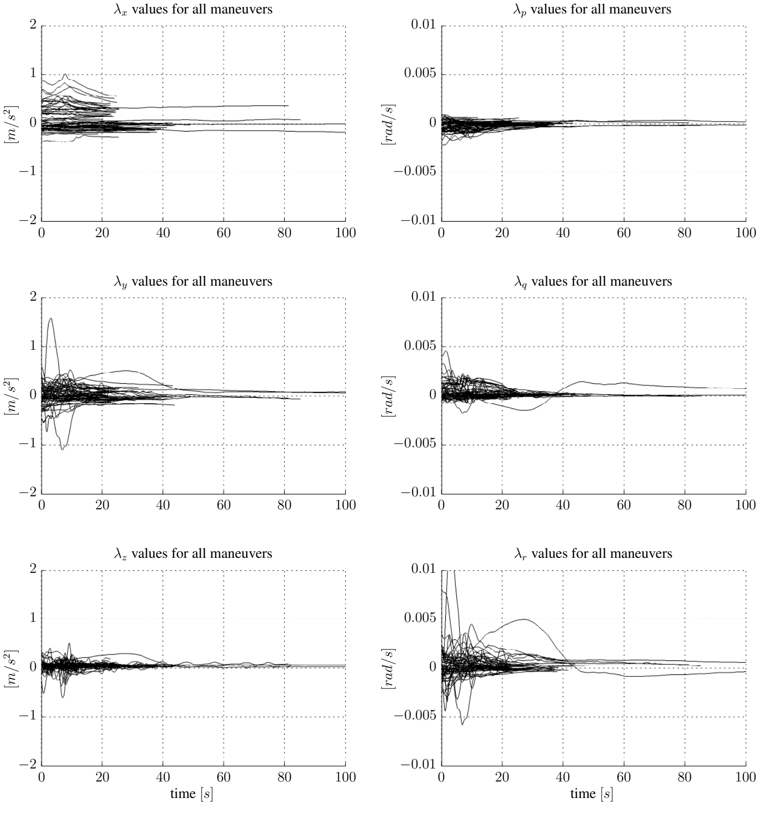


Figure 5.40: The sensor biases for all 79 longitudinal flight test maneuvers.

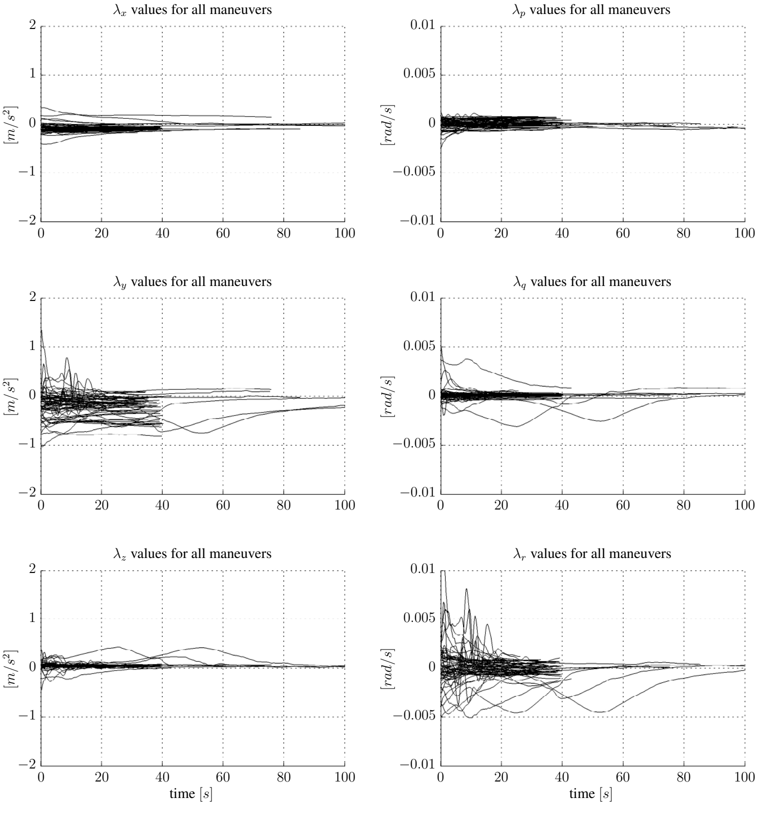


Figure 5.41: The sensor biases for all 62 lateral flight test maneuvers.

5.5 Spline Model Estimation

In this section, the new identification method based on multivariate simplex splines is used to identify spline based models for the aerodynamic force and moment coefficients of the Cessna Citation II laboratory aircraft. The data used for identification and validation was the result of 247 flight test maneuvers flown during seven test flights. The raw data was preprocessed after which flight path reconstruction techniques were used to estimate the crisp aircraft states, as explained in Sec. 5.4. The resulting dataset was split in an identification dataset, which consisted of roughly two-thirds of the maneuvers, and a validation dataset which consisted of the remaining third of the flight test maneuvers. The validation set was blindly selected from the complete dataset to prevent user induced biases in the results. The spline models were then identified using the identification dataset, and validated with the validation maneuvers.

The aim of this section is the presentation of the procedure for creating multivariate simplex spline based aerodynamic models using flight test data. First, in Sec. 5.5.1, an overview of the complete procedure is given. Next, in Sec. 5.5.2 the geometric model structure selection is discussed. Geometric model structure selection is the process in which candidate input dimensions and triangulations are defined based on a set of input and validation data. This candidate selection is performed using a novel method which analyses the occurrence and magnitude of hysteresis in datasets. After the candidate geometric model structures are selected, the candidate polynomial model structures are defined in Sec. 5.5.3. With a large set of candidate geometric and polynomial model structures, the prototype phase of aerodynamic model estimation could commence during which a total of 2142 spline models were created. In Sec. 5.5.4, the high level results from this phase are discussed. Additionally, it is demonstrated that simplex spline models are of a quality that is superior to ordinary polynomial models in the majority of cases. From the full set of prototype spline models, the best performing subsets are selected for the model refinement phase. The model refinement phase aims to create higher quality models by refining the triangulations and determining the optimal spline space for the spline models. In Sec. 5.5.6 the results from the model refinement phase are discussed.

5.5.1 A framework for AMI with simplex splines

In Figure 5.42 an overview of the spline model estimation part of the multivariate simplex spline based aerodynamic model identification (AMI) procedure is given. The input to the spline model estimator is a set of identification maneuvers, and a set of validation maneuvers. In Table 5.9, an overview is given of the number of identification and validation maneuvers.

The spline model estimation part in Figure 5.42 can be divided into a geometric model structure selection block, a polynomial model structure selection block, a parameter

Table 5.9: Properties of the identification and validation datasets.

State plane	Identification maneuvers	Validation maneuvers
longitudinal	57	22
lateral	42	20
coupled	78	28

estimation block and a model validation block. The most complex of these blocks is the geometric structure selection block. In this block a set of input dimensions is selected from a list of candidate dimensions, after which a triangulation is created which is embedded in the space spanned by the input dimensions. The second model structure selection block is the polynomial model structure selection. The selection of the polynomial model structure itself is rather trivial, as only the degree and continuity of the spline space needs to be set. However, care must be taken to avoid B-net propagation for certain combinations of polynomial degree and continuity order, as explained in Sec. 3.3.4.

With a valid triangulation and a given polynomial model structure, parameter estimation can commence. The output from the parameter estimator are the estimated B-coefficients, and the B-coefficient covariance matrix introduced in Sec. 4.2. The resulting multivariate spline model is then validated using a set of validation data. Based on the results from model validation, the loop is either repeated or exits with a multivariate simplex spline model that best fits the given identification and validation datasets.

Spline model estimation is conducted in two distinct phases, as was explained in Sec. 4.1. During the first phase, which is the prototype phase, the best performing geometric model structure is determined by creating spline models of varying degrees and continuity orders on the simplest possible triangulation, the triangulation of the hypercube. Based on this initial analysis, the best performing candidate dimensions are selected.

Using the best performing spline functions resulting from the prototype phase, the next model refinement phase can commence. During the refinement phase, the resolution of the triangulations is increased to improve model quality. Additionally, the differential constraints presented in Sec. 3.4 can be applied to further increase the quality of the model in areas with insufficient data coverage.

In the next two sections the geometric and polynomial model structure selection procedures will be discussed in depth.

5.5.2 Geometric model structure selection

Geometric model structure selection consists of the selection of the model dimensions, and the creation of a triangulation in the selected model dimensions. Finding the true set of dimensions of a model given a set of data is a key task in system identification. This task is challenging, especially when the number of dimensions is greater than 3. In this thesis

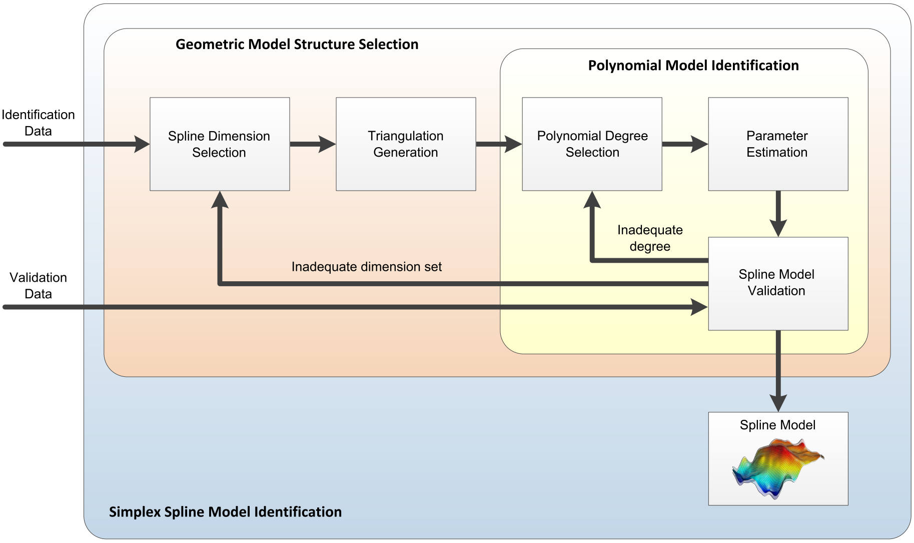

Geometric Model Structure Selection
Polynomial Model Identification
Identification Data
Spline Dimension Selection
Triangulation Generation
Polynomial Degree Selection
Parameter Estimation
Inadequate degree
Inadequate dimension set
Validation Data
Spline Model Validation
Spline Model
Simplex Spline Model Identification

Figure 5.42: An overview of the procedure for aerodynamic model estimation with simplex splines.

a novel method for determining the dimensions of a model is used. This method uses the occurrence of hysteresis as an indicator for missing model dimensions. In the following, the mathematical reasoning behind the use of this indicator is given.

First let x_i be an input dimension of some physical system S with a domain $\mathcal{D}(S) \in \mathbb{R}^s$. For example, for an aircraft x_i could be the angle of attack α, or the angle of sideslip β. Every dimension x_i of a physical system has some influence on its dynamics, otherwise it would not be considered a system dimension in the first place. This effectively means that measurements made on the output $\mathbf{y}(t)$ of S along a trajectory with a non-zero component in the direction of the axis x_i contain the influence of x_i on S. The complete input-observation pair for this trajectory would then be $\mathbf{X} = (\mathbf{x}(t), \mathbf{y}(t))$. Now let $\tilde{\mathbf{x}}(t) \in \mathbb{R}^{s-k}$, with $1 \leq k < s$ be a projection of the real trajectory $\mathbf{x}$ on the lower dimensional space $\mathbb{R}^{s-k}$. The physical interpretation of this would be that $\tilde{\mathbf{x}}(t)$ is the measured state vector of a physical system along a trajectory, while $\mathbf{x}(t)$ are its true states which are not all measured. The observations $\mathbf{y}(t)$ made along the trajectory $\tilde{\mathbf{x}}(t)$ are still the same as those made for the true trajectory $\mathbf{x}(t)$. Because $\tilde{\mathbf{x}}(t)$ is a projection of $\mathbf{x}(t)$ on a space of reduced dimension, however, there can be values in $\mathbf{x}(t)$ which are separated in $\mathbb{R}^s$ along k dimensions, but which are identical in $\mathbb{R}^{s-k}$. In that case it is possible that there are multiple, possibly widely varying, values of $\mathbf{y}(t)$ at the same location in $\mathbb{R}^{s-k}$. Hysteresis has occurred in the space $\mathbb{R}^{s-k}$, but only because k input dimensions of the system where neglected.

An illustration of this reasoning is given in Figure 5.43, where apparent hysteresis effects are shown in the left hand plot. The hysteresis in this case is caused by the fact that the

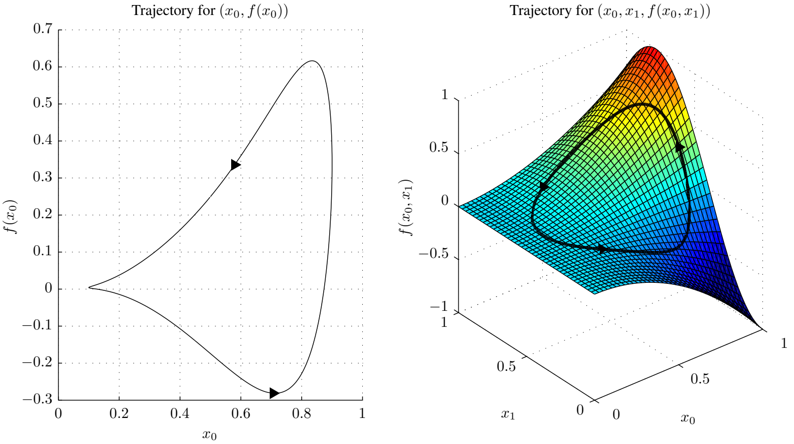


Figure 5.43: Hysteresis as a result of hidden model dimensions; apparent hysteresis for a trajectory in the $(x_0, f(x_0))$ plane (left) resulting from effects of the hidden dimension x_1 (right).

domain of the system being modeled is $\mathcal{D}(S) \in \mathbb{R}^2$, which is shown in the right hand plot. When measurements on the system are made along the trajectory $\mathbf{x}(t) \in \mathbb{R}^2$, no hysteresis occurs. Only when projecting the true trajectory $\mathbf{x}(t)$ onto $\mathbb{R}^1$ such that $\tilde{\mathbf{x}}(t) \in \mathbb{R}^1$ does hysteresis in the measurements on the system occur.

The actual hysteresis analysis is performed as follows. First, a grid of hypercubes of a selected set of dimensions is inserted in the data space. The data points are then assigned to the hypercube which envelopes them. Next, the range of values of the aerodynamic coefficients is determined for the data points in every hypercube. Finally, the aerodynamic coefficient ranges between hypercubes are compared. If there is a significant overlap of the aerodynamic coefficient ranges between hypercubes, then this may point to the occurrence of hysteresis, and a possible missing dimension. After hysteresis has been discovered, a new candidate dimension is added to the hypercube grid, after which the process is repeated. When the ranges of the aerodynamic coefficients between hypercubes show an overlap which is in the order of magnitude of the measurement noise, then an adequate set of dimensions has been selected.

It should be noted here that the accuracy of the method depends for a large part on the resolution of the grid of hypercubes. It was found that at least 10 hypercube vertices should be placed along every dimension. In spaces of high dimensionality (i.e. $n > 8$), the total set of hypercubes can easily outnumber the total number of available data points. For example, for $n = 8$ a total of 100 million 8-dimensional hypercubes would be necessary to cover the complete data space. This means that for any reasonable analysis, hundreds of millions

Table 5.10: The dimension candidates of the aerodynamic models for the longitudinal coefficients C_X, C_Z and C_m.

n	Candidate Dimension Sets
	C_X
1	$\alpha, \hat{q}, \delta_e, T_e$
2	$(\alpha, \hat{q}), (\alpha, M), (\alpha, \delta_e), (\alpha, T_e), (\alpha, \beta), (\hat{q}, M), (\hat{q}, T_e), (\delta_e, T_e)$
3	$(\alpha, \hat{q}, \delta_e), (\alpha, M, \delta_e), (\alpha, \beta, \delta_e), (\alpha, \hat{q}, T_e), (\alpha, M, T_e), (\alpha, \beta, T_e), (\alpha, \delta_e, T_e), (\alpha, \beta, M), (\alpha, \beta, \hat{q}), (\alpha, \hat{q}, M), (\alpha, M, \delta_e), (\alpha, M, T_e), (\hat{q}, \delta_e, T_e)$
4	$(\alpha, \hat{q}, \delta_e, T_e), (\alpha, \hat{q}, M, T_e)\ (\alpha, M, \delta_e, T_e)\ (\alpha, \beta, M, T_e)\ (\alpha, \beta, M, T_e)\ (\alpha, \beta, \delta_e, T_e)\ (\alpha, \beta, M, \delta_e)$
5	$(\alpha, \hat{q}, M, \delta_e, T_e), (\alpha, \beta, \hat{q}, M, T_e)\ (\alpha, \beta, \hat{q}, M, T_e)\ (\alpha, \beta, M, \delta_e, T_e)$
	C_Z
1	$\alpha, \hat{q}, \delta_e, T_e$
2	$(\alpha, \hat{q}), (\alpha, M), (\alpha, \delta_e), (\alpha, T_e), (\alpha, \beta), (\hat{q}, M), (\hat{q}, T_e), (\delta_e, T_e)$
3	$(\alpha, \hat{q}, \delta_e), (\alpha, M, \delta_e), (\alpha, \beta, \delta_e), (\alpha, \hat{q}, T_e), (\alpha, M, T_e), (\alpha, \beta, T_e), (\alpha, \delta_e, T_e), (\alpha, \beta, M), (\alpha, \beta, \hat{q}), (\alpha, \hat{q}, M), (\alpha, M, \delta_e), (\alpha, M, T_e), (\hat{q}, \delta_e, T_e)$
4	$(\alpha, \hat{q}, \delta_e, T_e), (\alpha, \hat{q}, M, T_e)\ (\alpha, M, \delta_e, T_e)\ (\alpha, \beta, M, T_e)\ (\alpha, \beta, M, T_e)\ (\alpha, \beta, \delta_e, T_e)\ (\alpha, \beta, M, \delta_e)$
5	$(\alpha, \hat{q}, M, \delta_e, T_e), (\alpha, \beta, \hat{q}, M, T_e)\ (\alpha, \beta, \hat{q}, M, T_e)\ (\alpha, \beta, M, \delta_e, T_e)$
	C_m
1	δ_e
2	$(\alpha, \delta_e), (\hat{q}, \delta_e)$
3	$(\alpha, \hat{q}, \delta_e), (\alpha, M, \delta_e), (\alpha, \beta, \delta_e), (\alpha, \delta_e, T_e), (\alpha, M, \delta_e), (\hat{q}, \delta_e, T_e), (\alpha, \dot{\alpha}, \delta_e), (\dot{\alpha}, \delta_e, T_e)$
4	$(\alpha, \hat{q}, \delta_e, T_e), (\alpha, \hat{q}, M, \delta_e), (\alpha, \dot{\alpha}, \delta_e, T_e), (\alpha, \dot{\alpha}, M, \delta_e), (\alpha, M, \delta_e, T_e), (\alpha, \beta, M, \delta_e), (\alpha, \beta, \hat{q}, \delta_e), (\alpha, \beta, \dot{\alpha}, \delta_e)$
5	$(\alpha, \beta, \hat{q}, \delta_e, T_e), (\alpha, \beta, \hat{q}, M, \delta_e), (\alpha, \beta, \dot{\alpha}, \delta_e, T_e), (\alpha, \beta, \dot{\alpha}, M, \delta_e), (\alpha, \hat{q}, M, \delta_e, T_e), (\alpha, \dot{\alpha}, M, \delta_e, T_e)$

of measurement points should be available. This problem with high dimensional spaces is aptly formulated by Tarentola in [204]: "*There is one problem with large-dimensional spaces that is easy to underestimate: they tend to be terribly empty.*"

Using the hysteresis analysis, the candidate dimension sets for the aerodynamic models were determined. In Table 5.10 the candidate dimension sets for the longitudinal models are shown. The candidate dimension sets for the lateral models are shown in Table 5.11. In the table, the dimensionless roll rate $\hat{p}$, pitch rate $\hat{q}$, yaw rate $\hat{r}$, and engine thrust $\hat{T}_e$ are defined as follows:

$$\hat{p} = \frac{pb}{2V}, \quad \hat{q} = \frac{qc}{V}, \quad \hat{r} = \frac{rb}{2V}, \quad \hat{T}_e = \frac{T_e}{\frac{1}{2}\rho V^2 S}, \tag{5.66}$$

With model dimension candidates selected, the next step in the geometric model structure selection can commence; the triangulation generation. In Chpt. 2 a geometric

Table 5.11: The dimension candidates of the aerodynamic models for the lateral coefficients C_Y, C_l and C_n.

n	C_Y
2	(β, δ_r), (β, M), $(\beta, \hat{r})$, $(\hat{r}, M)$
3	(β, M, δ_r), $(\beta, \dot{\beta}, M)$, $(\beta, \hat{r}, M)$, $(\hat{r}, M, \delta_r)$, $(\dot{\beta}, M, \delta_r)$, $(\beta, \hat{r}, \delta_r)$, $(\hat{r}, \delta_a, \delta_r)$, $(\alpha, \beta, \delta_r)$, (α, β, M)
4	$(\beta, \hat{r}, M, \delta_r)$, $(\beta, \dot{\beta}, M, \delta_r)$, $(\beta, M, \delta_a, \delta_r)$, $(\hat{r}, M, \delta_a, \delta_r)$, $(\dot{\beta}, M, \delta_a, \delta_r)$, $(\alpha, \beta, \delta_a, \delta_r)$, $(\alpha, \beta, M, \delta_r)$, $(\beta, \hat{p}, \hat{r}, M)$
5	$(\beta, \hat{r}, M, \delta_a, \delta_r)$, $(\alpha, \beta, \hat{r}, M, \delta_r)$, $(\beta, \dot{\beta}, M, \delta_a, \delta_r)$, $(\beta, \hat{p}, \hat{r}, M, \delta_r)$
	C_l
1	δ_a
2	(β, δ_a), $(\hat{p}, \delta_a)$
3	(β, M, δ_a), $(\beta, \dot{\beta}, \delta_a)$, $(\beta, \hat{p}, \delta_a)$, $(\alpha, \beta, \delta_a)$
4	$(\beta, M, \hat{p}, \delta_a)$, $(\beta, \dot{\beta}, M, \delta_a)$, $(\alpha, \beta, M, \delta_a)$, $(\beta, \hat{p}, \hat{q}, \delta_a)$
5	$(\beta, M, \hat{p}, \hat{r}, \delta_a)$, $(\beta, \dot{\beta}, M, \delta_a, \delta_r)$, $(\alpha, \beta, M, \hat{p}, \delta_a)$, $(\alpha, \beta, \dot{\beta}, M, \delta_a)$, $(\alpha, \beta, \hat{p}, \hat{q}, \delta_a)$, $(\beta, \hat{p}, M, \delta_a, \delta_r)$, $(\alpha, \beta, M, \delta_a, \delta_r)$
	C_n
1	δ_r
2	(β, δ_r), $(\hat{r}, \delta_r)$
3	$(\alpha, \beta, \delta_r)$, (β, δ_r, M), $(\beta, \hat{r}, \delta_r)$, $(\beta, \dot{\beta}, \delta_r)$
4	$(\alpha, \beta, \delta_a, \delta_r)$, $(\alpha, \beta, M, \delta_r)$, $(\alpha, \beta, \hat{r}, \delta_r)$, $(\beta, \hat{r}, M, \delta_r)$, $(\beta, \dot{\beta}, M, \delta_r)$, $(\beta, \hat{r}, \delta_a, \delta_r)$, $(\beta, \hat{q}, \hat{r}, \delta_r)$, $(\beta, M, \delta_a, \delta_r)$
5	$(\beta, M, \hat{p}, \hat{r}, \delta_r)$, $(\beta, \dot{\beta}, M, \delta_a, \delta_r)$, $(\alpha, \beta, M, \hat{r}, \delta_r)$, $(\alpha, \beta, \dot{\beta}, M, \delta_r)$, $(\alpha, \beta, \hat{p}, \hat{q}, \delta_r)$, $(\beta, \hat{r}, M, \delta_a, \delta_r)$, $(\alpha, \beta, M, \delta_a, \delta_r)$

triangulation optimization method was introduced which was based on a geometrically optimal fill of the convex hull of the data set. This method was used for the creation of the triangulations for the multivariate simplex spline model.

In order to determine the best candidate from the preselected dimensions in Table 5.10 and Table 5.11, prototype spline models were created for every individual candidate dimension. These prototype spline models use one of the simplest possible triangulations, which is the Type I triangulation of a single hypercube. Based on the performance on the single Type I hypercube triangulation, the best overall performing spline function can be selected. The triangulation of this spline function will then be refined to further increase the quality of the model. In Sec. 2.3.3 it was shown that the total number of simplices for a symmetric Type I triangulations is equal to $n!$, with n the dimension of the spline model. In Table 5.12 these numbers are repeated for clarity.

5.5.3 Polynomial model structure selection

As explained in Sec. 4.1 the polynomial model structure selection process for simplex splines is relatively straightforward. Because the basis functions of the simplex splines form

Table 5.12: Simplex count for Type I triangulations of the n-dimensional cube.

n	Simplex Count Type I triangulation
2	2
3	6
4	24
5	120

Table 5.13: Polynomial model structures for all aerodynamic models.

r	Spline spaces
-1	$\mathcal{S}_1^{-1}, \mathcal{S}_2^{-1}, \mathcal{S}_3^{-1}, \mathcal{S}_4^{-1}, \mathcal{S}_5^{-1}$
0	$\mathcal{S}_1^{0}, \mathcal{S}_2^{0}, \mathcal{S}_3^{0}, \mathcal{S}_4^{0}, \mathcal{S}_5^{0}$
1	$\mathcal{S}_2^{1}, \mathcal{S}_3^{1}, \mathcal{S}_4^{1}, \mathcal{S}_5^{1}$

a basis for the space of all polynomials of degree d, it is not necessary to select individual polynomial terms. Rather, a complete spline space of a given degree and continuity order is selected, which then contains all possible polynomial terms of total degree d.

For the spline model identification prototype phase a total of 14 different degree-continuity order combinations are used, see Table 5.13. Because the triangulation consists of only a single hypercube, B-net propagation is not an issue, and any combination of polynomial degree and continuity order can be tested. The tested polynomial model structures can be divided into three groups; a group of discontinuous spline spaces (continuity order -1), a group of value-continuous spline spaces (continuity order 0), and a group of first order continuous spline spaces (continuity order 1).

In the following example, an elaboration of the polynomial model structure of a 3^{rd} degree spline model is provided.

Example 23 (A complete 3^{rd} degree model structure for the pitching moment coefficient)**.** *Let $(\delta_e, \hat{q}, M)$ be the assumed geometric model structure for the pitching moment coefficient C_m. If a 3^{rd} degree model structure is to be used, the model has the following definition:*

$$f(\delta_e, \hat{q}, M) \in \mathcal{S}_3^1 \tag{5.67}$$

This means that for every single simplex, we have the following (equivalent) complete model structure:

$$\begin{aligned} f(\delta_e, \hat{q}, M) \quad = \quad & c_1\delta_e^3 + \\ & c_2\delta_e^2\hat{q} + c_3\delta_e^2 M + \\ & c_4\delta_e\hat{q}^2 + c_5\delta_e\hat{q}M + c_6\delta_e M^2 + \\ & c_7\hat{q}^3 + c_8\hat{q}^2 M + c_9\hat{q}M^2 + c_{10}M^3 \end{aligned} \tag{5.68}$$

with $c_1, c_2, \ldots, c_{10}$ *the polynomial coefficients that are estimated during parameter estimation.*

5.5.4 Prototype phase results

Spline model estimation consists of two distinct phases. In the first phase, the prototype phase, a large number (i.e. 2000+) of simplex spline models are created using the ordinary least squares B-coefficient estimator from Sec. 4.2. The goal of this phase is the selection of the best performing candidate dimension set. The triangulation used during the prototype phase is the Type I triangulation of the hypercube. For the polynomial model structures all spline spaces from Table 5.13 are used.

Parameter estimation is performed using a set of identification data which consists of a subset of the flight test maneuvers. The remaining maneuvers are then used to validate the estimated spline models, see Table 5.4. For assessing the quality of the created spline models, three different quality measures are used; the root mean square (RMS) of the residual, the maximum absolute residual, and the coefficient of determination. These model quality measures will first be discussed.

First, let $\mathcal{X}_{val} \in \mathbb{R}^s$ be a set containing a total of N_{val} validation data locations. The set of validation locations is formed by the union of N_{val} vectors in $\mathbb{R}^s$ as follows:

$$\mathcal{X}_{val} = \bigcup_{i=1}^{N_{val}} \mathbf{x}_i, \quad \mathbf{x}_i \in \mathbb{R}^s, \tag{5.69}$$

with $\mathbf{x}_i$ a single validation data location.

The model residual of the general aerodynamic coefficient C and the spline function s can then be defined as follows:

$$\epsilon(\mathcal{X}_{val}) := C(\mathcal{X}_{val}) - s(\mathcal{X}_{val}), \tag{5.70}$$

with $C(\mathcal{X}_{val})$ the measured values of one of the six aerodynamic coefficients at every location in the validation dataset $\mathcal{X}_{val}$, and with $s(\mathcal{X}_{val})$ the values predicted by the simplex spline model at these same locations. The residual RMS is then calculated as follows:

$$RMS\epsilon := \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\epsilon(\mathbf{x}_i))^2}, \quad \mathbf{x}_i \in \mathcal{X}_{val}. \tag{5.71}$$

The quality of aerodynamic models can be better assessed and compared if the residual RMS is scaled by the range of values of the aerodynamic coefficient at the validation locations as follows:

$$RMS_{rel}\epsilon = \frac{RMS\epsilon}{|\max C(\mathcal{X}_{val}) - \min C(\mathcal{X}_{val})|}. \tag{5.72}$$

In the following, the scaled RMS will be called the 'relative RMS', or RMS_{rel} score. The relative RMS gives a good and easily interpretable indication of the quality of a model. For example, if $RMS_{rel} = 1.0$ the root of the mean squared residual is of the same magnitude as the range of the data being approximated, from which it can be concluded that this particular model is of inadequate quality. In this thesis, a model is considered to be of adequate quality if it has an RMS_{rel} score of $< 5 \cdot 10^{-2}$, while a model is considered excellent if it achieves an RMS_{rel} score of $< 1 \cdot 10^{-2}$.

The RMS_{rel} score is just part of the whole story. A model can achieve an adequate RMS_{rel} score, but still have significant outliers, indicating a possible overfitting model or a numerical instability in the B-coefficient estimator. This is why the maximum residual is also considered an important model quality measure. Using Eq. 5.70 the maximum residual is defined as follows:

$$\epsilon_{max} := \max |\epsilon(\mathcal{X}_{val})|. \tag{5.73}$$

Again, a better assessment of the quality of the spline models can be obtained if the maximum error is scaled by the range of $C(\mathcal{X}_{val})$ as follows:

$$\epsilon_{relmax} := \frac{\max |\epsilon(\mathcal{X}_{val})|}{|\max C(\mathcal{X}_{val}) - \min C(\mathcal{X}_{val})|}. \tag{5.74}$$

This is the maximum relative error ϵ_{relmax}. The maximum relative error should not be used as the primary model quality measure since a single outlier of an otherwise adequate spline model can lead to it being classified as inadequate. Together with the relative RMS score, however, it provides an efficient way of assessing the overall quality of a spline model.

A final measure of model quality is the well-known coefficient of determination R^2 which is defined using Eq. 5.70 as follows:

$$R^2 := 1 - \frac{\sum_{i=1}^{N_{val}} \epsilon_i(\mathcal{X}_{val})^2}{\sum_{i=1}^{N_{val}} \left(C_i(\mathcal{X}_{val}) - \overline{C(\mathcal{X}_{val})}\right)^2}, \tag{5.75}$$

with $\overline{C(\mathcal{X}_{val})}$ the mean value of $C(\mathcal{X}_{val})$.

The coefficient of determination was not used as a primary measure of model quality because in some cases it produced overly optimistic validation results for models that were of obviously low quality.

The complete set of prototype spline models consisted of 2142 models. In Figure 5.44, Figure 5.45, Figure 5.46, Figure 5.47, Figure 5.48, and Figure 5.49 the overall model validation results are shown for all models. In all figures a more negative RMS_{rel} and ϵ_{relmax} score should be interpreted as a more accurate validation result and thus a higher quality model. In the figures, the model validation results on the exact same datasets of 2142 ordinary polynomial models are shown. Like the simplex spline models, the polynomial models are estimated using an ordinary least squares estimator. The gray lines in the figures show the connection between the simplex spline model of degree d and the corresponding polynomial model of degree d on the same dataset.

Three very important conclusions can be drawn from Figure 5.44, Figure 5.45, Figure 5.46, Figure 5.47, Figure 5.48, and Figure 5.49. First, the simplex spline based models invariably are of higher quality than ordinary polynomial models if the set of candidate dimensions is adequately chosen. A good example of this is the plot for the pitching moment coefficient C_m. In this plot, almost all simplex spline models are of a significantly higher quality than polynomial models of the same degree on the same set of dimensions.

The second conclusion is that if the set of model dimensions is not adequately chosen, then the simplex spline models produce highly inadequate results. In this case, the polynomial models outperform the spline models, sometimes by many orders of magnitude. An example of this is the plot for C_X in Figure 5.44. Most spline models at the right side of the plot, which corresponds with models of high dimension, produce low quality results and are outperformed by simple polynomials of the same total degree. This can be explained as follows. A simplex spline function has an approximation power that is proportional with the total number of simplices on which it is defined. In this sense, a simple polynomial model is exactly equal to a simplex spline polynomial on a single simplex, a fact that was proved in Sec. 3.5. If an inadequate set of dimensions is chosen, the B-coefficient estimator still finds an optimal solution which in most cases accurately fits the identification dataset. On the validation dataset, however, this same spline function will produce wildly varying results. This situation is demonstrated in the center of the plot for C_l in Figure 5.47, where there is 1 spline function which achieves an RMS_{rel} value of around $10^{-1.85}$. This is a suspicious result, because the other spline functions of the same dimension but with different polynomial degrees produce validation errors which are significantly larger.

This leads to the third conclusion, which is that high quality spline functions defined on an adequately chosen set of dimensions will produce validation results which between different spline degrees do not differ by more than a factor 5 (± 0.7 orders of magnitude). Furthermore, increasing the degree of a spline function should reduce the validation error, at least up to some point. On the other hand, increasing the continuity of the spline function should somewhat increase the validation error, because a higher continuity order results in a lower approximation power, as was explained in Sec. 3.3.3.

5.5.5 Prototype phase final model structure selection

Based on the results from the prototype model estimation, the best performing spline model structures are selected. For most force and moment coefficients, however, there remain two or more viable geometric model structure candidates, see Table 5.14. In the final selection procedure, a single geometric model structure will be selected for each force and moment coefficient. Additionally, the specific degree and continuity order of the spline model will

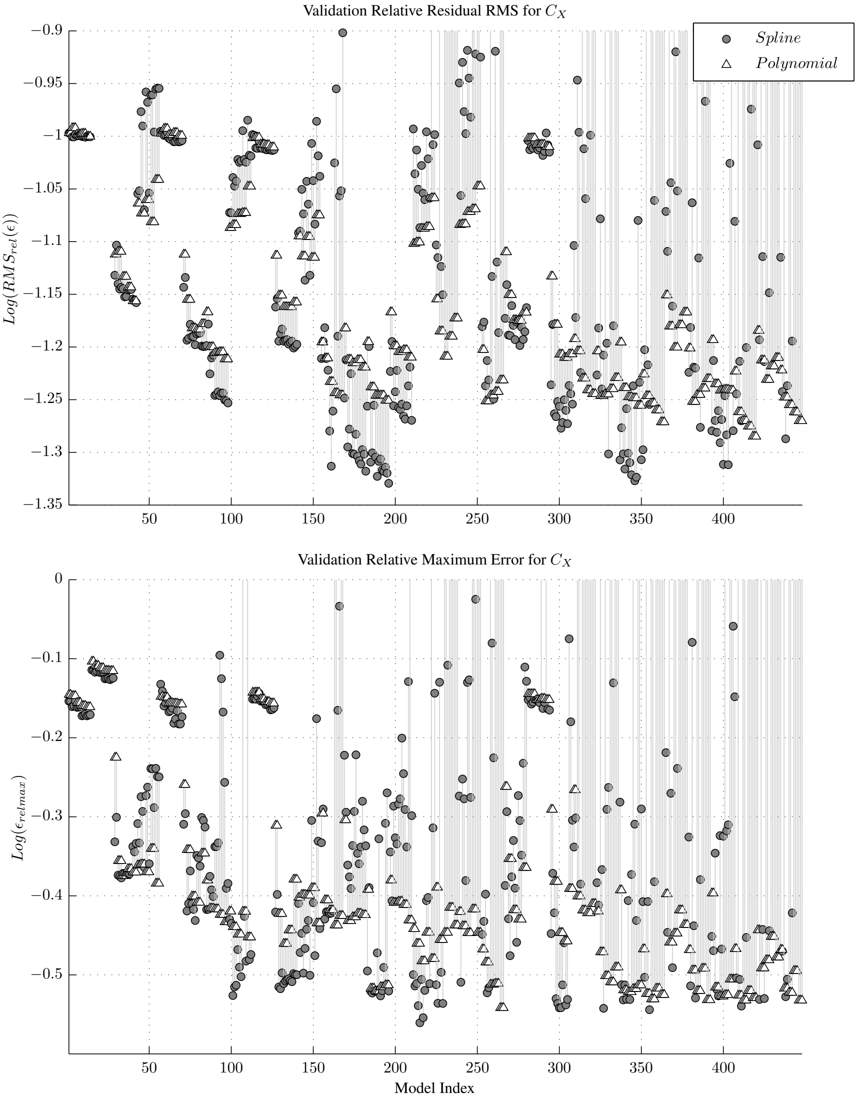


Figure 5.44: A qualitative comparison between the relative residual RMS (top) and the relative maximum error (bottom) in the model validation results of all 448 spline based aerodynamic models for C_X. The performance is compared with polynomials of the same dimension and total degree as the spline models. A score of -1 corresponds with a 90% accurate model, while a score of -1.5 corresponds with a 97% accurate model.

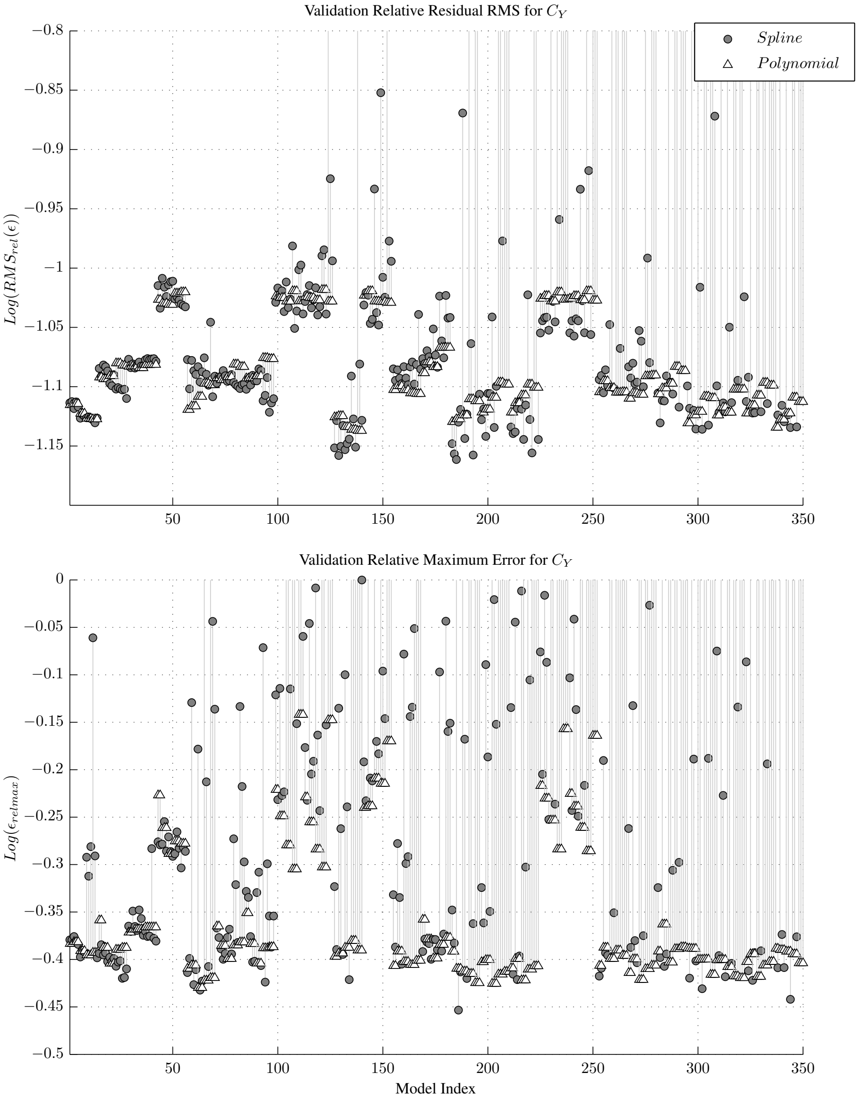


Figure 5.45: A qualitative comparison between the relative residual RMS (top) and the relative maximum error (bottom) in the model validation results of all 350 spline based aerodynamic models for C_Y. The performance is compared with polynomials of the same dimension and total degree as the spline models. A score of -1 corresponds with a 90% accurate model, while a score of -1.5 corresponds with a 97% accurate model.

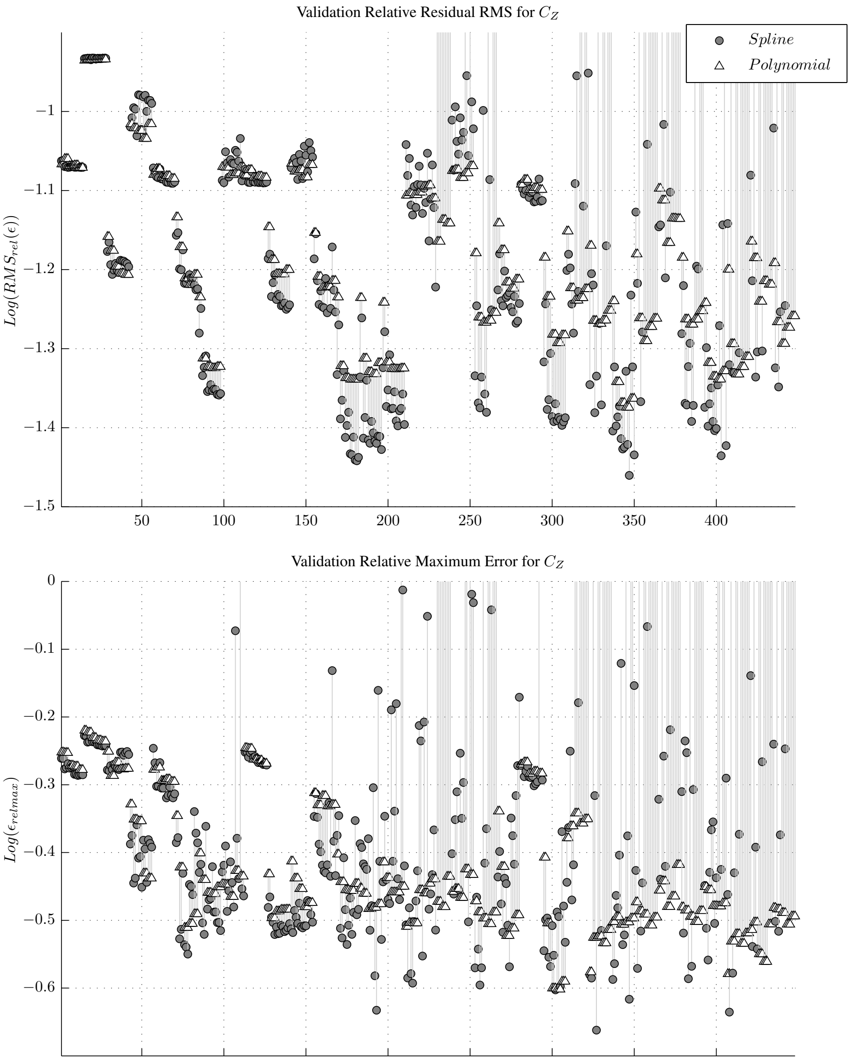


Figure 5.46: A qualitative comparison between the relative residual RMS (top) and the relative maximum error (bottom) in the model validation results of all 448 spline based aerodynamic models for C_Z. The performance is compared with polynomials of the same dimension and total degree as the spline models. A score of -1 corresponds with a 90% accurate model, while a score of -1.5 corresponds with a 97% accurate model.

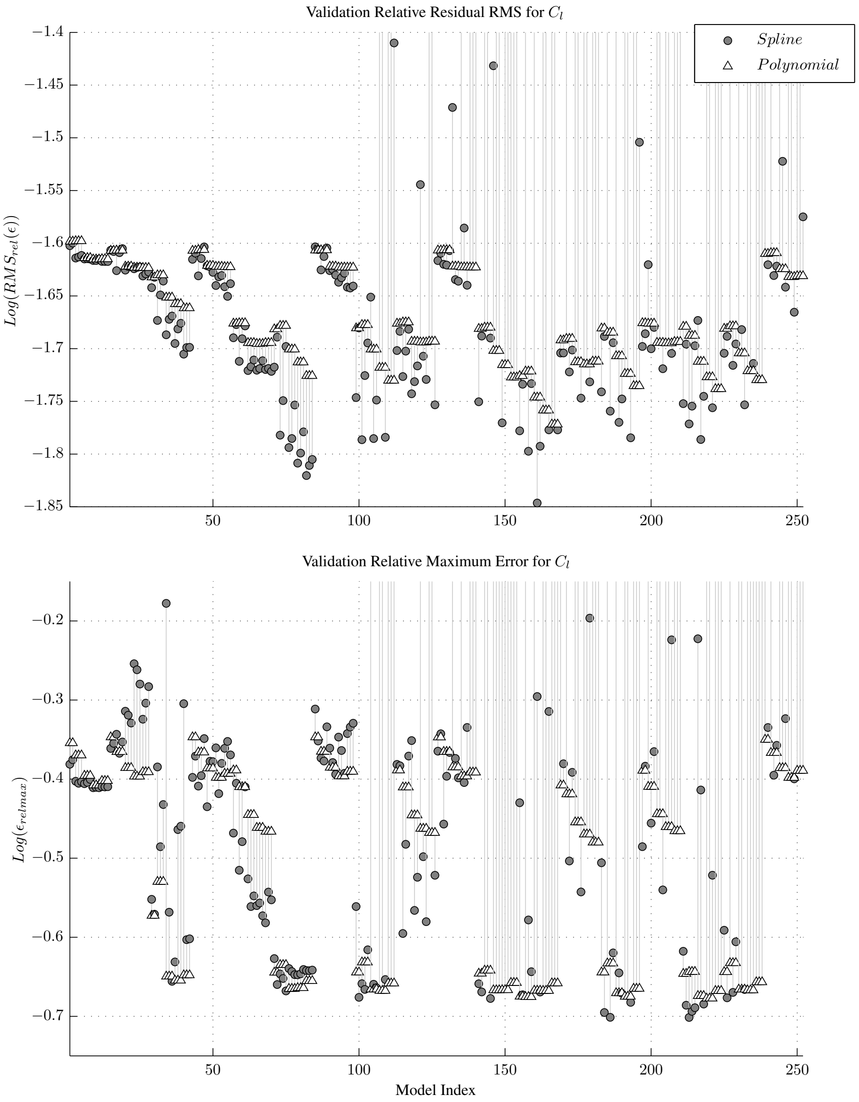


Figure 5.47: A qualitative comparison between the relative residual RMS (top) and the relative maximum error (bottom) in the model validation results of all 252 spline based aerodynamic models for C_l. The performance is compared with polynomials of the same dimension and total degree as the spline models. A score of -1 corresponds with a 90% accurate model, while a score of -1.5 corresponds with a 97% accurate model.

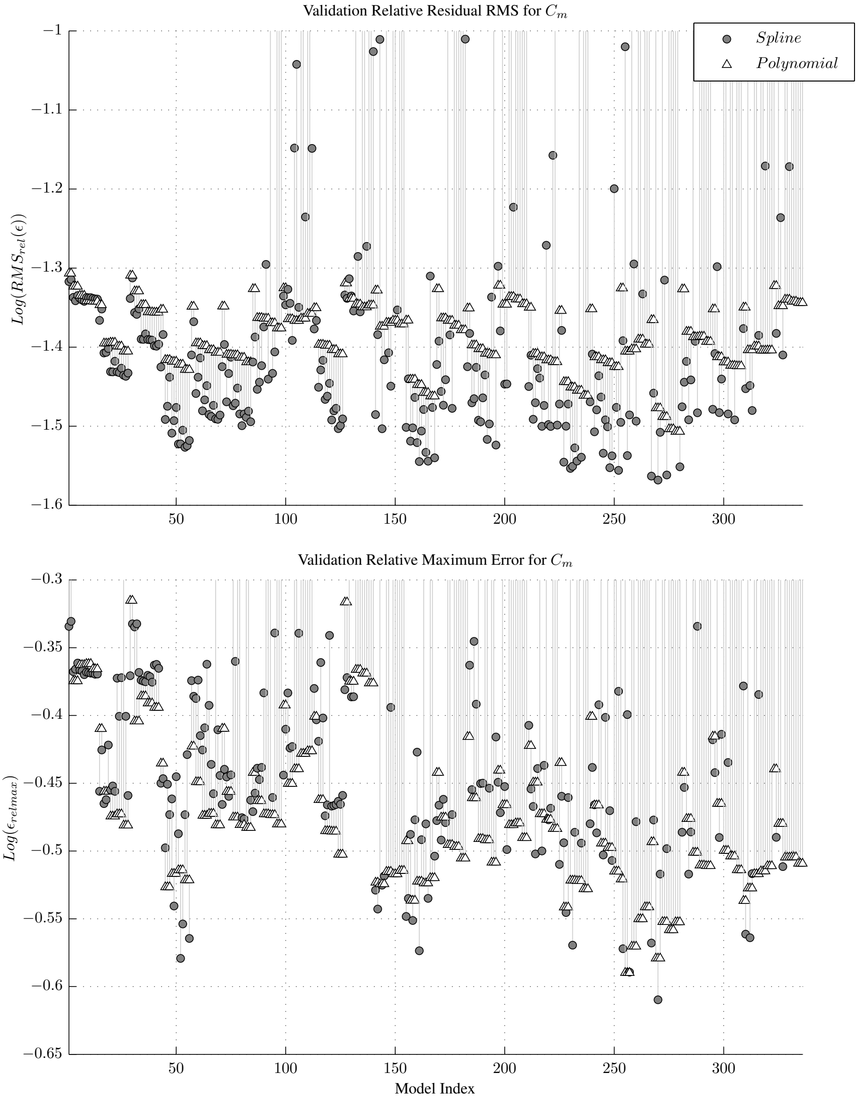


Figure 5.48: A qualitative comparison between the relative residual RMS (top) and the relative maximum error (bottom) in the model validation results of all 336 spline based aerodynamic models for C_m. The performance is compared with polynomials of the same dimension and total degree as the spline models. A score of -1 corresponds with a 90% accurate model, while a score of -1.5 corresponds with a 97% accurate model.

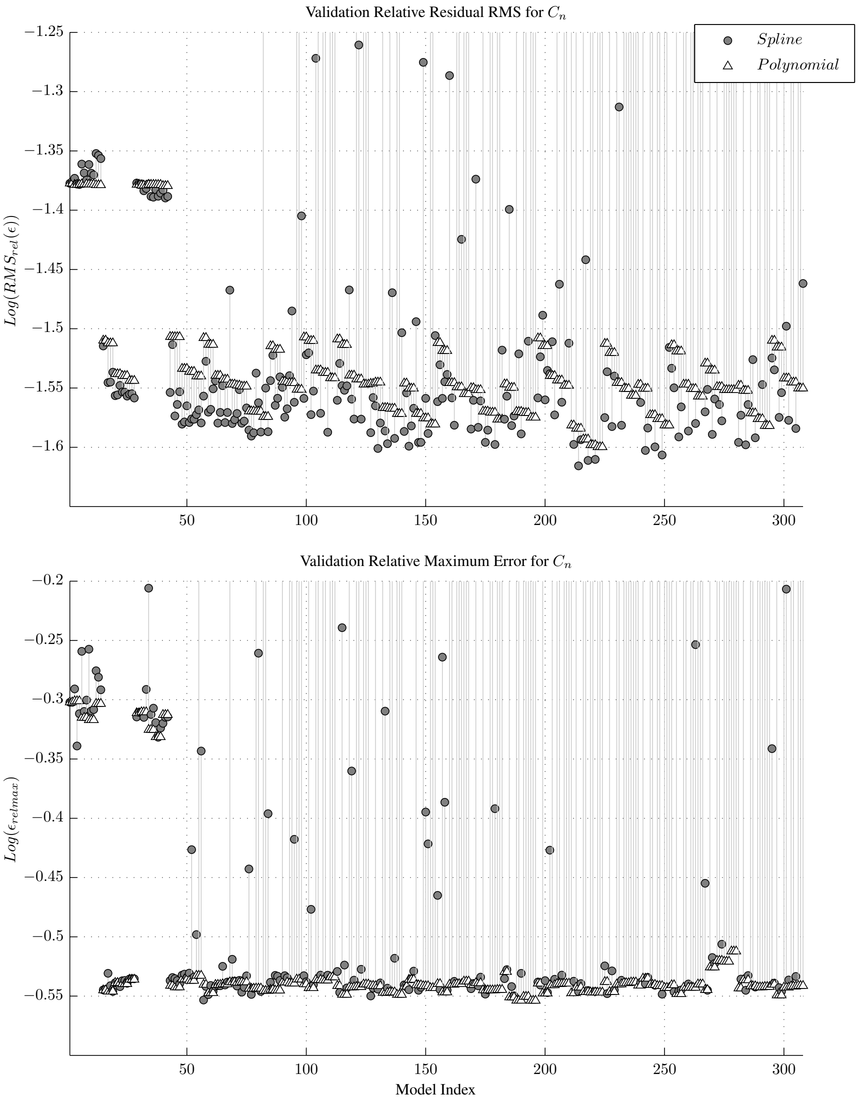


Figure 5.49: A qualitative comparison between the relative residual RMS (top) and the relative maximum error (bottom) in the model validation results of all 308 spline based aerodynamic models for C_n. The performance is compared with polynomials of the same dimension and total degree as the spline models. A score of -1 corresponds with a 90% accurate model, while a score of -1.5 corresponds with a 97% accurate model.

Table 5.14: The selected geometric model structures for the decoupled aerodynamic models.

Coefficient	Candidate Dimension Sets for Refinement
C_X	(α, M, δ_e), $(\alpha, \hat{q}, M, \delta_e)$
C_Y	$(\beta, \hat{r}, \delta_r)$
C_Z	$(\alpha, \hat{q}, \delta_e)$, $(\alpha, \hat{q}, M, \delta_e)$
C_l	$(\beta, \hat{p}, \delta_a)$, $(\beta, M, \hat{p}, \hat{r}, \delta_a)$
C_m	$(\alpha, \beta, \hat{q}, \delta_e)$, $(\alpha, \dot{\alpha}, \beta, \delta_e)$, $(\alpha, \beta, \hat{q}, M, \delta_e)$
C_n	$(\beta, M, \hat{r}, \delta_r)$, $(\beta, M, \hat{p}, \hat{r}, \delta_r)$

be determined. This determination will be made on the basis of a number of criteria. The first criterion is that the quality of the model should improve with increasing spline degree, at least up to a certain degree. The second criterion is that the model quality should be consistently higher than that of ordinary polynomials of the same degree. The final criterion is that a model of higher dimension will only be selected if it produces results of significantly higher quality than models of lower dimension.

During this final selection procedure, an additional set of spline spaces was used to help determine the optimal polynomial degree for the model. The additional spline spaces are $\mathcal{S}_5^3$, $\mathcal{S}_6^{-1}$, $\mathcal{S}_6^0$, $\mathcal{S}_6^1$, $\mathcal{S}_6^2$, and $\mathcal{S}_6^3$.

In the following, the final model structure selection procedure for each aerodynamic force and moment coefficient will be discussed. It is important to note that the results shown in the following paragraphs are the results from model validation, and not from model identification.

5.5.5.1 C_X final model structure selection

The validation performance of the remaining two spline model structure candidates for the longitudinal force coefficient C_X is plotted in Figure 5.50 and Figure 5.51. A comparison between these two plots shows that the 3-dimensional spline models produce lower (=better) overall RMS_{rel} values than the 4-dimensional models. The inclusion of the dimensionless pitch rate does not significantly improve model quality. The selected spline space is a fourth degree model with second order continuity between the spline pieces. This spline space is selected because it produced one of the lowest RMS_{rel} values, while also producing one of the lowest maximum relative error values. The selected spline model for C_X therefore is $f(\alpha, M, \delta_e) \in \mathcal{S}_4^2$.

5.5.5.2 C_Y final model structure selection

The lateral force coefficient C_Y proved to be the most challenging coefficient to model. It is hypothesized that this is the result of insufficient excitation of the aircraft in the Y_B direction. The model structure that produced adequate results contained the dimensions β,

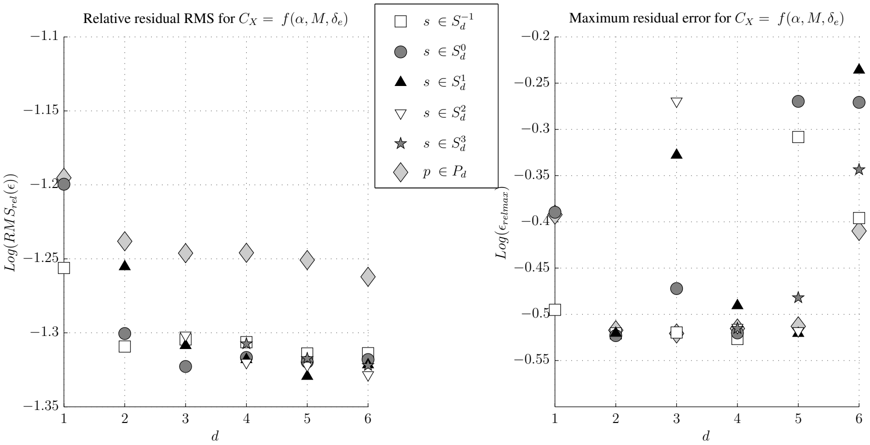


Figure 5.50: Validation results for the spline model $f(\alpha, M, \delta_e)$ for C_X. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

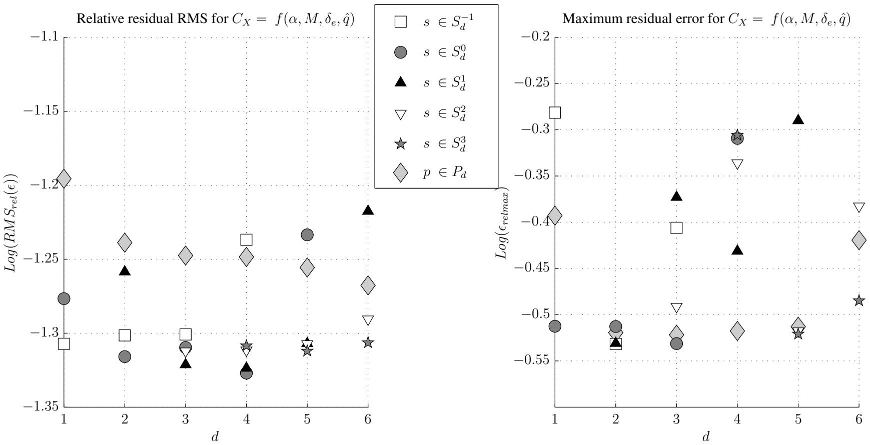


Figure 5.51: Validation results for the spline model $f(\alpha, M, \delta_e, \hat{q})$ for C_X. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

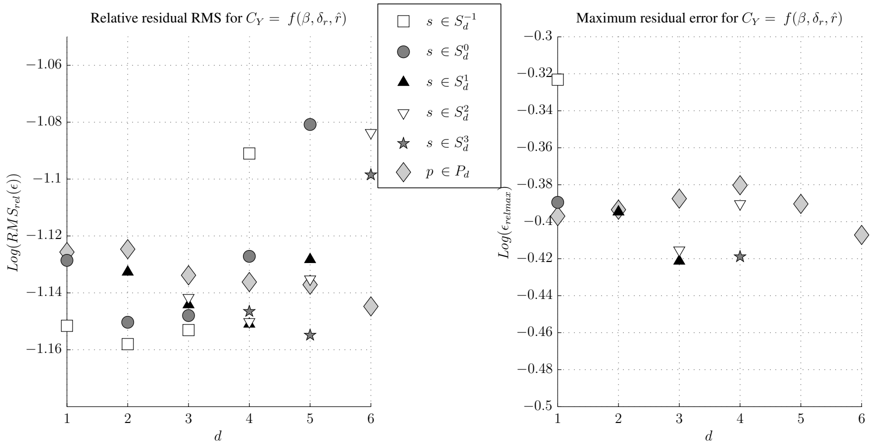


Figure 5.52: Validation results for the spline model $f(\beta, \delta_r, \hat{r})$ for C_Y. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

$\hat{r}$, and δ_r. In Figure 5.52 the validation results of the highest scoring dimension candidate are shown. The selected spline space for C_Y is $f(\beta, \delta_r, \hat{r}) \in \mathcal{S}_3^1$, a third degree spline space with second order continuity. This spline space produced adequate RMS_{rel} values, as well as the lowest relative maximum error.

5.5.5.3 C_Z final model structure selection

In Figure 5.53 and Figure 5.54 the validation results of the two best performing spline model structure candidates for the vertical force coefficient C_Z are shown. In this case the 4-dimensional models $f(\alpha, M, \delta_e, \hat{q})$ produce significantly higher quality validation results than the 3-dimensional models. From the 4-dimensional spline spaces, the spline space $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ is selected for refinement. This particular spline space produced high RMS_{rel} scores as well as high relative maximum error scores.

5.5.5.4 C_l final model structure selection

In Figure 5.55 and Figure 5.56 the validation results of the two remaining spline model structure candidates for the aerodynamic rolling moment coefficient C_l are shown. The first candidate is the 3-dimensional set $(\beta, \delta_a, \hat{p})$, and the second the 5-dimensional set $(M, \beta, \delta_a, \hat{p}, \hat{r})$. The figures show that the validation results of some of the 5-dimensional spline functions are of significantly higher quality than the 3-dimensional spline functions. However, a number of spline spaces produce very low quality results, most notably the spline spaces with low continuity orders. This points to numerical issues which find their

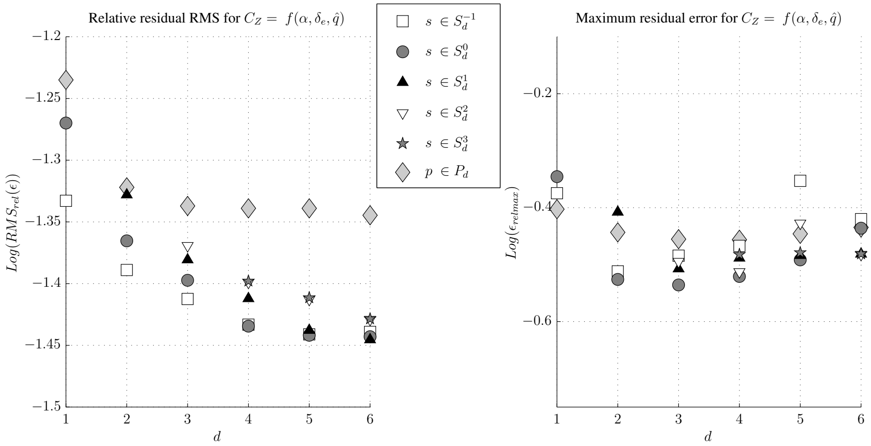


Figure 5.53: Validation results for the spline model $f(\alpha, \delta_e, \hat{q})$ for C_Z. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

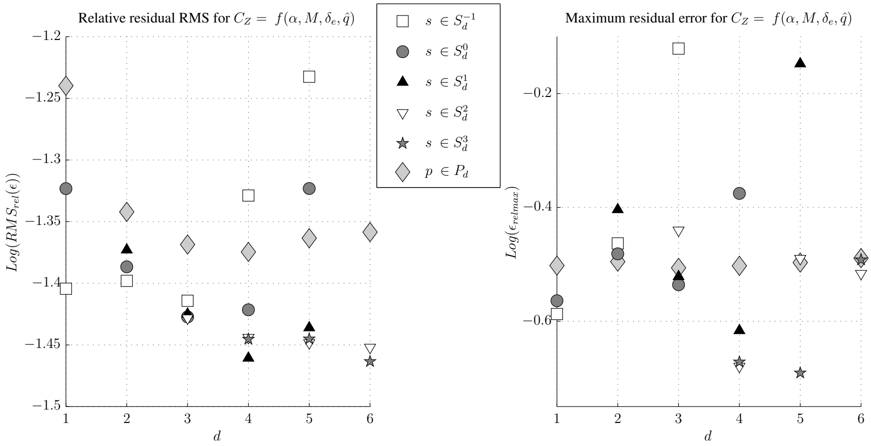


Figure 5.54: Validation results for the spline model $f(\alpha, M, \delta_e, \hat{q})$ for C_Z.

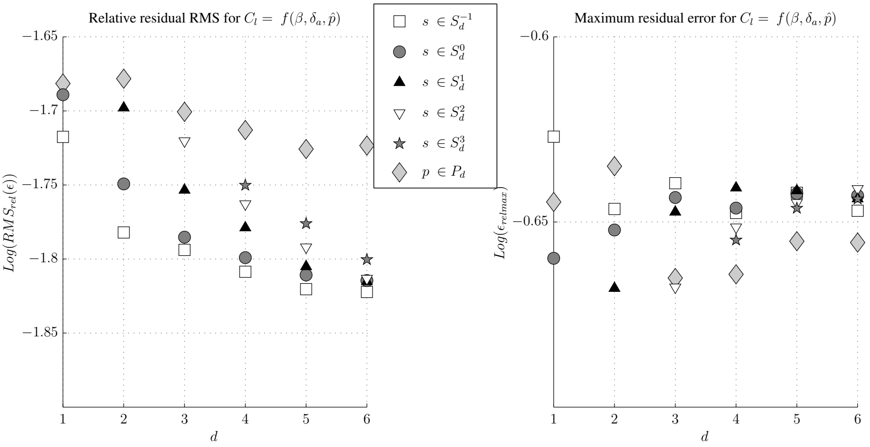


Figure 5.55: Validation results for the spline model $f(\beta, \delta_a, \hat{p})$ for C_l. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

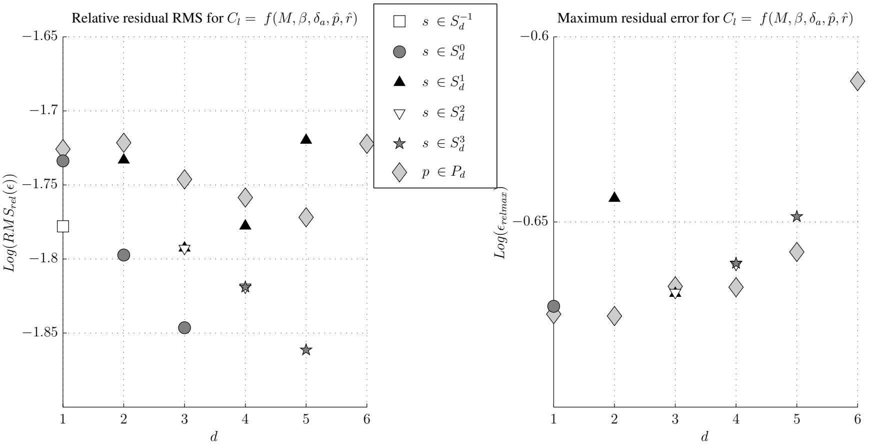


Figure 5.56: Validation results for the spline model $f(M, \beta, \delta_a, \hat{p}, \hat{r})$ for C_l. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

cause in the data coverage problem. It was found earlier that the data coverage problem is more severe for higher dimensional splines. Given the current dataset, the spline function $f(\beta, \delta_a, \hat{p}) \in \mathcal{S}_4^2$ is selected for further refinement.

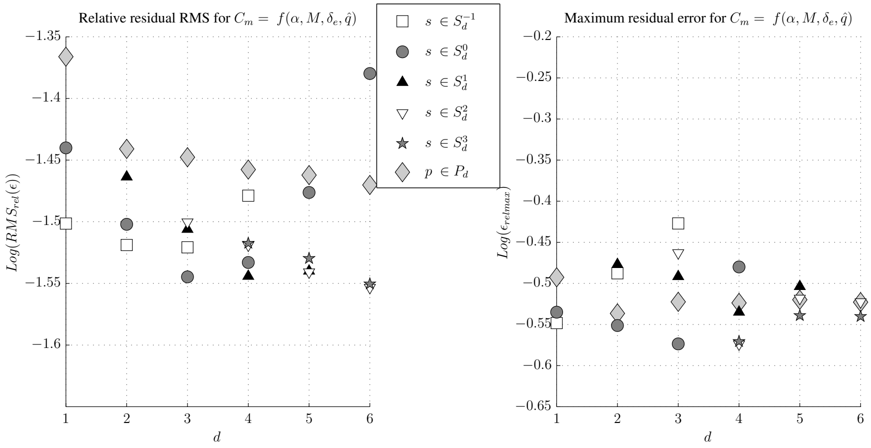


Figure 5.57: Validation results for the spline model $f(\alpha, M, \delta_e, \hat{q})$ for C_m. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

5.5.5.5 C_m final model structure selection

For the pitching moment coefficient C_m, three candidate model structures remain in the form of the two 4-dimensional sets $(\alpha, M, \delta_e, \hat{q})$ and $(\alpha, \beta, \delta_e, \hat{q})$, and the 5-dimensional set $(\alpha, M, \beta, \delta_e, \hat{q})$. The validation results for these three candidate model structures is shown in Figure 5.57, Figure 5.58, and Figure 5.59. From these figures it can be seen that the 5-dimensional model structure produces adequate results, but only for a subset of spline spaces. Again, this is the result of numerical issues caused by the data coverage problem. The two 4-dimensional models consistently produce adequate quality models. Of these two remaining candidates, the set $(\alpha, M, \delta_e, \hat{q})$ results in lower relative maximum error values. From Figure 5.57 it can be seen that the $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ spline function produces adequate values for RMS_{rel} and for ϵ_{relmax}, leading to its selection as the refinement candidate for C_m

5.5.5.6 C_n final model structure selection

In Figure 5.60 and Figure 5.61 the validation results from the modeling of the yawing moment coefficient C_n are shown. In these figures, the $f(M, \beta, \delta_r, \hat{r})$ and $f(M, \beta, \delta_r, \hat{p}, \hat{r})$ spline models are plotted. From the figure, it can be seen that the 5-dimensional model produces adequate quality results, but only for lower degree models and high relative continuity orders. This indicates that there are numerical problems in the estimator, which again is caused by insufficient per-simplex data coverage. While 4-dimensional spline models are of somewhat lower quality, the overall results are more consistent. From the

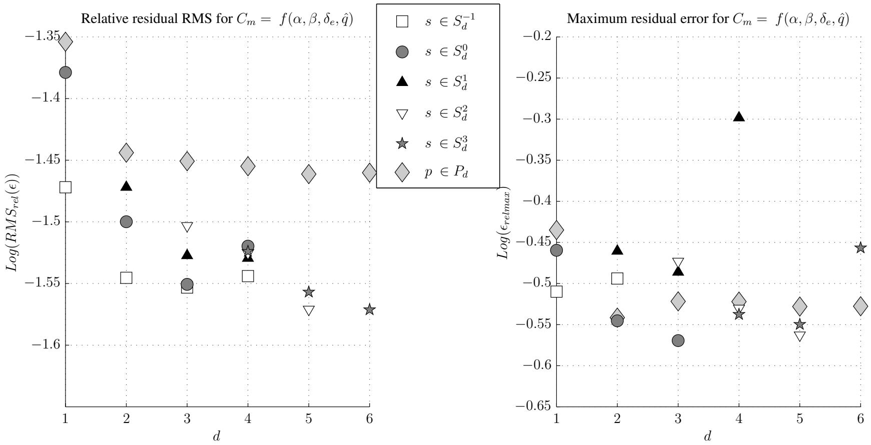


Figure 5.58: Validation results for the spline model $f(\alpha, \beta, \delta_e, \hat{q})$ for C_m. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

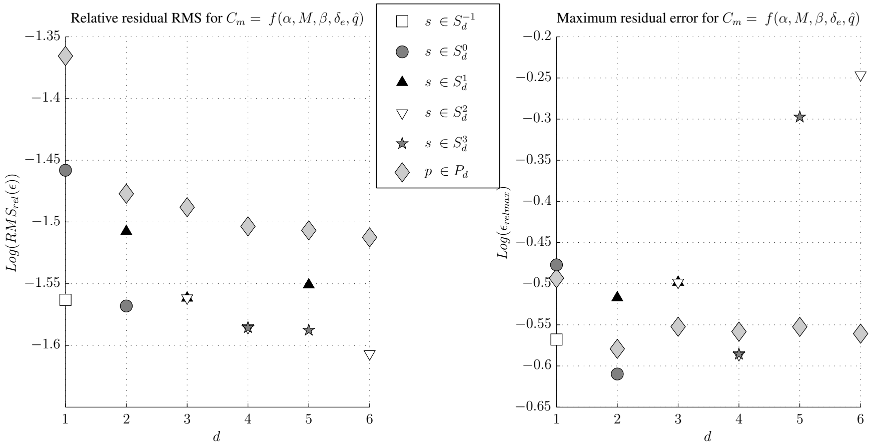


Figure 5.59: Validation results for the spline model $f(\alpha, M, \beta, \delta_e, \hat{q})$ for C_m. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

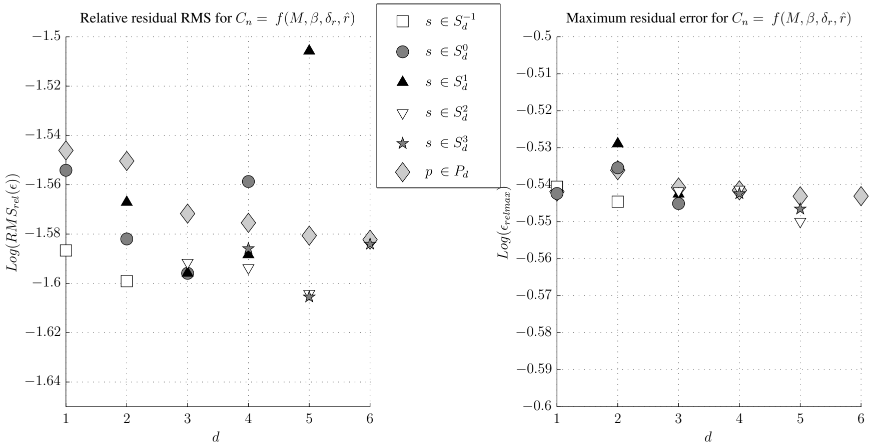


Figure 5.60: Validation results for the spline model $f(M, \beta, \delta_r, \hat{r})$ for C_n. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

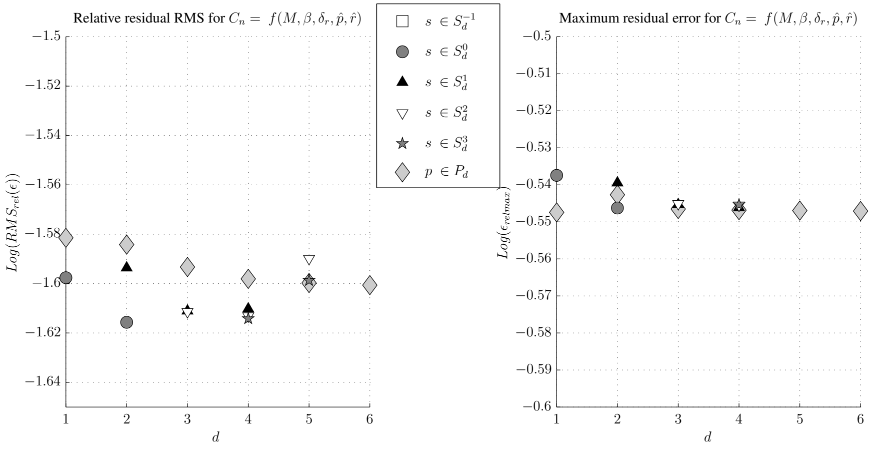


Figure 5.61: Validation results for the spline model $f(M, \beta, \delta_r, \hat{p}, \hat{r})$ for C_n. The ordinary polynomial models are indicated as $p \in \mathcal{P}_d$.

4-dimensional set of spline spaces the spline function $f(M, \beta, \delta_r, \hat{r}) \in \mathcal{S}_4^2$ is selected for further refinement.

Table 5.15: The results from the model refinement phase.

ADM	Spline	Initial Triangulation	Final Triangulation
C_X	$f(\alpha, M, \delta_e) \in \mathcal{S}_4^2$	$\mathcal{T}_6$	$\mathcal{T}_6$
C_Y	$f(\beta, \delta_r, \hat{r}) \in \mathcal{S}_3^1$	$\mathcal{T}_6$	$\mathcal{T}_6$
C_Z	$f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$	$\mathcal{T}_{24}$	$\mathcal{T}_{22}$
C_l	$f(\beta, \delta_a, \hat{p}) \in \mathcal{S}_4^2$	$\mathcal{T}_6$	$\mathcal{T}_6$
C_m	$f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$	$\mathcal{T}_{24}$	$\mathcal{T}_{22}$
C_n	$f(M, \beta, \delta_r, \hat{r}) \in \mathcal{S}_4^2$	$\mathcal{T}_{24}$	$\mathcal{T}_{22}$

5.5.6 Model refinement

The prototype phase of the aerodynamic model identification experiment is now complete, resulting in the selection of the spline dimensions and spline spaces shown in Table 5.15. The refinement phase of spline model estimation can commence. During model refinement, it was found that data coverage was by far the most important factor for the quality of a spline model. In most cases, the spline models became of higher quality after simplices with insufficient data coverage were actually removed from the triangulation, see Table 5.15. The data coverage problem effectively prevented the creation of triangulations of significantly higher resolution than that of the prototype hypercube triangulation. Perhaps the most important recommendation made in this thesis is that a software tool should be developed for checking, during flight, the data coverage of candidate triangulations.

The differential constraints from Sec. 3.4.1 were imposed on the edges of the triangulations to reduce polynomial divergence. The differential constraints effectively enabled the successful estimation of spline functions for triangulations with inadequate local data coverage, thereby reducing the severity of the data coverage problem.

5.6 Spline Model Validation

A model has no value without having been extensively validated. All spline models created in Sec. 5.5 are put through the model validation steps presented in Sec. 4.3 to confirm their quality. These steps include a global model inspection and flight dynamics analysis, a B-coefficient stability analysis, a model residue analysis and a number of statistical tests. Some of these quality methods, like the spatial B-coefficient variance analysis and the B-coefficient stability analysis, are unique to simplex spline based models. Together, these analysis methods provide a complete overview of the quality of a multivariate simplex spline model.

The first analysis is an inspection of the raw validation results and model residuals, which are discussed in Sec. 5.6.1. Following this, a global model inspection is performed in Sec. 5.6.2. In Sec. 5.6.3 the results from a model residue analysis are presented. The stability of the spline models is analyzed in Sec. 5.6.4 based on the values of the B-coefficients of the simplex splines. Subsequently, in Sec. 5.6.5 a statistical model quality analysis is performed using the tools and techniques presented earlier in Sec. 4. This chapter is summarized in Sec. 5.6.6 with a conclusion of the aerodynamic model identification experiment.

5.6.1 Raw validation results

The raw results of the model validation give a more complete indication of the quality of the multivariate spline based aerodynamic models than the global quality assessment discussed in Sec. 5.5. These results will now be discussed for the final six selected aerodynamic models from Sec. 5.5.6.

First, in Figure 5.62 the results from the validation of the 4-dimensional spline model for the longitudinal force coefficient C_X are shown. The results show that the model only produces a globally accurate fit to the validation data, while failing to accurately predict local details ($RMS_{rel} = 4.79\%$). This can be explained by the fact that all force coefficients, including C_X, are highly sensitive to errors in the estimate of aircraft mass. Additionally, engine thrust was not a model dimension. While many prototype models were created that included engine thrust as a dimension (see Table 5.10), none produced models of adequate quality. The most likely cause for this is the fact that there was very little (if any) excitation in the direction of aircraft thrust.

In Figure 5.63 the validation results of the spline model for the lateral force coefficient C_Y are shown. This spline model produces reasonable quality validation results for most of the validation maneuvers. However, the spline model predictions on C_Y for a number of maneuvers at the beginning of the validation sequence in Figure 5.63 show significant discrepancies with the measured C_Y, leading to low scores for model quality($RMS_{rel} = 7.18\%$).

The validation results of the spline model for the vertical force coefficient C_Z are shown in Figure 5.64. Of all aerodynamic force coefficients, C_Z proved to be the least challenging to model accurately ($RMS_{rel} = 3.6\%$). As in the raw validation results for C_X and C_Y, significant offsets between model prediction and measured values are present. Again, it is assumed that this is caused mainly by errors in the estimation of aircraft mass. This assumption is strengthened by the fact that the most significant prediction errors of the models for C_X, C_Y, and C_Z occur at the same maneuvers. For example, all force predictions in the sample interval $[0 \;\; 2 \cdot 10^4]$, which contains 4 maneuvers, show significant errors.

The validation results of the spline model for the rolling moment coefficient C_l are shown in Figure 5.65. These validation results show that the model for C_l is of adequate quality for high amplitude maneuvers, like the maneuver shown in *Closeup*#2. For lower amplitude maneuvers, however, the spline model consistently underestimates C_l, leading to a less than adequate overall quality score ($RMS_{rel} = 7.18\%$). It is hypothesized that this is

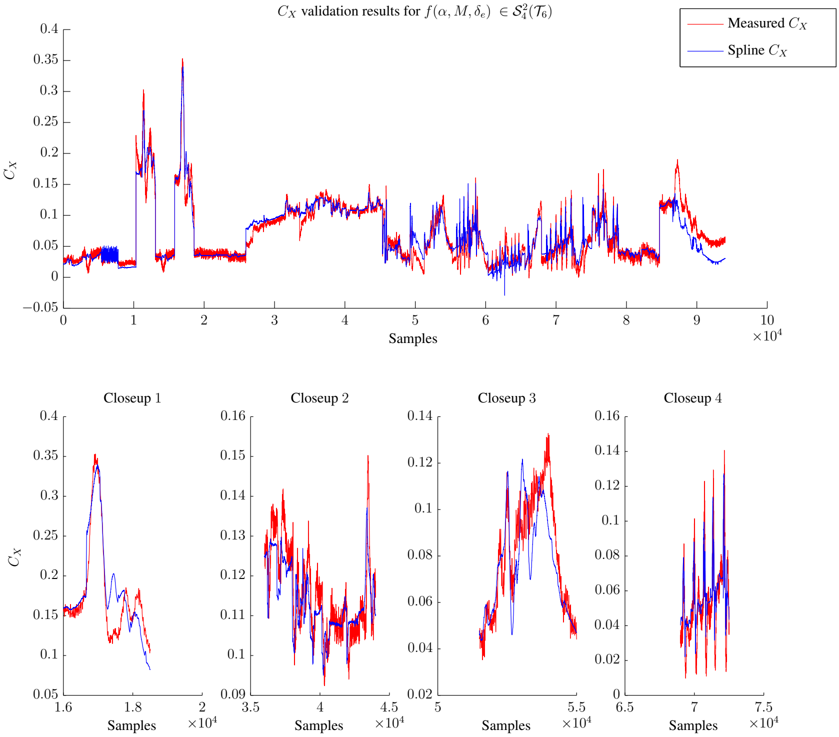


Figure 5.62: Validation results for the spline model $f(\alpha, M, \delta_e) \in \mathcal{S}_4^2$ for C_X.

caused by the fact that the Mach number is not a dimension of this particular model. During the prototype phase, models for C_l which included the Mach number were created, but all suffered from numerical issues during estimation.

In Figure 5.66 the results of the validation of the spline model for C_m on the validation maneuvers is shown. This spline model is of adequate quality, especially for high amplitude maneuvers ($RMS_{rel} = 3.03\%$). Like the models for the aerodynamic forces, the spline model for C_m produces inadequate predictions in the interval $[0 \;\; 2 \cdot 10^4]$, strengthening the argument that the mass model produces inaccurate predictions, in this case of the moments of inertia.

Finally, the validation results of the spline model for the yawing moment coefficient C_n are shown in Figure 5.67. This model is of adequate overall quality ($RMS_{rel} = 2.55\%$), although some high amplitude maneuvers are not adequately modeled. It is worth noting

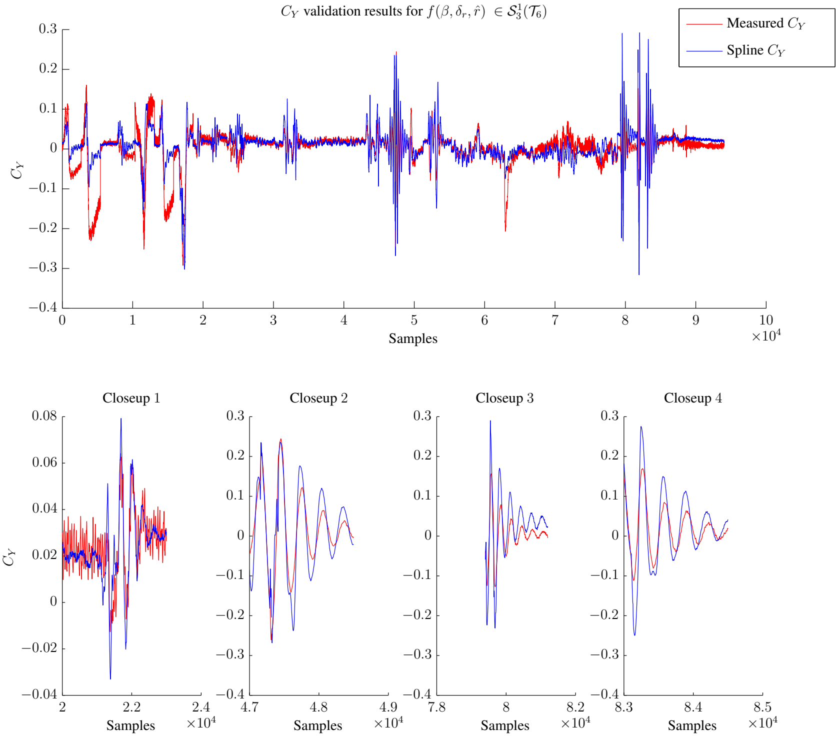


Figure 5.63: Validation results for the spline model $f(\beta, \delta_r, \hat{r}) \in \mathcal{S}_3^1$ for C_Y.

that the measured values for C_n contains significant levels of (sensor) noise, which may mask low magnitude features.

5.6.2 Global model inspection

The global quality of a spline model can be analyzed best by evaluating the spline function on a number of grid points spaced equally throughout the spline model domain. This analysis helps to determine the overall shape and smoothness of the spline models, which in turn enable an interpretation of the models from a flight dynamics perspective. The plots shown in this paragraph consist of *slices* through the multi-dimensional models, as the higher dimensional models themselves cannot be visualized directly. Additionally, the slices have been trimmed to show only those regions of the spline domain that are inside the

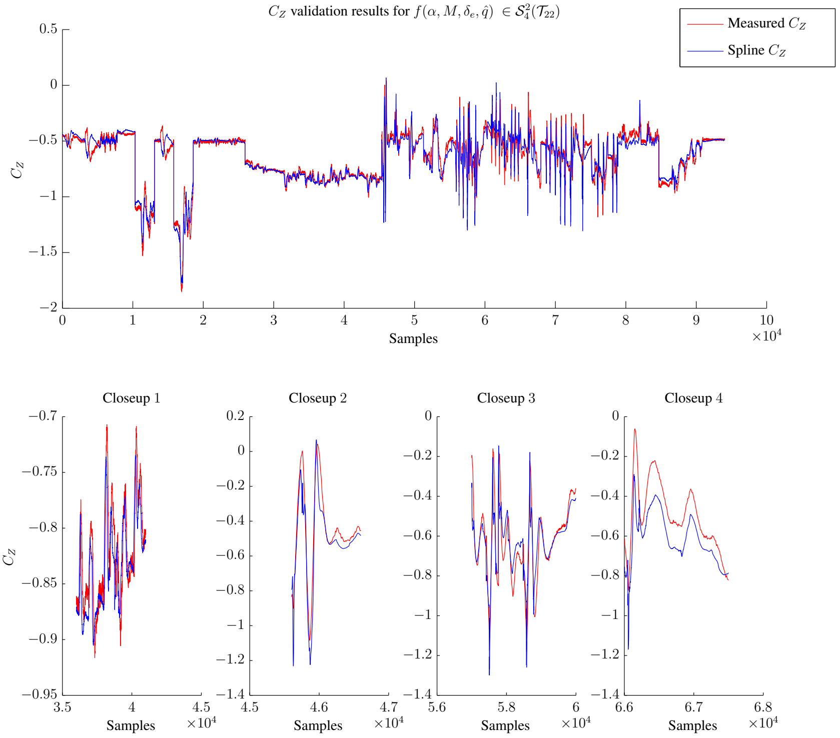


Figure 5.64: Validation results for the spline model $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_Z.

convex hull of the identification dataset. This ‘validity region’ was defined in Sec. 5.18, and shown in Figure 5.20.

In Figure 5.68 four 2-dimensional slices through the domain of the spline function for C_X are shown. The slices in this case are made along the elevator deflection (δ_e) axis. The surfaces shown in the individual plots thus show the effects of a constant elevator deflection and varying Mach number and angle of attack. An important detail in Figure 5.68 is that the slope of $f(\alpha, M, \delta_e)$ in the direction of α is positive, i.e. $C_{X_\alpha} > 0$, which is an expected result.

A global overview of the 3-dimensional spline model for C_Y is shown in Figure 5.69. In this case 4 slices along the rudder deflection (δ_r) axis are plotted, where the surfaces show the effect of β and $\hat{r}$ on C_Y. In this case the slope of $f(\beta, \delta_r, \hat{r})$ in the direction of β is negative, i.e. $C_{Y_\beta} < 0$, which again is an expected result.

A total of eight 2-dimensional slices through the 4-dimensional model for C_Z are shown

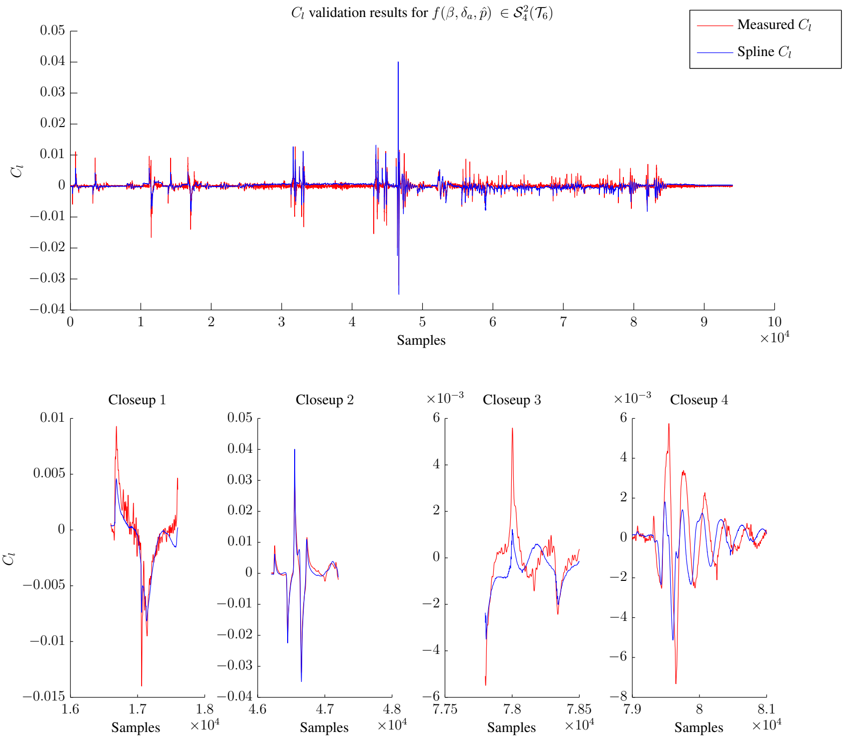


Figure 5.65: Validation results for the spline model $f(\beta, \delta_a, \hat{p}) \in \mathcal{S}_4^2$ for C_l.

in Figure 5.70 and Figure 5.71. The slices in this case are made along the elevator (δ_e) and dimensionless pitch rate ($\hat{q}$) axes. The slope of $f(\alpha, M, \delta_e, \hat{q})$ in the direction of α is negative, i.e. $C_{Z_\alpha} < 0$. However, as the angle of attack increases, C_{Z_α} becomes less negative, indicating that the upper part of the lift slope has been reached.

In Figure 5.72 four slices along the δ_a axis through the 3-dimensional model for the rolling moment coefficient C_l are shown. It can be observed that C_{l_β} is negative, but only slightly so. Only at large angles of sideslip does C_{l_β} become significantly negative; C_{l_β} clearly is a nonlinear function in β.

A global overview of the 4-dimensional model for the aerodynamic pitching moment coefficient C_m is shown in Figure 5.73 and Figure 5.74. The figures show eight slices through the model for C_m along the dimensionless pitch rate ($\hat{q}$) and the elevator deflection (δ_e) axes. The figures show that the slope of $f(\alpha, M, \delta_e, \hat{q})$ with respect to α is strongly negative, i.e. $C_{m_\alpha} < 0$. This is an expected result for a statically stable aircraft like the

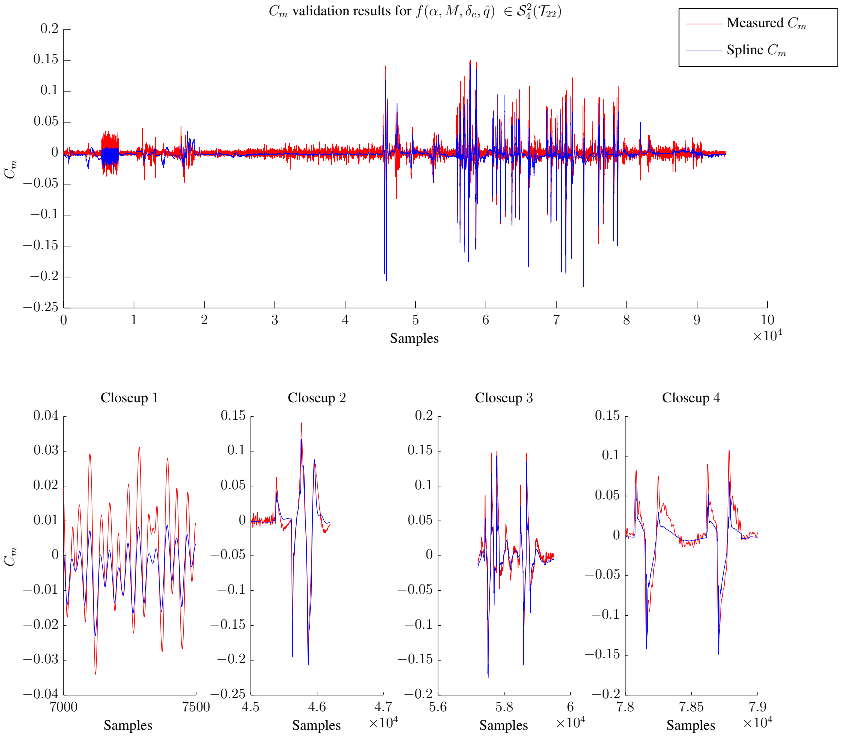


Figure 5.66: Validation results for the spline model $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_m.

Cessna Citation II. Additionally, it can be seen from the figure that C_m becomes more negative with a more positive elevator deflection. This implies that $C_{m_{\delta_e}} < 0$, which is again an expected result.

Finally, in Figure 5.75 and Figure 5.76 eight slices along the rudder deflection and dimensionless yaw rate axes of the 4-dimensional model for the yawing moment coefficient C_n are shown. From the figure it can be seen that the slope of $f(M, \beta, \delta_r, \hat{r})$ in the direction of β is positive, that is, $C_{n_\beta} > 0$. For an aircraft with a damped Dutch Roll motion, like the Cessna Citation II, this is an expected result. The rudder scales the complete model output, with a more positive rudder deflection leading to a more negative C_n, i.e. $C_{n_{\delta_e}} < 0$.

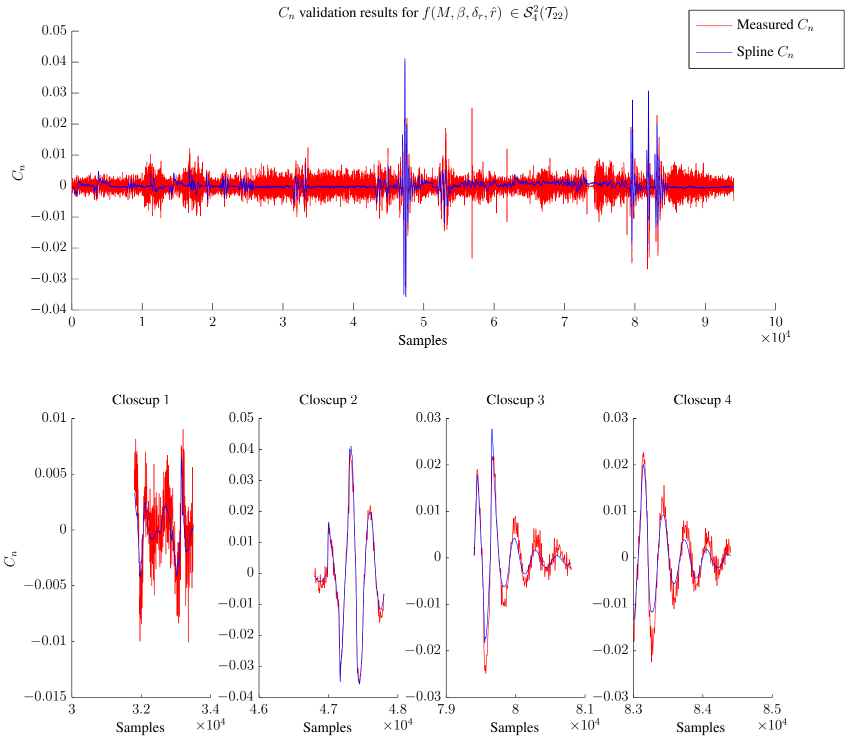


Figure 5.67: Validation results for the spline model $f(M, \beta, \delta_r, \hat{r}) \in \mathcal{S}_4^2$ for C_n.

5.6.3 Model residual analysis

In Sec. 4.3 a spatial model residual analysis method based on Chebyshev's inequality was introduced. This analysis method results in empirical confidence intervals for the B-coefficients of multivariate simplex splines. The empirical confidence intervals use the identification model residual together with Chebyshev's inequality, which is repeated here for clarity:

$$Pr(|\epsilon(\mathcal{X}_{id}) - \mu(\epsilon(\mathcal{X}_{id}))| \geq a\sigma) \leq \frac{1}{a^2}, \quad a \geq 1,$$

with $\epsilon(\mathcal{X}_{id})$ the identification residual equivalent to Eq. 5.70, with σ the local standard deviation of the model residue, and with a some constant. The constant a can now be chosen such that a predefined fraction of all model residuals are inside the confidence bounds. In this thesis, the requirement was made that at least 97% of all model residuals should be inside the model confidence bounds, resulting in $a = 6$. It should be noted

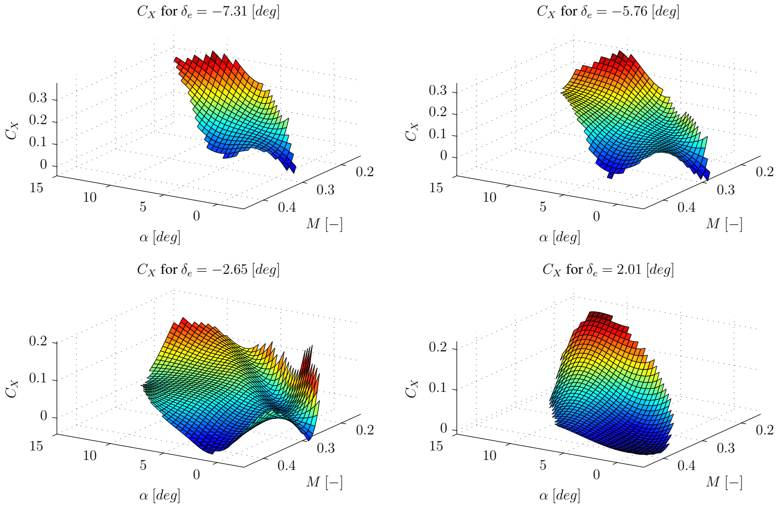


Figure 5.68: Four slices through the spline model $f(\alpha, M, \delta_e) \in \mathcal{S}_4^2$ for C_X.

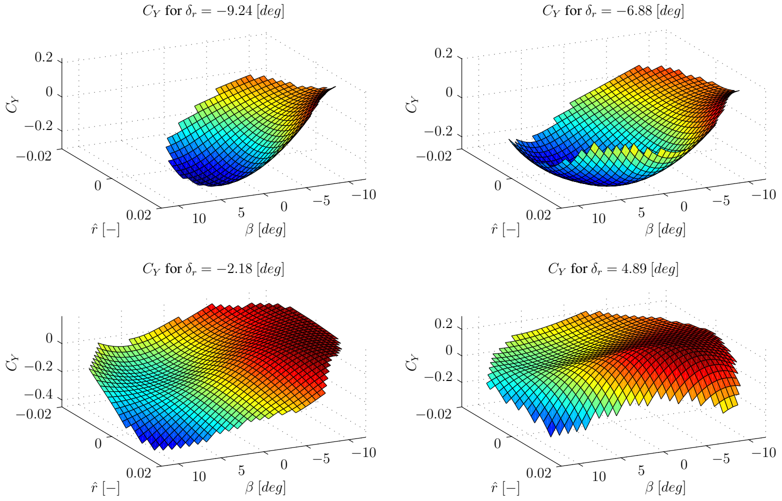
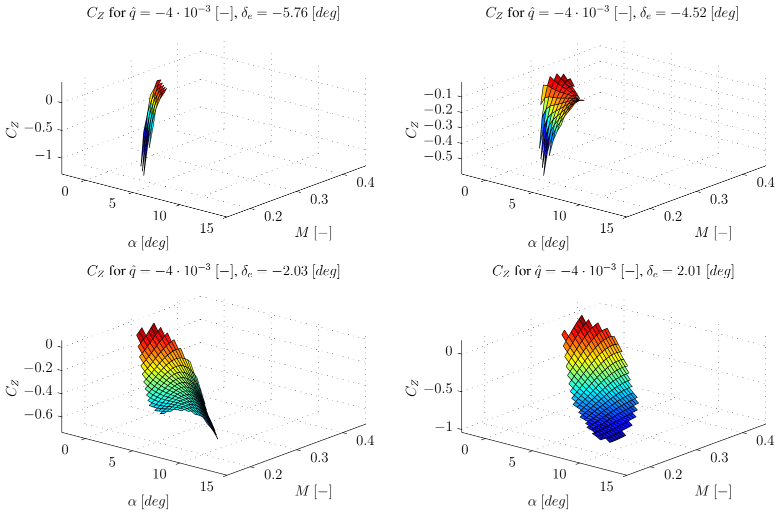


Figure 5.69: Four slices through the spline model $f(\beta, \delta_r, \hat{r}) \in \mathcal{S}_3^1$ for C_Y.



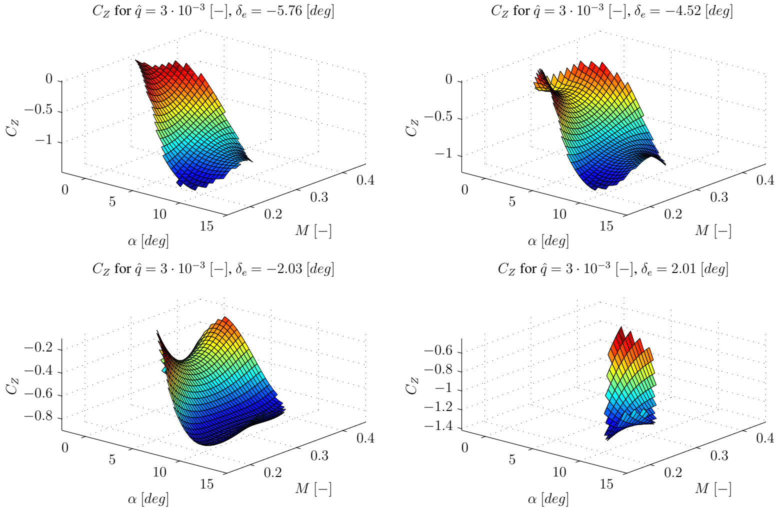
C_Z for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = -5.76$ $[deg]$

C_Z for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = -4.52$ $[deg]$

C_Z for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = -2.03$ $[deg]$

C_Z for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = 2.01$ $[deg]$

Figure 5.70: Four slices through the spline model $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_Z.



C_Z for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -5.76$ $[deg]$

C_Z for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -4.52$ $[deg]$

C_Z for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -2.03$ $[deg]$

C_Z for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = 2.01$ $[deg]$

Figure 5.71: Four slices through the spline model $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_Z.

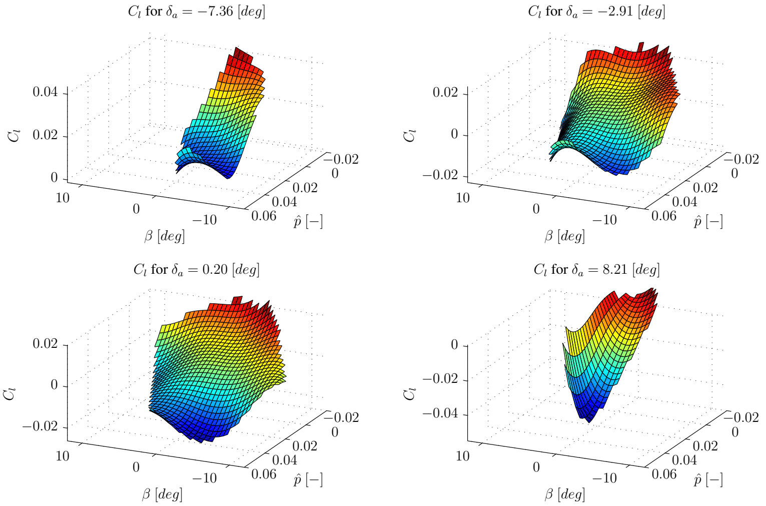


Figure 5.72: Four slices through the spline model $f(\beta, \delta_a, \hat{p}) \in \mathcal{S}_4^2$ for C_l.

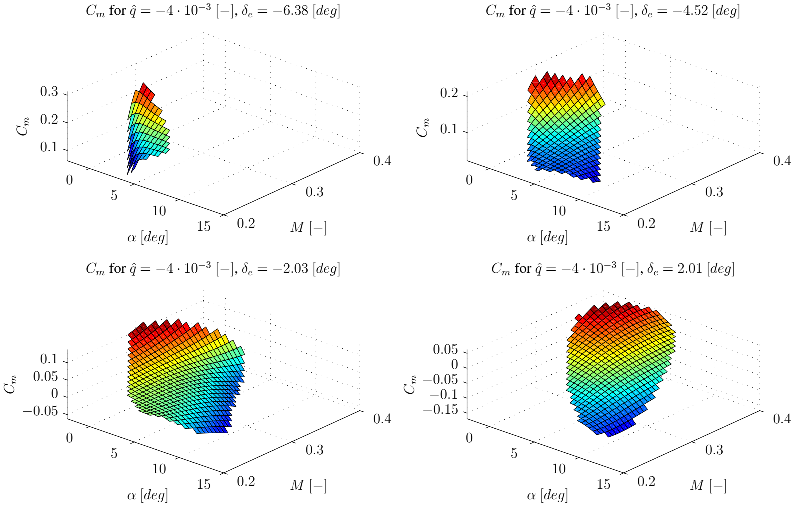
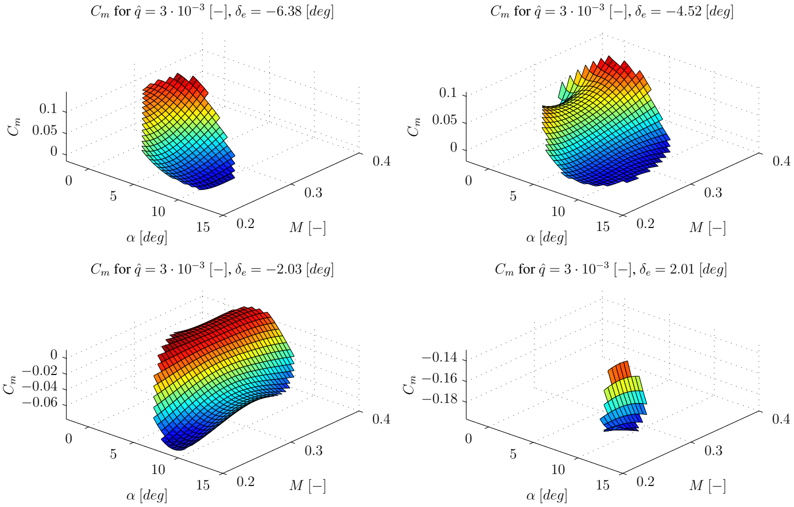


Figure 5.73: Four slices through the spline model $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_m.



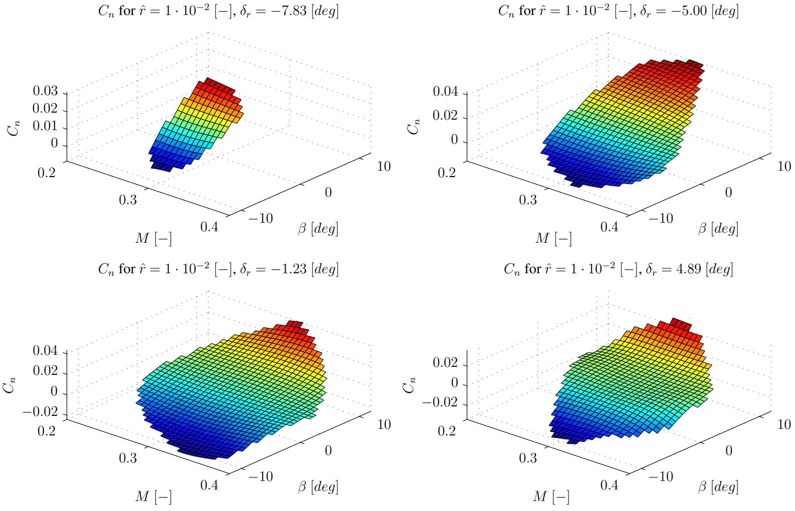
C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -6.38$ $[deg]$

C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -4.52$ $[deg]$

C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -2.03$ $[deg]$

C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = 2.01$ $[deg]$

Figure 5.74: Four slices through the spline model $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_m.



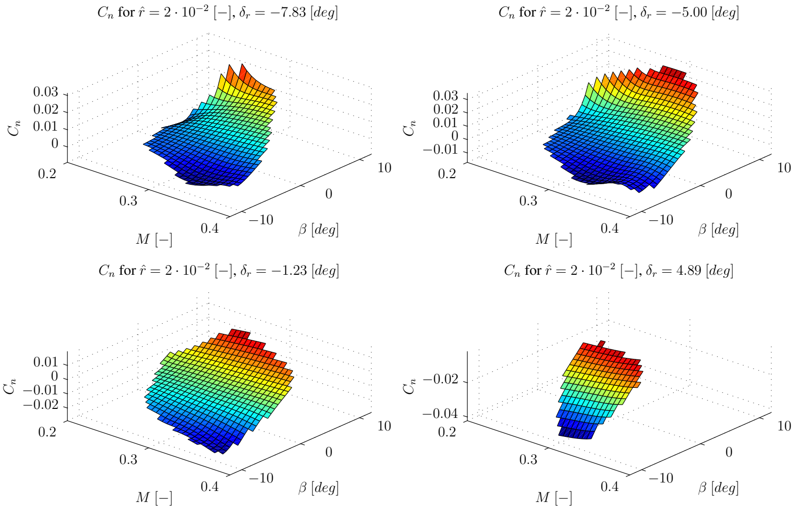
C_n for $\hat{r} = 1 \cdot 10^{-2}$ $[-]$, $\delta_r = -7.83$ $[deg]$

C_n for $\hat{r} = 1 \cdot 10^{-2}$ $[-]$, $\delta_r = -5.00$ $[deg]$

C_n for $\hat{r} = 1 \cdot 10^{-2}$ $[-]$, $\delta_r = -1.23$ $[deg]$

C_n for $\hat{r} = 1 \cdot 10^{-2}$ $[-]$, $\delta_r = 4.89$ $[deg]$

Figure 5.75: Four slices through the spline model $f(M, \beta, \delta_r, \hat{r}) \in \mathcal{S}_4^2$ for C_n.



C_n for $\hat{r} = 2 \cdot 10^{-2}$ $[-]$, $\delta_r = -7.83$ $[deg]$

C_n for $\hat{r} = 2 \cdot 10^{-2}$ $[-]$, $\delta_r = -5.00$ $[deg]$

C_n for $\hat{r} = 2 \cdot 10^{-2}$ $[-]$, $\delta_r = -1.23$ $[deg]$

C_n for $\hat{r} = 2 \cdot 10^{-2}$ $[-]$, $\delta_r = 4.89$ $[deg]$

Figure 5.76: Four slices through the spline model $f(M, \beta, \delta_r, \hat{r}) \in \mathcal{S}_4^2$ for C_n.

that the confidence models are accurate only inside the data envelope E_X from Eq. 5.18 as, by definition, no identification residuals are available outside E_X. In the following, the confidence models for the six aerodynamic models are presented and discussed. In the following figures, the distance between the confidence model and the spline model gives an indication of the relative width of the confidence bounds.

In Figure 5.77 a total of four slices through the spline model for C_X are shown. Projected in scale on top of the spline model is the 97% confidence model for this particular spline function. Figure 5.77 proves that the spline model for C_X is of less than adequate quality, as the confidence intervals are wide relative to the spline model values.

In Figure 5.78 four slices through the confidence model of the spline model for C_Y are shown. In this case, the distance between the confidence model and the spline model is relatively small, except for large rudder deflections. This means that the confidence bounds have a relatively small width, indicating that the model for C_Y is of adequate quality.

Eight slices through the confidence model and the spline model for C_Z are shown in Figure 5.79 and Figure 5.80. The confidence bounds are of average relative width, indicating that the model for C_Z is of adequate overall quality.

Four slices through the confidence model and the spline model for C_l are shown in Figure 5.81. The confidence model for C_l shows a clear bulge for high positive values of $\hat{p}$. In this region, the spline models also shows a significant nonlinearity. This nonlinearity, however, is located completely inside the confidence bounds, which in turn means that it is more likely to be a modeling error than an actual nonlinearity in C_l.

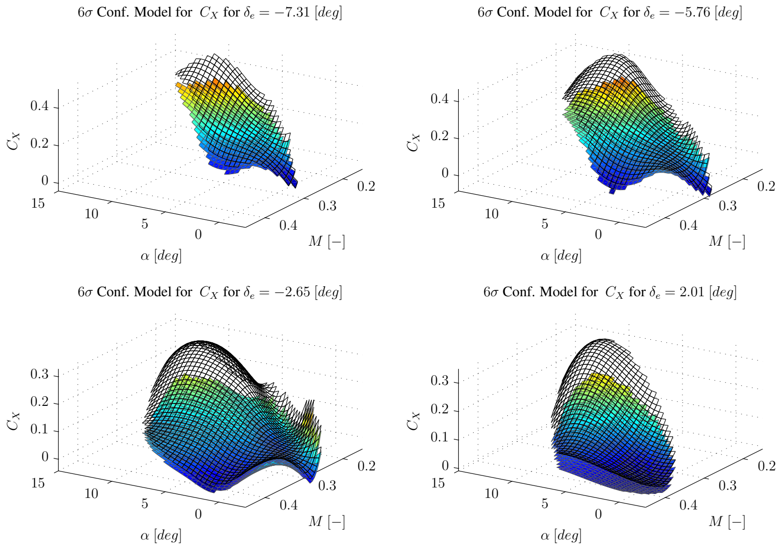


Figure 5.77: Four slices through the 97% confidence model for C_X.

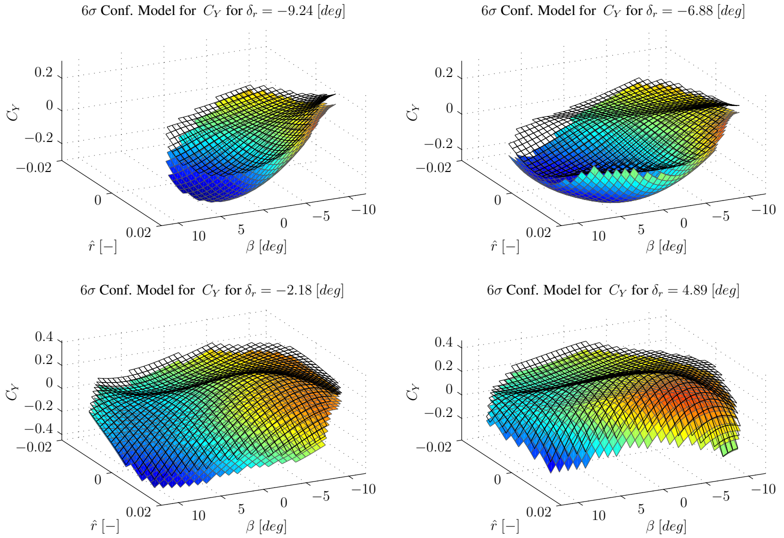
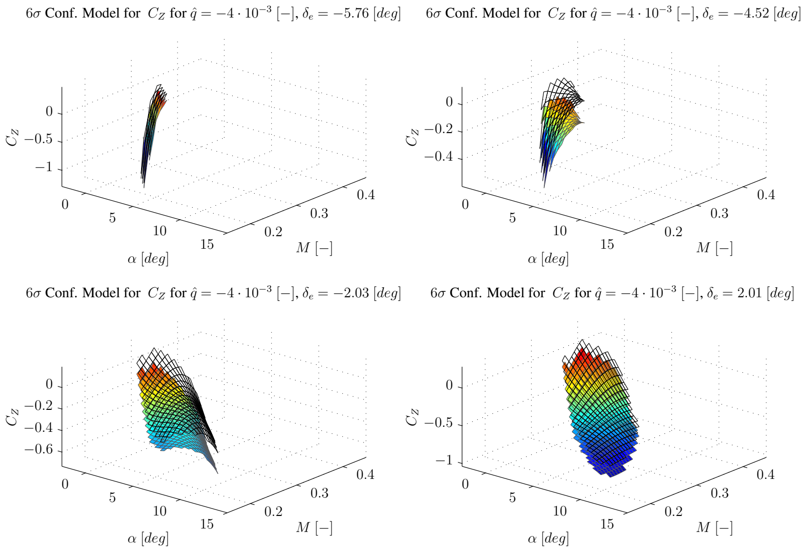


Figure 5.78: Four slices through the 97% confidence model for C_Y.



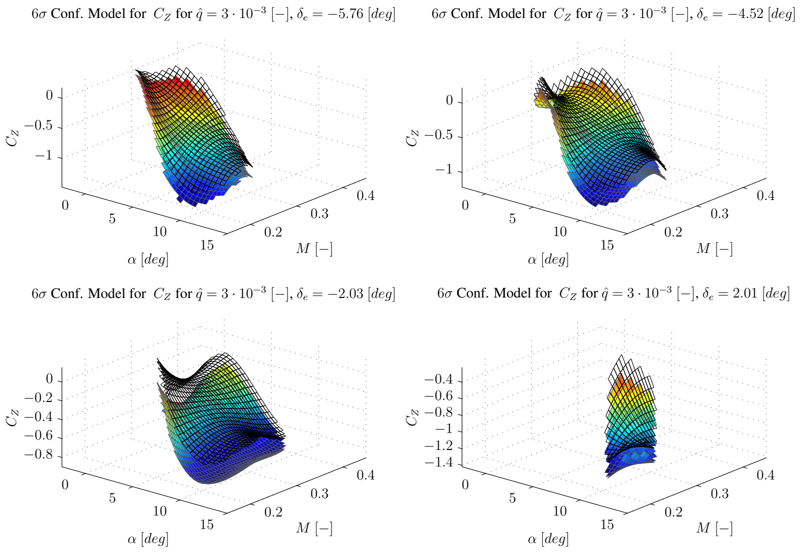
6σ Conf. Model for C_Z for $\hat{q} = -4 \cdot 10^{-3}\ [-]$, $\delta_e = -5.76\ [deg]$

6σ Conf. Model for C_Z for $\hat{q} = -4 \cdot 10^{-3}\ [-]$, $\delta_e = -4.52\ [deg]$

6σ Conf. Model for C_Z for $\hat{q} = -4 \cdot 10^{-3}\ [-]$, $\delta_e = -2.03\ [deg]$

6σ Conf. Model for C_Z for $\hat{q} = -4 \cdot 10^{-3}\ [-]$, $\delta_e = 2.01\ [deg]$

Figure 5.79: Four slices through the 97% confidence model for C_Z.



6σ Conf. Model for C_Z for $\hat{q} = 3 \cdot 10^{-3}\ [-]$, $\delta_e = -5.76\ [deg]$

6σ Conf. Model for C_Z for $\hat{q} = 3 \cdot 10^{-3}\ [-]$, $\delta_e = -4.52\ [deg]$

6σ Conf. Model for C_Z for $\hat{q} = 3 \cdot 10^{-3}\ [-]$, $\delta_e = -2.03\ [deg]$

6σ Conf. Model for C_Z for $\hat{q} = 3 \cdot 10^{-3}\ [-]$, $\delta_e = 2.01\ [deg]$

Figure 5.80: Four slices through the 97% confidence model for C_Z.

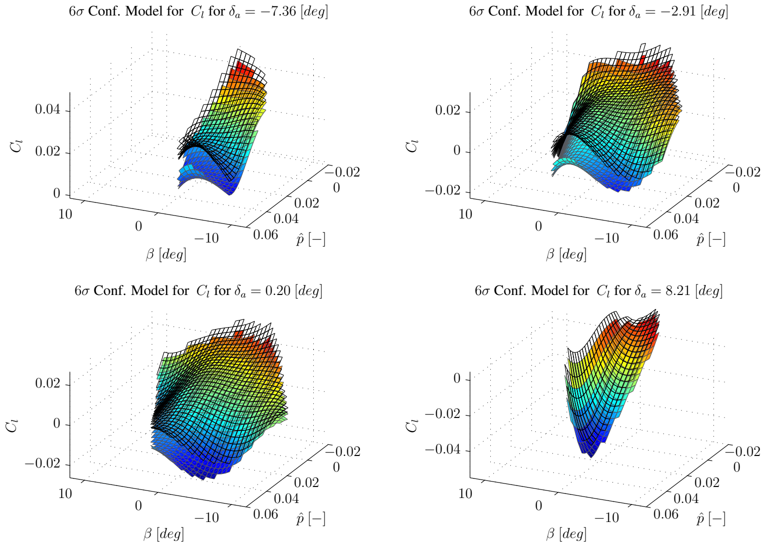


Figure 5.81: Four slices through the 97% confidence model for C_l.

In Figure 5.82 and Figure 5.83 a total of eight slices through the confidence model and the spline model for C_m are shown. The confidence bounds for negative values of $\hat{q}$ are relatively tight as can be seen in Figure 5.82. For positive values of $\hat{q}$ the confidence bounds widen considerably, especially for more positive elevator deflection angles, as can be seen in Figure 5.83. The explanation for this is that for a stable aircraft, positive pitch rates are directly correlated with negative elevator deflections. A positive pitch rates with a negative elevator deflection is a relatively rare occurrence. The result of this is that the amount of data in these regions of the flight envelope is limited, leading to a lower per-simplex data coverage and with that, lower estimator performance. In other words, the wider confidence bounds in Figure 5.83 are the result of the data coverage problem .

Eight slices through the confidence model and the spline model for C_n are shown in Figure 5.84 and Figure 5.85. The distance of this confidence model to the spline model is very small, meaning that the confidence interval is tight. This is another indication that the spline model for C_n is of a high quality.

5.6.4 Stability analysis

A unique property of a simplex spline based model is that its well-behavedness and stability are completely determined by the maximum absolute values of its B-coefficients. In Chpt. 3 an expression for bounds on the values of B-form polynomials was presented. These bounds

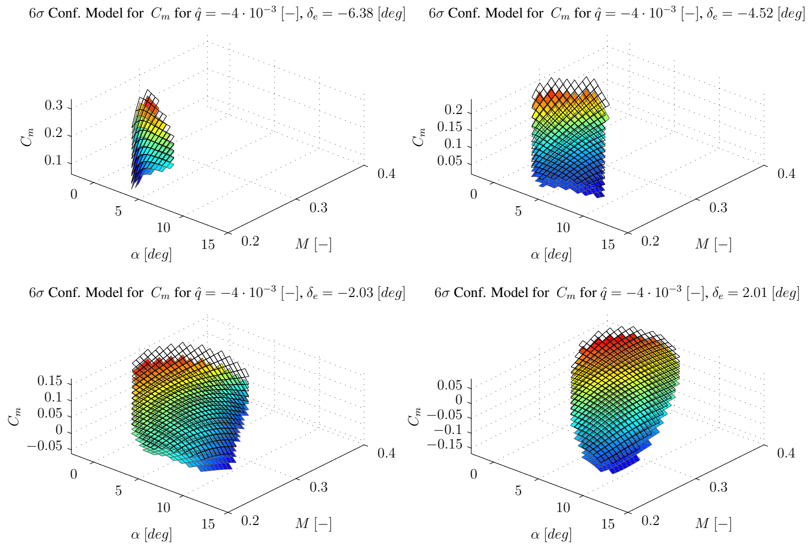


Figure 5.82: Four slices through the 97% confidence model for C_m.

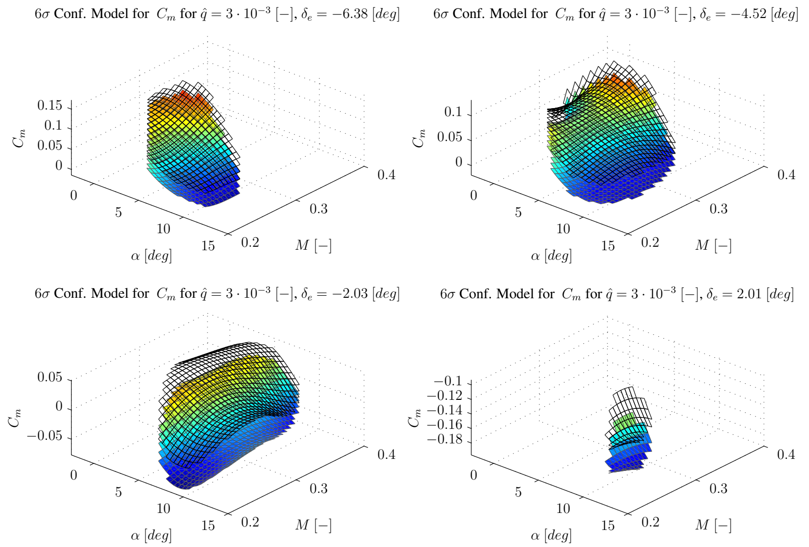


Figure 5.83: Four slices through the 97% confidence model for C_m.

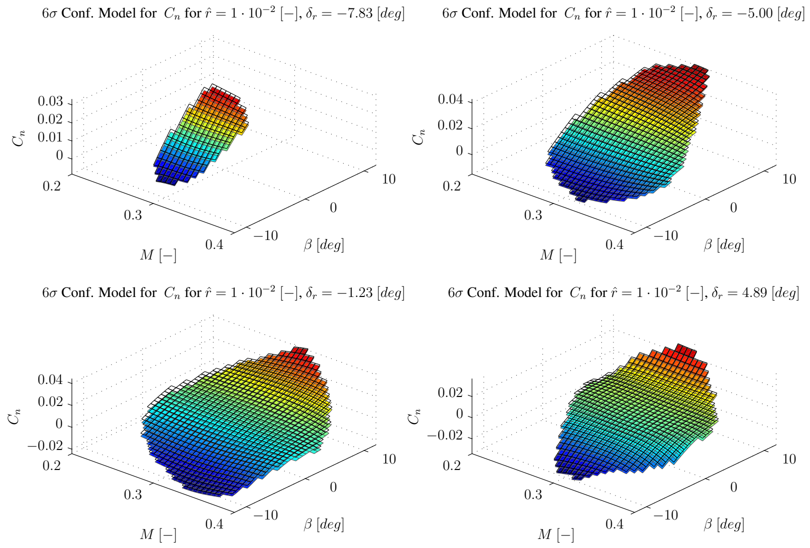


Figure 5.84: Four slices through the 97% confidence model for C_n.

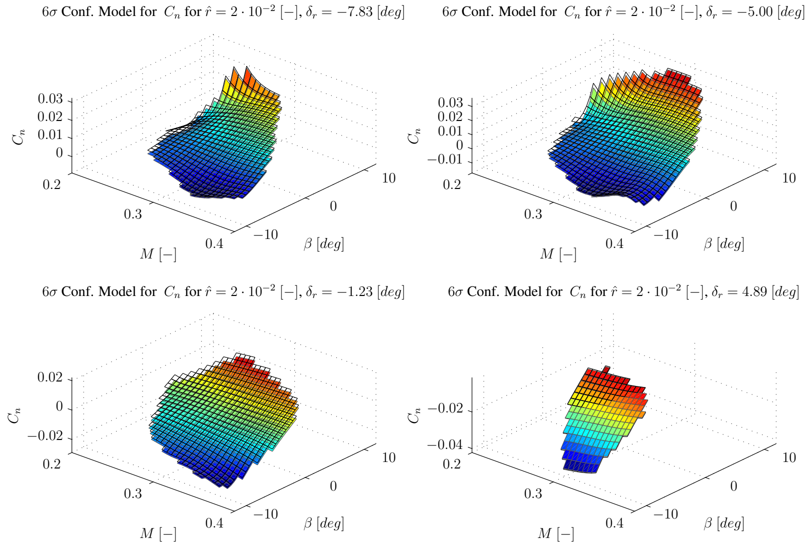


Figure 5.85: Four slices through the 97% confidence model for C_n.

were used in Sec. 4.3 to derive an expression for the stability of a simplex spline based model. This expression, which was derived in Eq. 4.87, was found to be:

$$\|s\| \leq \|\mathbf{c}\|, \tag{5.76}$$

with s the complete simplex spline function, and with $\mathbf{c}$ the global vector of estimated B-coefficients. This expression states that the maximum value in the vector of B-coefficients bounds the maximum value of the spline polynomial within the spline domain.

In the following the values of the B-coefficients of all spline models will be shown. A distinction will be made between B-coefficients inside the validity region E_X of the spline function as defined in Eq. 5.18. This distinction is warranted because the influence of B-coefficients on the global spline function is local, as was explained in Chpt. 2. This means that any B-form polynomial inside E_X is loosely bounded by the B-coefficients in E_X, while having hard bounds given by the complete set of B-coefficients. A measure for the well-behavedness of the B-coefficients is the ratio between the maximum absolute B-coefficient value and the range of E_X:

$$R_{E_X} = \frac{\text{range}\ \{\mathbf{c} \in E_X\}}{\text{range}\ E_X}, \tag{5.77}$$

The values of the B-coefficients of the spline model for C_X are shown in Figure 5.86. In the left hand plot of Figure 5.86, the complete set of B-coefficients of $f(\alpha, M, \delta_e) \in \mathcal{S}_4^2$ are shown. In this case, the white dots correspond with B-coefficients inside E_X. The B-coefficients inside E_X are also plotted in the right hand plot of Figure 5.86. The shaded area in the figures shows the bounds of the observed values of C_X inside the validity region. The figure shows that the values of the B-coefficients inside E_X are of relatively low magnitude. In particular, it is found that $R_{E_X} = 2.15$. This means that any B-form polynomial is loosely bounded inside E_X by these coefficient values. In this case, it is safe to conclude that the model for C_X is well-behaved and stable within E_X.

In Figure 5.87, the values of the B-coefficients of the spline model for C_Y are shown. This time, the values of the B-coefficients inside E_X are somewhat larger (i.e. $R_{E_X} = 5.77$) than the bounds of the observed values of C_X, which is represented by the shaded area in Figure 5.87. Inside E_X the spline model can be assumed to be relatively stable.

The B-coefficients values of the spline model for C_Z are shown in Figure 5.88. The B-coefficient values inside E_X are not well-behaved, as some have values that are more than one order of magnitude larger than the measured values for C_Z inside E_X ($R_{E_X} = 37.5$). It can be assumed that this model is subject to significant instabilities inside and outside of E_X.

The B-coefficients values of the spline model for C_l are plotted in Figure 5.89. The B-coefficients of the spline function $f(\beta, \delta_a, \hat{p}) \in \mathcal{S}_4^2$ for C_l are somewhat better behaved than the B-coefficients of the spline function for C_Z (i.e. $R_{E_X} = 10.7$). However, care should

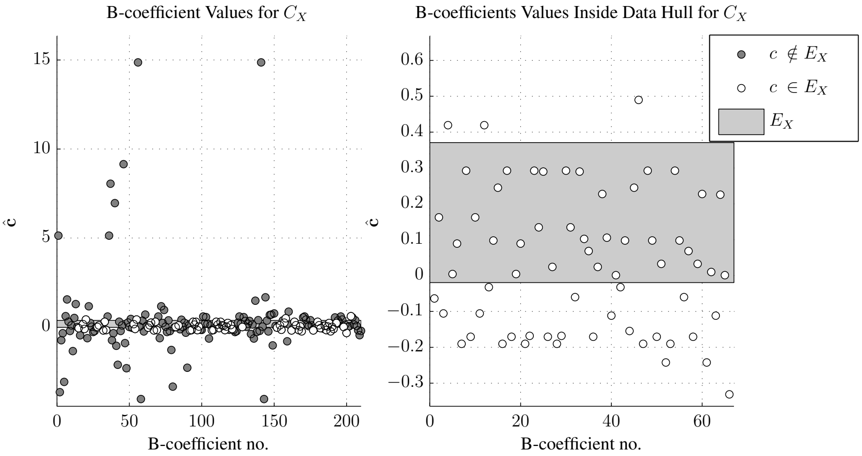


Figure 5.86: B-coefficient values of the spline model $f(\alpha, M, \delta_e) \in \mathcal{S}_4^2$ for C_X.

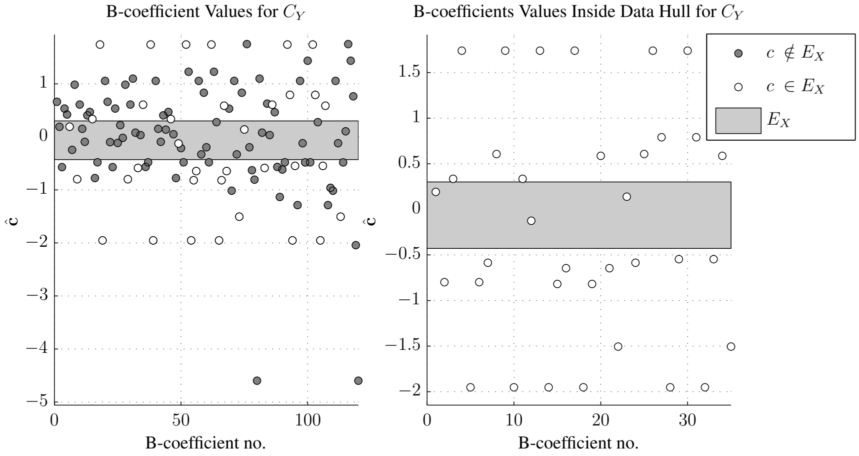


Figure 5.87: B-coefficient values of the spline model $f(\beta, \delta_r, \hat{r}) \in \mathcal{S}_3^1$ for C_Y.

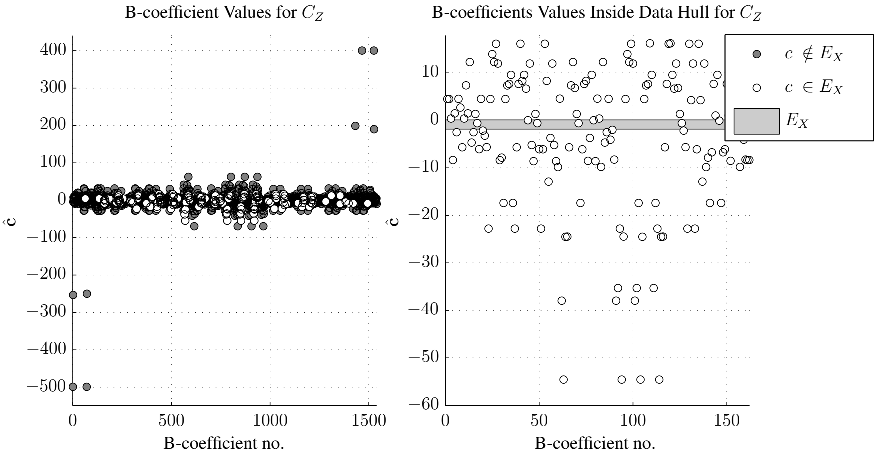


Figure 5.88: B-coefficient values of the spline model $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_Z.

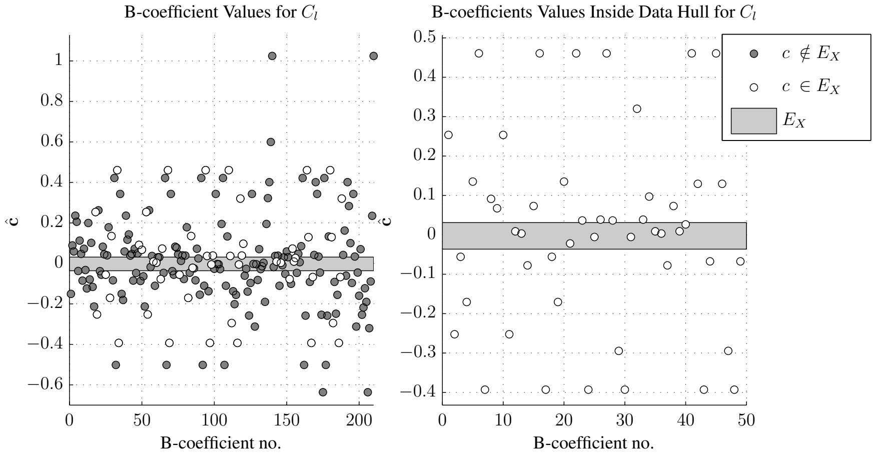


Figure 5.89: B-coefficient values of the spline model $f(\beta, \delta_a, \hat{p}) \in \mathcal{S}_4^2$ for C_l.

be taken with this spline function as stability cannot be guaranteed, especially outside of E_X.

In Figure 5.90 the B-coefficient values of the spline model for C_m are shown. This time, the B-coefficients inside E_X are relatively well-behaved ($R_{E_X} = 8.5$). These results show that caution should be taken with this particular spline model because there is a possibility that it is not well behaved inside E_X. Additionally, there are a number of outlier B-coefficients Outside E_X, which point to unstable model regions.

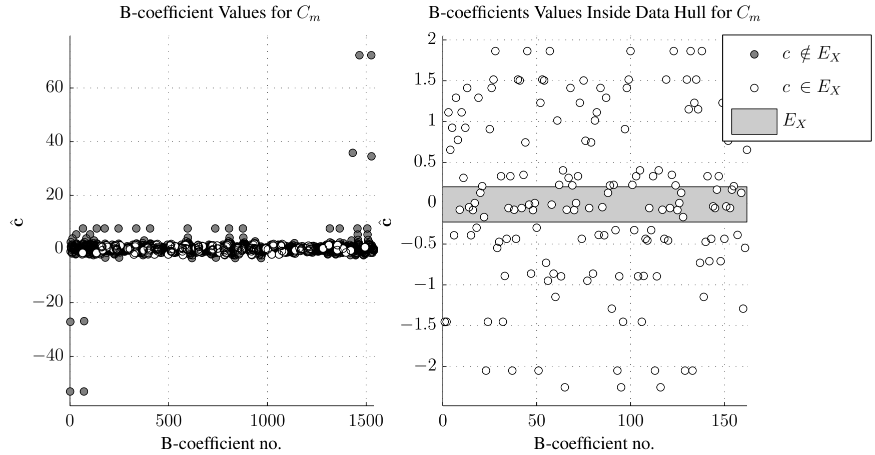


Figure 5.90: B-coefficient values of the spline model $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_m.

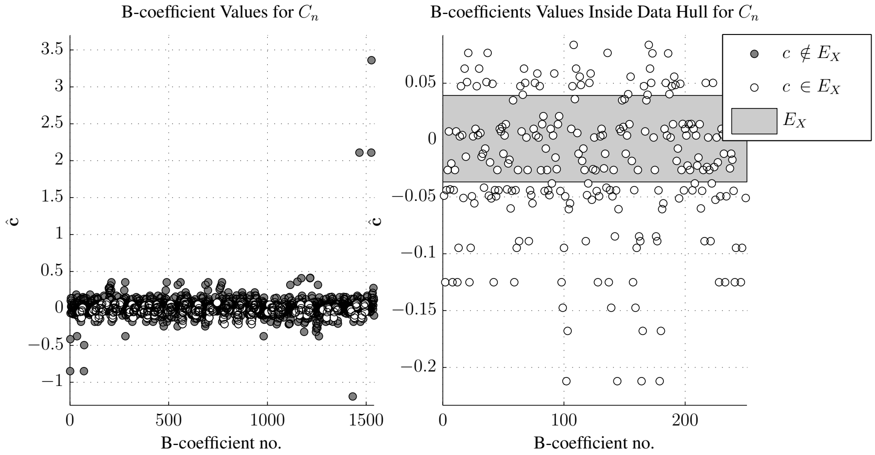


Figure 5.91: B-coefficient values of the spline model $f(M, \beta, \delta_r, \hat{r}) \in \mathcal{S}_4^2$ for C_n.

Finally, Figure 5.91 shows the values of the B-coefficients of the spline model for C_n. In this case all B-coefficients inside E_X are relatively well-behaved ($R_{E_X} = 3.75$), but there are some outliers outside E_X. It can therefore be concluded that the model is likely to be relatively stable inside E_X.

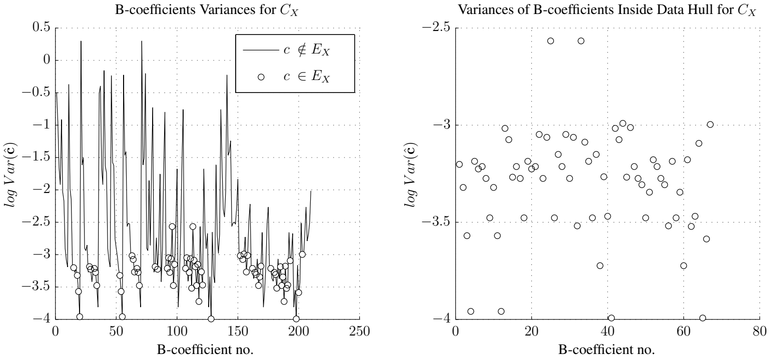


Figure 5.92: B-coefficient variances for the spline function $f(\alpha, M, \delta_e) \in \mathcal{S}_4^2$ for C_X.

5.6.5 Statistical analysis

The estimator for the B-coefficients of the multivariate simplex spline enables the statistical analysis of parameter estimation results. The exact method for estimating parameter covariances as well as the Cramér-Rao lower bounds was presented in Chpt. 4. In this paragraph, the results from a rigorous statistical analysis of the spline models are presented. It should be noted that all variances presented here are estimated using an ordinary least squares estimator, which implies that the residuals are white noise which, as was shown in Sec. 5.6.2, is clearly not the case. However, the results from the analysis presented here do provide lower bounds for the variances. Two different perspectives on the statistical analysis are provided. The first of these is a straightforward linear plot of the B-coefficient variances. The second perspective is the variance hyper surface, which were first introduced in [40]. The variance hyper surface is a linear interpolation of the B-coefficient variances, which is possible because the B-coefficients of the multivariate simplex splines have a spatial location within the spline domain. All 2-dimensional slices through the variance hyper surfaces correspond with the slices of their respective spline models shown in Sec. 5.6.2.

In Figure 5.92 a linear plot of the B-coefficient variances of the spline model for C_X is shown. The variances of the B-coefficients inside E_X are of limited magnitude and show little spread, indicating adequate estimator performance. Outside E_X, however, B-coefficient variances reach very significant values. In Figure 5.93 four slices through the 3-dimensional B-coefficient variance surface are shown. These slices correspond with the slices shown in the global model inspection from Sec. 5.6.2. In this case, the variance hypersurfaces clearly show that estimator performance drops dramatically outside E_X, i.e. the corners of the plots in the figure.

As a demonstration of the unique statistical validation methods enabled by the simplex

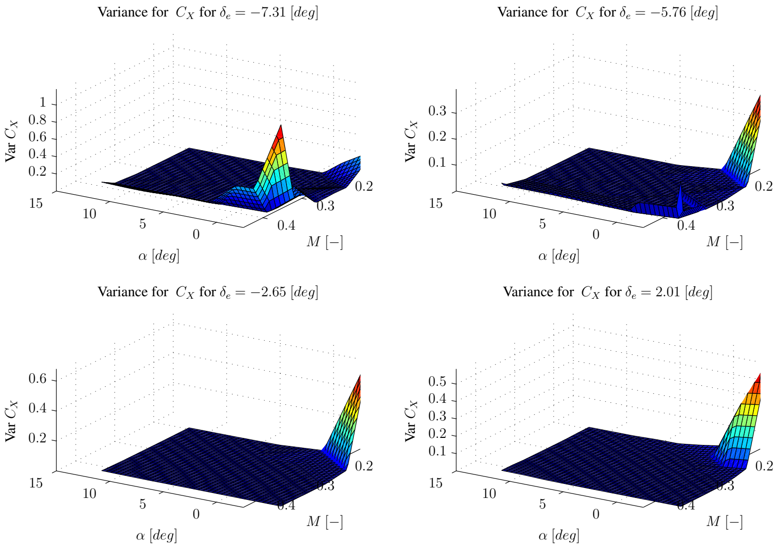


Figure 5.93: B-coefficient variance surfaces of the spline function $f(\alpha, M, \delta_e) \in \mathcal{S}_4^2$ for C_X.

splines, the B-net of the spline function for C_X is shown in Figure 5.94. In this case, the size and color of the B-coefficients are proportional with their variances, that is, the larger and more red (and less blue) a B-coefficient is, the higher the variance. Figure 5.94 clearly shows that B-coefficients located on external edges of the triangulation tend to have larger variances.

The estimated B-coefficient variances of the spline model for C_Y are shown in Figure 5.96. The magnitudes of the B-coefficient variances are relatively high, indicating inadequate estimator performance. However, the B-coefficient variance surfaces in Figure 5.96 peak at locations outside of E_X, indicating that the estimator is performing adequately inside E_X.

In Figure 5.97 the estimated B-coefficient variances of the spline model for C_Z are shown. These variances have a wide spread, indicating inadequate estimator performance. In Figure 5.98 and Figure 5.99 the variance hyper surfaces of the spline model are shown. In these figures it can be seen that the variance surface has peaks at locations inside E_X. This means that the B-coefficient estimator is performing inadequately inside E_X.

In Figure 5.100 the B-coefficient variances for the spline model for C_l are shown. The variances form a tight group, and have a low overall magnitude over the complete spline domain, indicating that the estimator is performing adequately. The B-coefficient variance

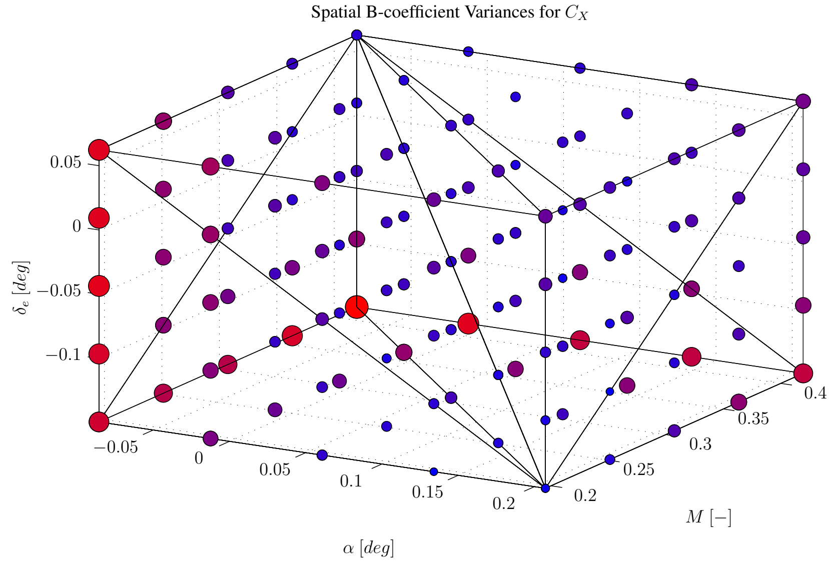


Figure 5.94: B-coefficient variances within the B-net of the spline model $f(\alpha, M, \delta_e) \in \mathcal{S}_4^2$ for C_X.

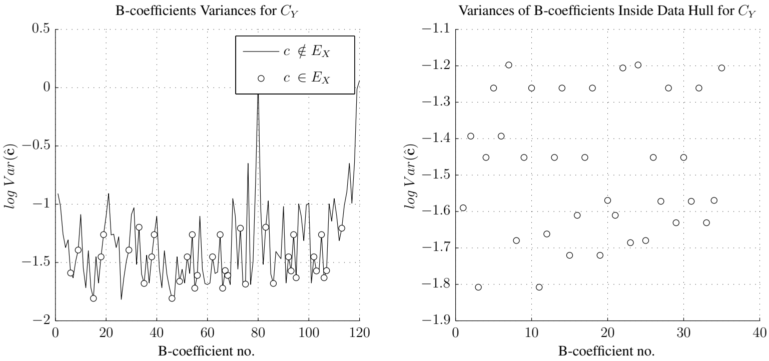


Figure 5.95: B-coefficient variances for the spline function $f(\beta, \delta_r, \hat{r}) \in \mathcal{S}_3^1$ for C_Y.

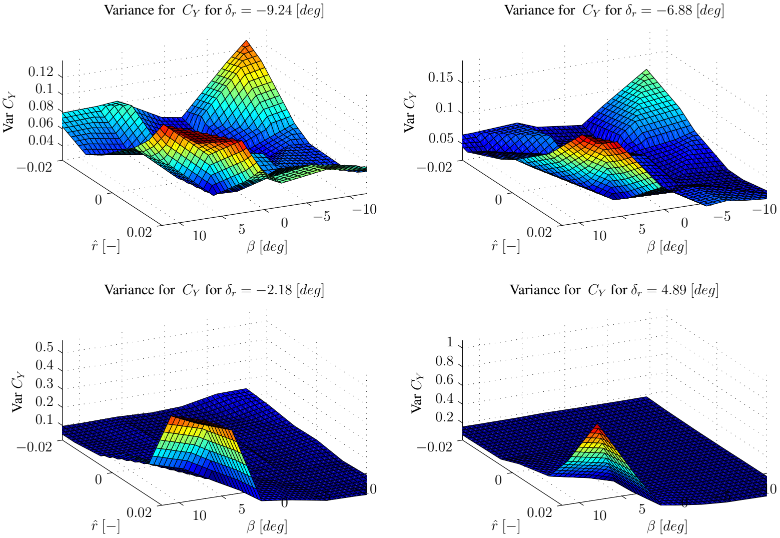


Figure 5.96: B-coefficient variance surfaces of the spline function $f(\beta, \delta_r, \hat{r}) \in \mathcal{S}_3^1$ for C_Y.

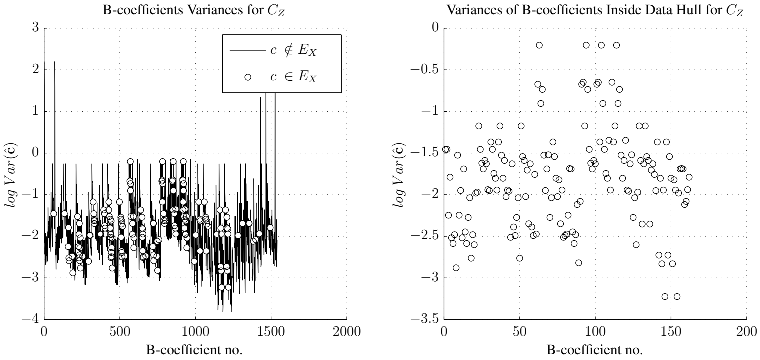
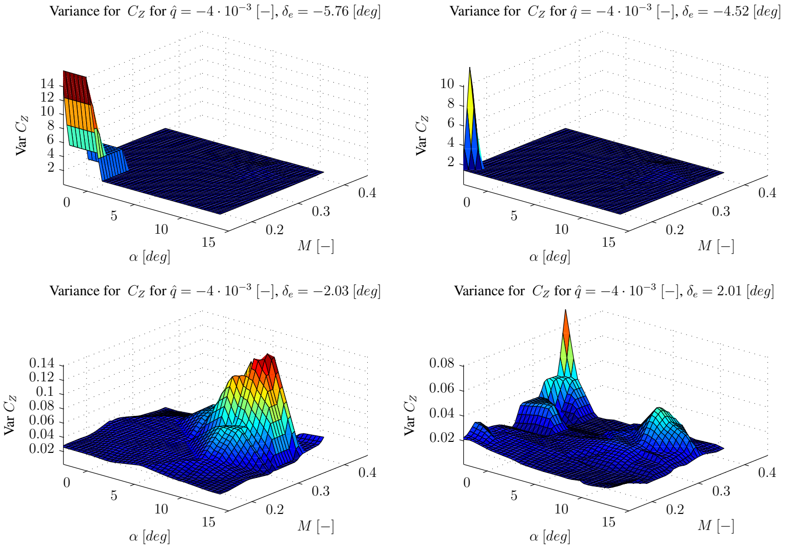


Figure 5.97: B-coefficient variances for the spline function $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_Z.



Variance for C_Z for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = -5.76$ $[deg]$

Variance for C_Z for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = -4.52$ $[deg]$

Variance for C_Z for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = -2.03$ $[deg]$

Variance for C_Z for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = 2.01$ $[deg]$

Figure 5.98: B-coefficient variance surfaces of the spline function $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_Z.

surfaces shown in Figure 5.101 reflect this fact, as the peaks in the variance surfaces are all outside E_X.

The B-coefficient variances of the model for C_m are shown in Figure 5.102. The B-coefficient variances again form a tight group indicating adequate estimator performance. The corresponding B-coefficient variance surfaces are shown in Figure 5.103 and Figure 5.104. These surfaces show that B-coefficient variances are highest close to the boundaries of the spline domain. Inside E_X there are a number of peaks in the variance surfaces, but they are of low magnitude. It can therefore be concluded that the estimator is performing adequately.

In Figure 5.105 the B-coefficient variances of the spline function for C_n are plotted. The variances are of very low magnitude and form a tight group. The B-coefficient variance hyper surfaces of this spline function are shown in Figure 5.106 and Figure 5.107. These surfaces show that B-coefficient variances are highest outside of E_X. There are some peaks inside E_X, but these are of very low magnitude. Therefore, it can be concluded that the B-coefficient estimator is performing adequately.

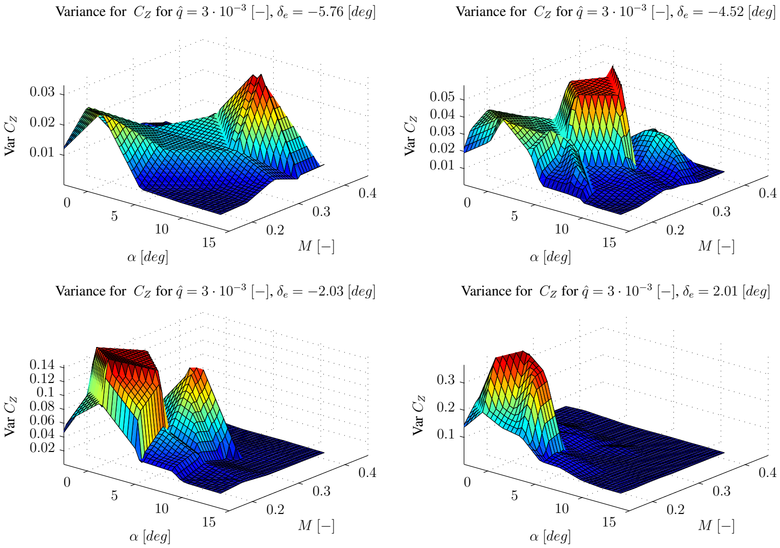


Figure 5.99: B-coefficient variance surfaces of the spline function $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_Z.

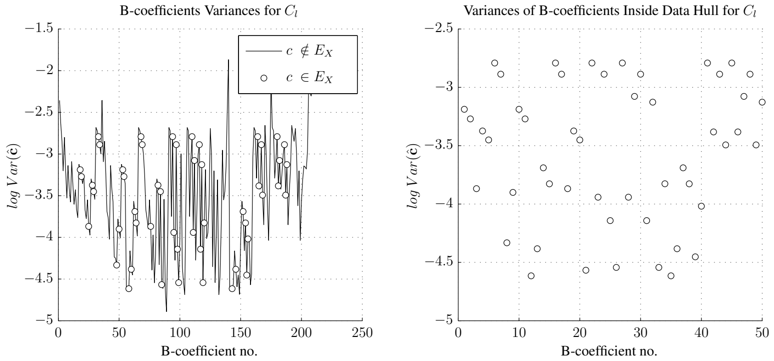


Figure 5.100: B-coefficient variances for the spline function $f(\beta, \delta_a, \hat{p}) \in \mathcal{S}_4^2$ for C_l.

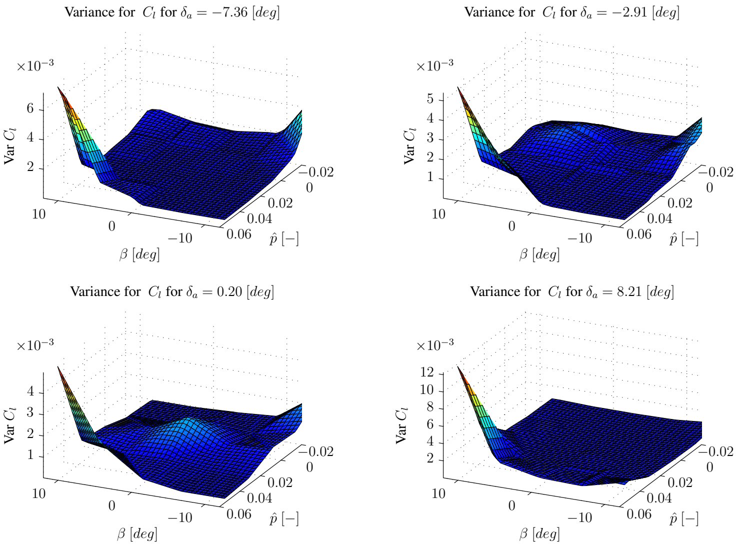


Figure 5.101: B-coefficient variance surfaces of the spline function $f(\beta, \delta_a, \hat{p}) \in \mathcal{S}_4^2$ for C_l.

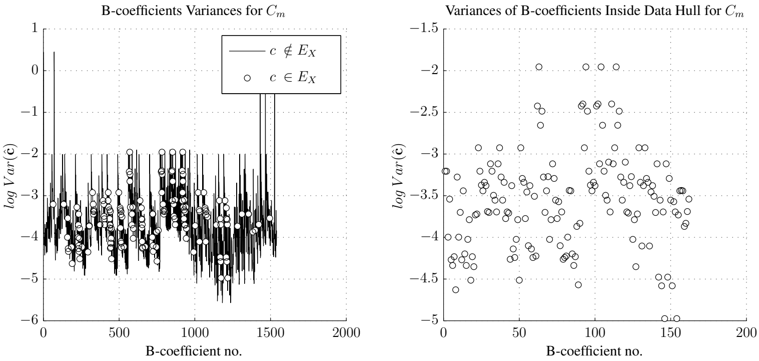
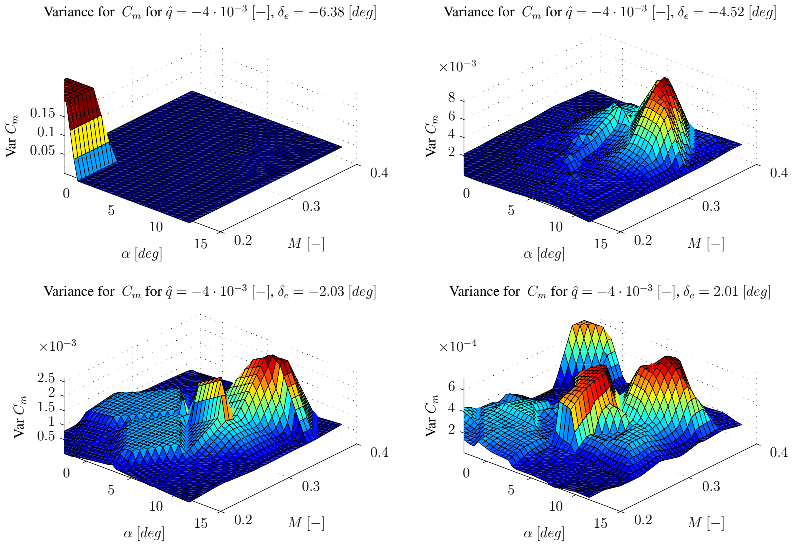


Figure 5.102: B-coefficient variances for the spline function $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_m.



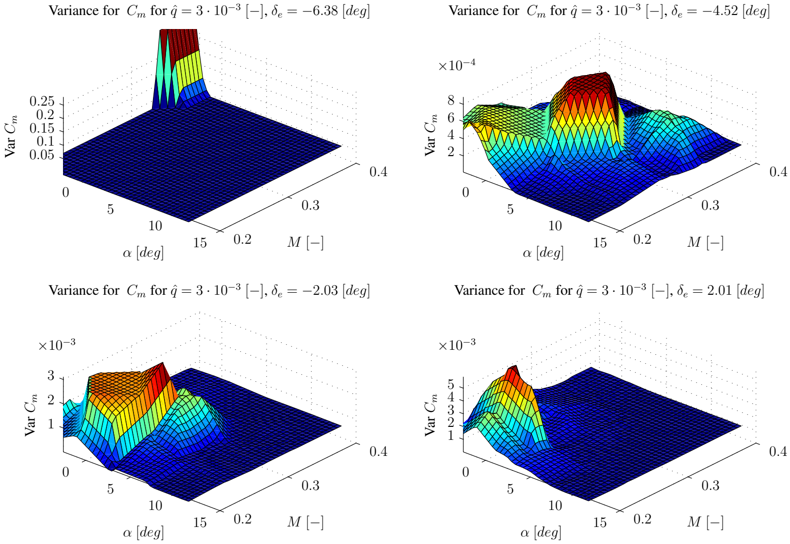
Variance for C_m for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = -6.38$ $[deg]$

Variance for C_m for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = -4.52$ $[deg]$

Variance for C_m for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = -2.03$ $[deg]$

Variance for C_m for $\hat{q} = -4 \cdot 10^{-3}$ $[-]$, $\delta_e = 2.01$ $[deg]$

Figure 5.103: B-coefficient variance surfaces of the spline function $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_m.



Variance for C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -6.38$ $[deg]$

Variance for C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -4.52$ $[deg]$

Variance for C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = -2.03$ $[deg]$

Variance for C_m for $\hat{q} = 3 \cdot 10^{-3}$ $[-]$, $\delta_e = 2.01$ $[deg]$

Figure 5.104: B-coefficient variance surfaces of the spline function $f(\alpha, M, \delta_e, \hat{q}) \in \mathcal{S}_4^2$ for C_m.

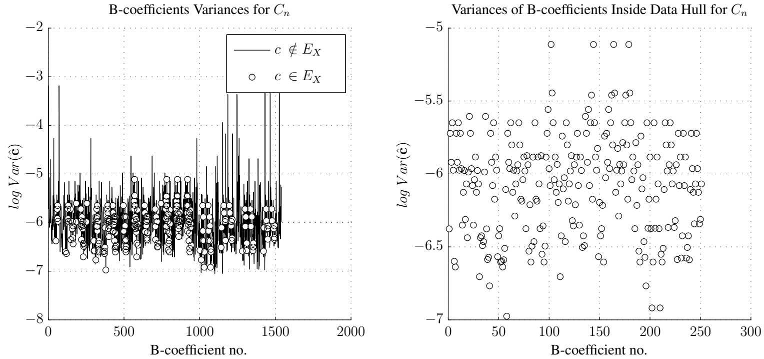


Figure 5.105: B-coefficient variances for the spline function $f(M, \beta, \delta_r, \hat{r}) \in \mathcal{S}_4^2$ for C_n.

Table 5.16: Final quantitative evaluation of the validation results

	C_X	C_Y	C_Z	C_l	C_m	C_n
Dimensions	(α, M, δ_e)	$(\beta, \delta_r, \hat{r})$	$(\alpha, M, \delta_e, \hat{q})$	$(\beta, \delta_a, \hat{p})$	$(\alpha, M, \delta_e, \hat{q})$	$(M, \beta, \delta_r, \hat{r})$
Triangulation	$\mathcal{T}_6$	$\mathcal{T}_6$	$\mathcal{T}_{22}$	$\mathcal{T}_6$	$\mathcal{T}_{22}$	$\mathcal{T}_{22}$
Spline Space	$\mathcal{S}_4^2$	$\mathcal{S}_3^1$	$\mathcal{S}_4^2$	$\mathcal{S}_4^2$	$\mathcal{S}_4^2$	$\mathcal{S}_4^2$
RMS$_{rel}$ ϵ	4.79%	7.18%	3.6%	1.73%	3.03%	2.55%
ϵ_{relmax}	30.61%	37.91%	20.90%	22.32%	26.68%	28.76%
R^2	87.07%	41.59%	90.78%	56.34%	64.44%	63.99%

5.6.6 Final model quality assessment

Using the combined results from the model validation methods presented in this section, a final evaluation of spline performance can be made. These results are summarized in Table 5.16, and will briefly be discussed here. Note that all numerical results are the results from the model validation, and not from model identification. The results shown in Table 5.16 were the most optimal results that could be achieved with the current dataset. A final qualitative evaluation of the spline models based on the various model validation results is presented in Table 5.17.

In general, it can be concluded that the spline model for C_n was of excellent quality, while the spline models for C_m and C_l were of adequate quality. The Spline models for C_X and C_Z were of mediocre quality and the spline model for C_Y was of inadequate quality. The aerodynamic coefficient that proved to be the most challenging to identify was the lateral force coefficient C_Y.

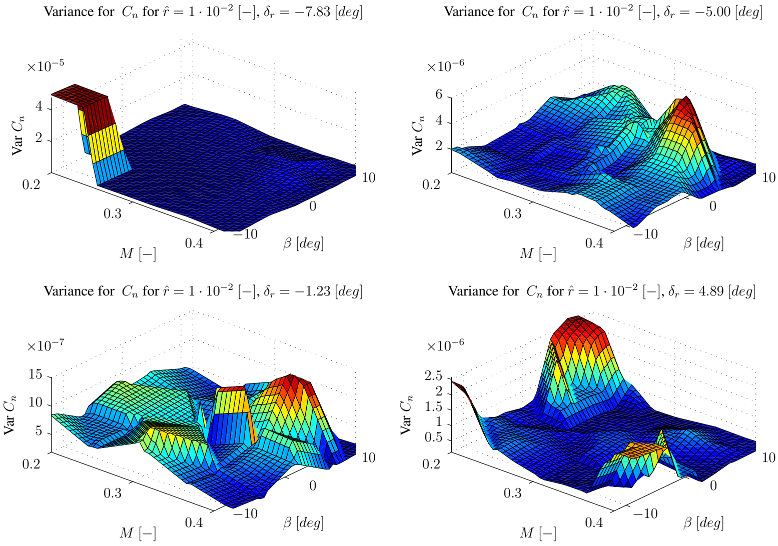


Figure 5.106: B-coefficient variance surfaces of the spline function $f(M, \beta, \delta_r, \hat{r}) \in \mathcal{S}_4^2$ for C_n.

Table 5.17: Final qualitative evaluation of the validation results.

	C_X	C_Y	C_Z	C_l	C_m	C_n
Raw validation results	$-$	$+$	$+/-$	$-$	$+$	$++$
Global model inspection	$+$	$+/-$	$+$	$+$	$+$	$+$
Residual analysis	$-$	$+/-$	$+$	$-$	$+$	$++$
Stability analysis	$++$	$+$	$--$	$-$	$+/-$	$+$
Statistical analysis	$+/-$	$-$	$-$	$+$	$++$	$++$
($++$:very good, $+$:good, $+/-$: mediocre, $-$: bad, $--$:very bad)						

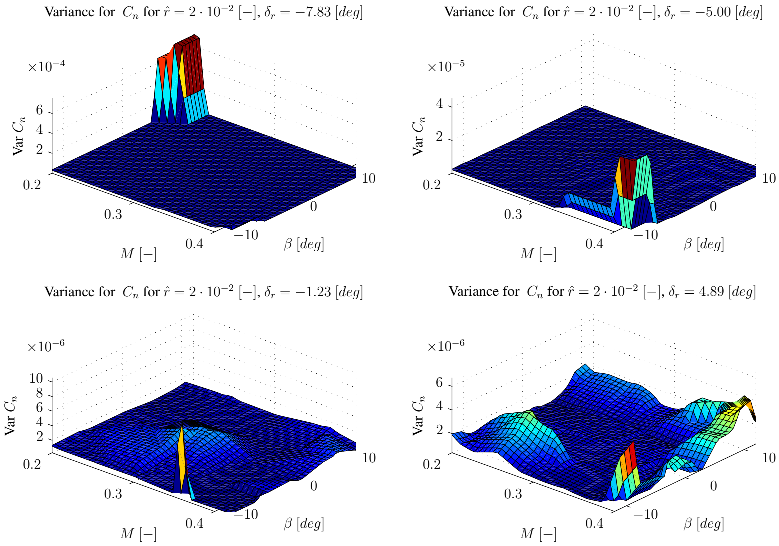


Figure 5.107: B-coefficient variance surfaces of the spline function $f(M, \beta, \delta_r, \hat{r}) \in \mathcal{S}_4^2$ for C_n.

Chapter 6

Conclusions

The aim of this thesis is the development of a new methodology for identifying global models of nonlinear systems using multivariate simplex splines. This aim is reflected by the main research statement of this thesis:
"*A methodology for global nonlinear model identification based on multivariate simplex splines can outperform current global model identification methods on aspects of model quality and model quality assessment.*"

The general conclusion of the work in this thesis results in the following response to the main research statement:

A new methodology for global model identification based on multivariate simplex splines was successfully developed. The new methodology was successfully applied to the aerodynamic model identification of the Cessna Citation II laboratory aircraft. It was demonstrated with a real-life aerodynamic model identification experiment that the new methodology not only produces global models of significantly higher quality than existing methods based on ordinary polynomials, but additionally provides a number of unique measures of model quality that allow for a much deeper and complete assessment of model quality.

The positive response to the research statement required the completion of a three-faceted research project in the fields of mathematical spline theory, system identification, and flight dynamics. The first facet of this research project was the development of new theories in the field of multivariate simplex splines in order to allow their use in a framework for system identification. The second facet of this research project was the development of a new methodology for system identification based on simplex splines. The third research facet was focused on using the new methodology in a real world application which was the identification of a complete set of aerodynamic models for the Cessna Citation II laboratory aircraft using flight test data.

It is the purpose of the remainder of this section to provide conclusions on the three main research facets. First, a conclusion on the mathematical theory of the multivariate simplex splines is presented in Sec. 6.1. Subsequently, in Sec. 6.2 conclusions of the research done in the field of system identification are presented. Finally, in Sec. 6.3 the research and experiments done in the field of aerodynamic model identification will be concluded.

6.1 Simplex spline theory

The mathematical foundations of the multivariate simplex splines were laid down quite recently, and many areas of research within the field remain unexplored. A further exploration of these areas could significantly increase the utility of the simplex splines in science and engineering. In this thesis the following contributions to multivariate simplex spline theory were made:

- A new matrix formulation for the multivariate simplex splines was developed (Sec. 3.2.1). This matrix formulation proved to be instrumental in the definition of the linear regression model for simplex splines.
- A new geometric triangulation optimization method was developed which resulted in triangulations well suited for use with simplex splines (Sec. 2.3.7)
- A new general formulation for the continuity conditions for the multivariate simplex splines was developed (Sec. 3.3.2). This new formulation is valid for all B-nets ordered according to the *B-net orientation rule* presented in this thesis.
- A new numerical method for defining smoothness matrices of full rank was developed (Sec. 3.3.5).
- A new effect was observed that negates the local approximator property of the simplex splines. This effect, called B-net propagation, was found to be present specifically in the popular Type I triangulation, and theories were developed for determining its occurrence and quantifying its effects (Sec. 3.3.4).
- A new one-step matrix formulation of the de Casteljau algorithm was developed, enabling the definition of the differential constraints (Sec. 3.1.3).
- A new type of linear constraints were developed in the form of the *differential constraints* (Sec. 3.4.1). Differential constraints bound the directional derivatives

of the simplex splines and enable recursive model estimators and local model extrapolation.

- Theories were developed for determining polynomial divergence on the boundaries of the spline domain (Sec. 3.4.3). These theories aid the required number and order of differential constraints necessary for bounded model extrapolation.
- Theories for a new formulation of the Bernstein basis polynomials in terms of global coordinates were developed. This new formulation allows the physical interpretation of the B-form polynomials and the development of rigorous triangulation optimization methods (Sec. 3.5.2).

The defining purpose of these developments is the creation of a new methodology for global model identification based on multivariate simplex splines. In short, the matrix formulation for the multivariate simplex splines not only simplifies their use in software applications (Matlab), but more importantly, allows for the definition of a linear regression model based on B-form polynomials. The power of the simplex spline comes from the combination of many basis polynomials defined on optimal triangulations. At the time of this writing, however, no general method for triangulation optimization exists in the literature. Therefore, a geometric triangulation optimization method was developed which produced triangulations that are specifically suited for use with simplex splines.

The definition of the continuity conditions used in the literature is not general enough to allow the creation of smoothness matrices on triangulations of general form. Therefore, a new formulation of the continuity conditions was developed which only requires that the B-nets on simplices are oriented using the *B-net orientation rule*. It was found that the smoothness matrix for general triangulations is rank deficient in most cases, with the result that any linear regression scheme for simplex splines is singular, and unsolvable using standard parameter estimation techniques. A new method for creating full rank smoothness matrices was therefore developed. This new method uses an estimate of the condition number of the smoothness matrix to determine whether a candidate continuity condition causes a rank deficiency. If so, the continuity condition is rejected, and if not, the continuity condition is added to the smoothness matrix. It was found that the most widely used triangulation type, the symmetric Type I triangulation, is subject to a newly discovered effect in the form of B-net propagation. B-net propagation is the sometimes undamped propagation of disturbances through the B-net of a complete triangulation. B-net propagation, when left unchecked, effectively transforms the multivariate simplex spline from a local approximator into a global approximator. It was proved in this thesis that only specific combinations of spline degree and continuity order produce undamped propagation effects. By avoiding these combinations, the local approximator property of the simplex splines is retained.

A new, highly promising application of the multivariate simplex spline was found in the form of a recursive simplex spline model estimator. The challenge with such an estimator is that the simplex splines are defined on triangulations which in their present form are static geometric structures. For real-time applications it cannot be guaranteed that every simplex in the triangulation is adequately covered with data. The result of this is that local model extrapolation will occur at regions inside the spline domain were data is scarce. In order to analyze the effects of this local model extrapolation, mathematical theories were developed which enabled the quantification of extrapolation effects. It was found that these extrapolation effects depend on the directional derivatives of the dataset being modeled. Additionally, it was found that polynomial divergence beyond the bounds of the dataset domain was unavoidable. In order to solve these problems a new type of linear constraints was created in the form of the differential constraints. The differential constraints required the formulation of a new form of the de Casteljau algorithm that reduced its iterative nature into a single-step form. The resulting differential constraints effectively allow the formulation of the directional derivatives of B-form polynomials in terms of their original vector of B-coefficients. Because of this, the differential constraints can be included in the Karush-Kuhn-Tucker solution system for the multivariate simplex splines. Additionally, and perhaps even more interesting, is the potential application of the differential constraints in the solving of partial differential equations using simplex splines, with differential constraints acting as Cauchy or Neumann boundary conditions.

The Bernstein basis polynomials of the simplex splines are defined in terms of barycentric coordinates which are local coordinates. The simplex spline polynomials therefore do not have any meaning beyond the simplex on which they are defined. In some cases it may be desirable, however, to obtain a global interpretation of the simplex spline polynomials, for example if a physical interpretation of a spline solution is required. In order to facilitate the physical interpretation of multivariate simplex splines, a new formulation of the Bernstein basis polynomials in global coordinates was developed. This new formulation not only makes it possible to determine the global physical meaning of a simplex spline function, but also enables new methods for triangulation optimization. Such a triangulation optimization method could use a global optimization technique like interval analysis to determine in a single step the optimal triangulation and B-coefficient values for a given set of data.

6.2 System identification with simplex splines

The theoretical contributions to the mathematical theory of multivariate simplex splines had a common objective which was the development of a new methodology for system identification based on the simplex splines. While these theoretical contributions enabled the methodology, its actual implementation was yet to be defined. In order to create a functional and practical method for system identification with simplex splines, the following developments were made:

- A new linear regression model based on the Bernstein basis polynomials of the multivariate simplex splines (Sec. 4.1.3) was developed.
- Theories were developed for the required per-simplex data volume and constraint matrix ranks (Sec. 4.2.1).
- A new equality constrained generalized least squares estimator for the B-coefficients of the multivariate simplex splines (Sec. 4.2.1) was developed.
- A new differential equality constrained recursive least squares estimator for the B-coefficients of the multivariate simplex splines (Sec. 4.2.2) was developed.
- A new method was developed for creating empirical confidence bounds for simplex splines using the spline model residual and Chebyshev's inequality.
- A completely new model quality assessment method based on the spatial location of the B-coefficients was discovered. This new method results in variance hypersurfaces which allow the pinpointing of regions of high variance inside the spline model (Sec. 4.3.2).

The new linear regression model based on Bernstein basis polynomials was essential to all further efforts in implementing the methodology. This new regression model allowed the application of existing parameter estimation techniques for the estimation of the B-coefficients of the multivariate simplex splines. It was soon found that simplex data coverage was one of the most important limiting factors during B-coefficient estimation. Data coverage was found to be the spatial analogue of system excitation; a sufficient data coverage is equivalent to a sufficient excitation. Two theories were proved which allowed the quantification of the required data coverage given a specific spline space and triangulation.

Using the linear regression model structure, a generalized least squares estimator was defined, and methods were developed for estimating parameter and measurement covariance matrices. The next step was the development of a recursive least squares estimator for the B-coefficients of the simplex splines. The recursive estimator allows the reconfiguration of simplex spline models in real-time, which will prove useful in adaptive model-based control systems based on simplex splines.

The new methodology for system identification with simplex splines leads to several new approaches to assessing the quality of the estimated models. One of these new approaches is a model residual analysis, which uses Chebyshev's inequality to estimate localized confidence bounds for simplex spline models. Another new quality assessment method is the creation of variance hyper surfaces based on the estimated parameter variances and the insight that B-coefficients have a spatial location within the spline domain. Finally, the stability of simplex spline models can be assessed simply by comparing the values of the B-coefficients with the range of the identification data values.

Summarizing, the new methodology for system identification with simplex splines has a number of advantages over other identification methods in use today. Firstly, the simplex splines have an arbitrarily high approximation power on a global model scale. Secondly, simplex spline models are parametric models, which allows for efficient approximation of very large datasets. Thirdly, the simplex splines are linear in the parameters, which means that linear regression methods can be used for their estimation. Fourthly, the simplex splines have a local polynomial basis, which implies that only small subsets of parameters and basis polynomials need to be considered during estimation and evaluation, resulting in efficient (sparse) computational schemes. And finally, the quality of simplex spline based models can be assessed using a number of unique and very powerful model quality assessment methods. These advantages became apparent during the identification of simplex spline based aerodynamic models for the Cessna Citation II laboratory aircraft.

6.3 Aerodynamic model identification with simplex splines

The new methodology for system identification based on multivariate simplex splines was used to identify aerodynamic models for the Cessna Citation II laboratory aircraft using a set of flight test data obtained during 247 flight test maneuvers. While the theoretical part of the system identification framework was complete, the following practical developments were necessary for a successful application and evaluation of the new framework:

- Flight tests were performed with the Cessna Citation II laboratory aircraft (Sec. 5.2, Sec. 5.3).
- Ground tests showed that the Q-Flex 3100 linear accelerometers in the current (2011) IMU aboard the PH-LAB contain noise levels that are an order of magnitude higher than manufacturer specifications (Sec. 5.3).
- A fully automated framework for data pre-processing, maneuver detection, and flight path reconstruction was implemented in Matlab (Sec. 5.3, Sec. 5.4).
- A new method for spline dimension selection based on the analysis of hysteresis effects was developed (Sec. 5.5).
- A general Matlab toolbox for the creation of multivariate simplex splines was created (Sec. 5.5).
- A high performance C++ implementation of the multivariate simplex spline toolbox was created (Sec. 5.5).
- A fully automated framework for aerodynamic model identification with multivariate simplex splines was implemented in Matlab (Sec. 5.5).

- A Matlab implementation of the various model quality analysis methods was made (Sec. 5.6).
- A validated set of longitudinal and lateral aerodynamic models were made for the Cessna Citation II (Sec. 5.6).

The main application of the new methodology for system identification developed in this thesis was that of aerodynamic model identification. Aerodynamic model identification aims to create models for the aerodynamic force and moment coefficients of aircraft using flight data. For this reason, a large database with flight data was assembled. This dataset contained flight data from a total of 7 test flights executed between 2006 and 2010. It was found that the data obtained during test flights performed before 2008 was seriously compromised by sensor glitches. Using different data reconstruction techniques, a significant fraction of this data could be recovered. The results from flight testing showed that the optimal input sequences in particular resulted in excellent excitations of aircraft dynamics.

A comparison between engine-idle and engine-off ground tests showed that the noise levels produced by the Q-Flex 3100 linear accelerometers inside the current (2011) IMU were an order of magnitude higher than those specified by the manufacturer, but only when the engines were running. The exact source of the noise is unknown at the time of this writing, but a mechanically induced vibration of the entire IMU assembly seems the most likely cause.

Using a fully automated framework for data pre-processing, automatic maneuver detection and flight path reconstruction, a set of 247 maneuver datasets was created. This dataset was split into an identification dataset and a validation dataset. The identification dataset formed the input to the multivariate simplex spline based aerodynamic model identification toolbox. This toolbox was implemented in Matlab, with performance critical parts implemented in C++. The resulting application was efficient enough to allow the creation of thousands of candidate aerodynamic models per day on a workstation class PC[1].

Following the identification of the aerodynamic models, the validation dataset was used to validate the identified aerodynamic models. During this phase, the unique model quality assessment methods available to the simplex splines were applied. The model validation proved that the new methodology for system identification with simplex splines invariably produces higher quality results than methods based on ordinary polynomials.

The full potential of the simplex splines as a tool for system identification was compromised by the data coverage problem. The data coverage problem, which is the result of an insufficient per-simplex data coverage, requires a different approach towards data gathering. An important conclusion is therefore that the new methodology for system identification with simplex splines should be used in conjunction with a new tool that, during experiment time, provides an indication of per-simplex data coverage.

---

[1] In this case an Intel Core i7-920 with 12 GB of RAM

Chapter 7

Recommendations

In this chapter the recommendations resulting from the research performed in the framework of this thesis will be presented. This research consisted of three main facets in three different fields of science. The first facet was the further development of the mathematical theory of the multivariate simplex splines with the aim of enabling their use in a framework for system identification. The second facet was the development of a new methodology for system identification based on the simplex splines. The third facet was the application of the new methodology for the identification of aerodynamic models for the Cessna Citation II laboratory aircraft. This research resulted in a number of recommendations that should aid future research efforts in these three fields of science.

In Sec. 7.1 recommendations in the field of multivariate simplex splines are presented. Recommendations in the field of system identification are presented in Sec. 7.2. Finally, in Sec. 7.3 recommendations in the field of aerodynamic model identification and flight dynamics are presented.

7.1 Simplex spline theory

A number of recommendations result from the research performed in the field of multivariate simplex spline theory. These recommendations will aid any future research efforts in the field of simplex spline based global model identification. The most important recommendations are the following:

- A truly general triangulation optimization method should be developed which is specifically aimed at creating optimal triangulations for multivariate simplex splines and which is fully integrated with the new global model identification methodology.

- An analytical method for creating full-rank smoothness matrices should be developed.
- Distributed computing schemes for creating simplex splines that take advantage of current trends in parallel computing should be developed.
- Methods for formulating physical equality and inequality constraints based on the global formulation of the B-form should be added to the current theory.

The first, and most important recommendation is that a general triangulation optimization method should be developed that is specifically suited for global model identification with simplex splines. Current methods for triangulation optimization, including the method presented in this thesis, result in triangulations that are optimal in a geometric sense. Effective triangulation methods for simplex splines, however, should also take the data coverage of simplices into account. It was proved in this thesis that an inadequate coverage of simplices with data results in singular B-coefficient estimation problems.

The second recommendation is that an analytical method for the creation of full-rank smoothness matrices should be developed. The method used in this thesis is a numerical method, which is computationally expensive, especially for large smoothness matrices. An analytical method would prevent the formulation of redundant continuity conditions altogether, resulting in a much lower computational load. At the time of this writing, however, such an analytical method has proved to be quite elusive to develop.

The current trend in computing is parallelization, which is made possible by very powerful multi-core CPU's and GPU's. The third recommendation therefore is to develop a parallel solver for the B-coefficients of the multivariate simplex splines. The main challenge in developing such a parallel solver is that the enforcement of continuity between simplices is a serial task, at least at this point in time.

Finally, it is recommended to develop methods for formulating equality and inequality constraints that have a physical meaning. It would be desirable for aerodynamic model identification to define differential constraints that bound the values of the (directional) stability derivatives, for example C_{m_α}, to within physical limits.

7.2 System identification with simplex splines

The following recommendations are made with the aim of advancing the new methodology for system identification with multivariate simplex splines:

- An orthogonal least squares estimator should be added to the framework.
- A maximum likelihood estimator should be added to the framework.
- A global optimization method for solving the triangulation optimization and B-coefficient estimation problem in a single step should be developed.

- An ability to handle linear inequality constraints should be added to the B-coefficient estimators.
- A model-based controller based on multivariate simplex splines should be developed.
- A fault tolerant simplex spline based controller utilizing the recursive estimator should be developed.

The development of an orthogonal least squares estimator for the B-coefficients of the multivariate simplex splines should further increase the power and utility of the methodology. The same holds for the development of a maximum likelihood estimator which transforms the current equation error method into an output error method. In that case, the new methodology can be used directly on the raw flight data, bypassing the requirement for flight path reconstruction.

Another tentative development is the creation of a global optimization method for solving the triangulation optimization problem and the B-coefficient estimation problem in a single step. Such a method should be based on the global formulation of the B-form presented in this thesis. A promising research direction for such a single step method is the combination of interval analysis with the B-form in global coordinates.

The methodology for system identification with simplex splines can be made more powerful by including linear inequality constraints to the B-coefficient estimators. Linear inequality constraints could then be used to loosely bound the spline functions and their directional derivatives.

A model based control scheme should be developed that is based entirely on simplex splines. One possibility would be a nonlinear dynamic inversion controller using simplex spline based plant models. Another, even more interesting possibility is the creation of an adaptive model based controller that utilizes the differential equality constrained recursive least squares estimator presented in this thesis.

Finally, a fault tolerant controller using an onboard adaptive spline model should be developed. Such a controller would utilize the recursive least squares estimator for simplex splines, and would be able to adapt to changing system dynamics.

7.3 Aerodynamic model identification with simplex splines

The following recommendations result from the process of flight testing with the Cessna Citation II, and the subsequent identification of its aerodynamic models:

- A high quality IMU making accurate measurements at the highest possible measurement rates ($>$1kHz) is essential for accurate aerodynamic model identification.

- The linear accelerometers in the current (2011) IMU produce excessive noise caused most likely by a mechanical vibration of the entire IMU assembly. It is recommended that this source of vibration is isolated and corrected for.
- A software tool for checking simplex data coverage during flight is highly desirable for aerodynamic model identification with simplex splines.
- To increase the coverage of the spline domain with data, measurements should be made at the highest possible sampling rate. From the perspective of the new identification methodology, the more data is logged during a flight, the better.
- Attempts should be made to use the new methodology to model the effects of nonlinear aerodynamic phenomena like shock waves and flow separation.
- Dynamic simulations of the flight test maneuvers should be performed using the identified aerodynamic models, the results of which should be compared with the measured aircraft dynamics.
- The new methodology should be tested against other advanced identification methods like artificial (polynomial) neural networks.

The output from the Q-Flex 3100 linear accelerometers in current IMU aboard the PH-LAB contains noise levels that are an order of magnitude higher than manufacturer specifications. The source of this noise is unknown at the time of this writing, but an engine induced mechanical vibration of the complete IMU assembly is the prime suspect. A recommendation is to further investigate this noise source, because any external cause will also negatively influence the next generation IMU.

The single most important recommendation to be made to further increase the application of the new identification methodology is a software tool that can be used during flight for assessing the degree of coverage of the flight envelope with measurement points. Such a software tool should be operated either by the pilot or by the experiment coordinator, and should give clear visual cues as to what degree the flight envelope has been covered with measurement points. These cues can then be used to execute subsequent maneuvers. Another tentative option is to integrate the data coverage tool with the autopilot, which then automatically executes maneuvers that are designed to provide the most complete coverage of the flight envelope.

In order to test the true potential of the new methodology, it should be used to model the effects of highly nonlinear aerodynamic phenomena like shock waves and flow separation. These phenomena tend to produce discontinuities in the measured aerodynamic forces and moments, and form a challenge for any continuous modeling method. The simplex splines, however, are capable of modeling discontinuities by either locally increasing the density of the triangulation in areas of the flight envelope in which discontinuities occur, or by configuring the triangulation such that areas of discontinuity coincide with the simplex edges.

An alternative measure of model quality is the comparison between simulated and actual flight. For this, the aerodynamic models identified in this thesis should be integrated with a dynamic simulation of the Cessna Citation II. The flight test maneuvers could then be simulated using the measured initial state and control inputs. A comparison between the simulated states and the true states would then provide another measure of model quality.

The performance of the new methodology for global model identification should be compared with other global model identification methods, most notably (polynomial) neural networks. The reason that this comparison was not made in this thesis is that current implementations of these alternative methods are unsuitable for large scale, high data-volume identification experiments as they produce non-sparse solution systems. For experiments on the scale of those performed in this thesis, this would lead to extremely high demands on computational resources.

Data coverage, which is the simplex spline analogue of system excitation, is the most important practical issue during real-life global model identification with simplex splines. In order to increase data coverage, measurements should be made at a rate of at least 1kHz. In order to accommodate these higher data rates, a number of FTIS systems need to be upgraded or replaced.

Appendix A

Reference Frame Definitions

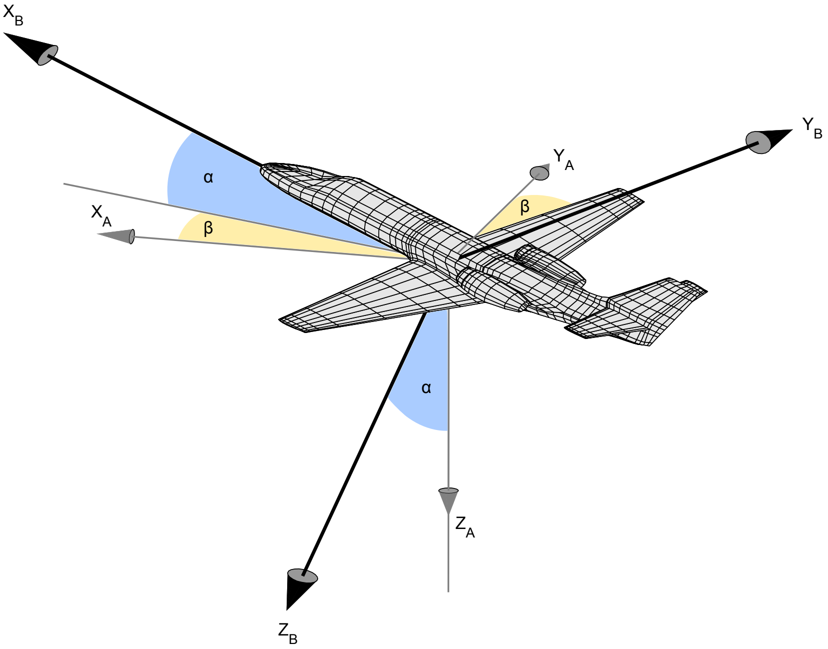


Figure A.1: The body fixed reference frame F_B and the aerodynamic reference frame F_A.

In this appendix, the reference frames used in this thesis are introduced. First, in

Figure A.1 the Body fixed reference frame F_B and the aerodynamic reference frame F_A are shown. While F_B is fixed to the aircraft, the F_A reference frame points in the direction of the incoming flow. The F_A reference frame is found by rotating F_B over α and β.

In Figure A.2 the body fixed reference frame F_B is shown together with the earth fixed reference frame F_E. The F_E reference frame in this case functions as an inertial reference frame.

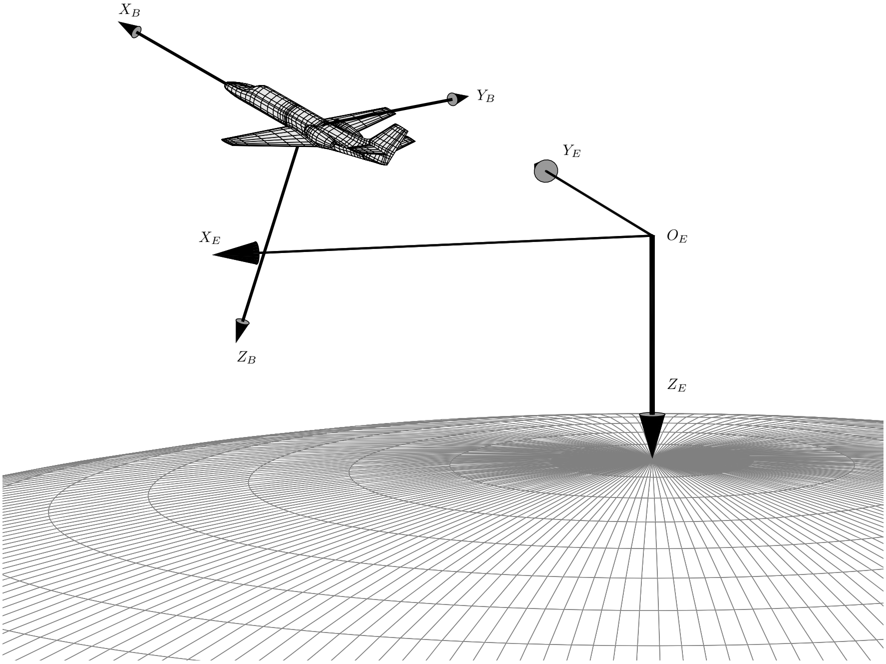


Figure A.2: The Earth fixed reference frame F_E and the body fixed reference frame F_B.

In Figure A.3 the location of the IMU in terms of the aircraft reference frame is shown. The aircraft reference frame itself is shown in Figure A.4.

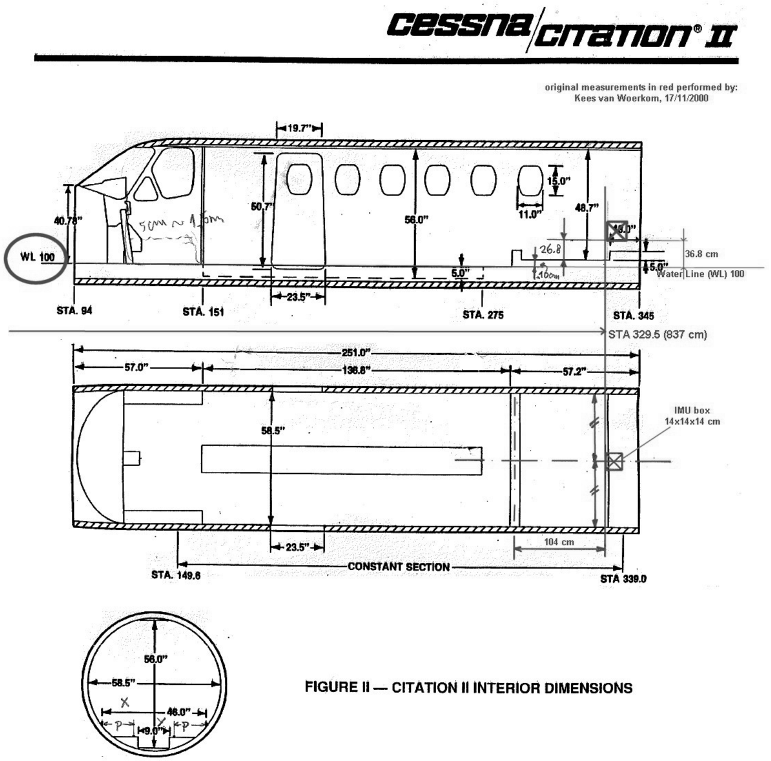


Figure A.3: Location of the IMU box in terms of the aircraft reference frame.

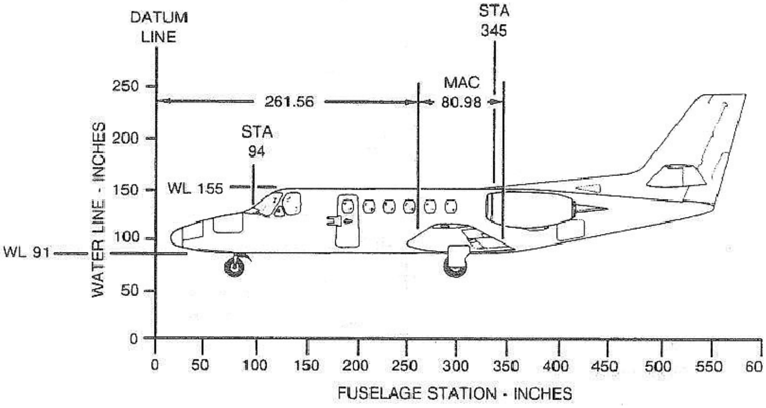


Figure A.4: Definition of the aircraft reference frame of the Cessna Citation II.

Appendix B

IMU Geometry

In this appendix, the geometric specifications of the IMU box installed in the PH-LAB are presented.

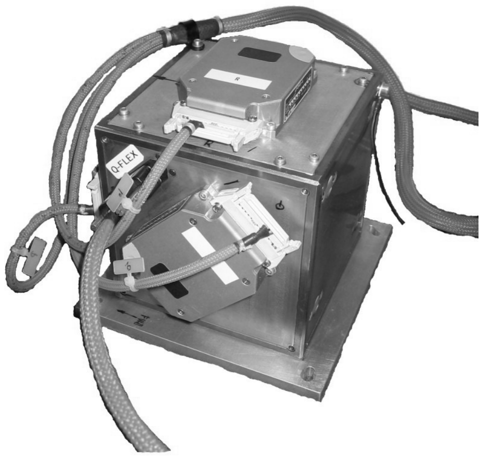
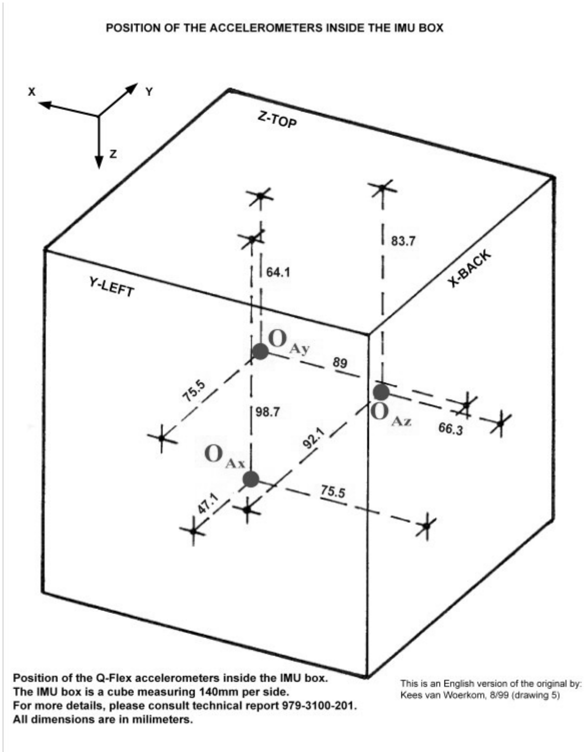


Figure B.1: The IMU box installed as installed in the PH-LAB. The LITEF fiber optic gyroscopes are fixed to the outside of the IMU box.



POSITION OF THE ACCELEROMETERS INSIDE THE IMU BOX

Position of the Q-Flex accelerometers inside the IMU box.
The IMU box is a cube measuring 140mm per side.
For more details, please consult technical report 979-3100-201.
All dimensions are in milimeters.

This is an English version of the original by:
Kees van Woerkom, 8/99 (drawing 5)

Figure B.2: Positions of the IMU accelerometers inside the IMU box.

Bibliography

[1] J. Abonyi, R. Babuška, and Ferenc Szeifert. Fuzzy modeling with multivariate membership functions: Gray-box identification and control design. *IEEE Trans. On Systems, Man, and Cybernetics*, 31:755–767, 2001.

[2] R. Ahrem, A. Beckert, and H. Wendland. A new multivariate interpolation method for large-scale spatial coupling problems in aeroelasticity. In *ECCOMAS CFD 2006*, 2006.

[3] I. J. Anderson, M. G. Cox, and J. C. Mason. Tensor-product spline interpolation to data on or near a family of lines. *Numerical Algorithms*, 5:193–204, 1993.

[4] G. Awanou. *Energy Methods in 3D spline approximations of the Navier-Stokes Equations*. PhD thesis, University of Georgia, 2003.

[5] G. Awanou and M. J. Lai. Trivariate spline approximations of 3d navier-stokes equations. *Mathematics of Computation*, 74:585–601, 2004.

[6] G. Awanou and M. J. Lai. On the convergence rate of the augmented lagrangian algorithm for nonsymmetric saddle point problems. *Applied Numerical Mathematics*, 54:122–134, 2005.

[7] G. Awanou, M. J. Lai, and P. Wenston. The multivariate spline method for scattered data fitting and numerical solutions of partial differential equations. In G. Chen and M. J. Lai, editors, *Wavelets and Splines*, pages 24–75, 2005.

[8] R. Babuška and H. Verbruggen. Neuro-fuzzy methods for nonlinear system identification. *Annual Reviews in Control*, 27:73–85, 2003.

[9] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[10] P. D. Bruce and M. G. Kellett. Modeling and identification of non-linear aerodynamic functions using b-splines. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 214:27–40, 2000.

[11] C. Cao and N. Hovakimyan. Novel l_1 neural network adaptive control architecture with guaranteed transient performance. *IEEE Transactions on Neural Networks*, 18(4):1160–1171, 2007.

[12] H.F. Chen. Recursive system identification. *Acta Mathematica Scientia*, 29B:650–672, 2009.

[13] S. Chen and S.A. Billings. Neural networks for nonlinear dynamic system modeling and identification. *Int. J. Control*, 56:319–346, 1992.

[14] L. P. Chew. Constrained delaunay triangulations. *Algorithmica*, 4:97–108, 1989.

[15] P. D. Christofides. Robust output feedback control of nonlinear singularly perturbed systems. *Automatica*, 36:45–52, 2000.

[16] C. W. Clenshaw and J. G. Hayes. Curve and surface fitting. *Journal of Applied Mathematics*, 1:164–183, 1965.

[17] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[18] R. W. Cottle. Minimal triangulations of the 4-cube. *Discrete Mathematics*, 40:25–29, 1982.

[19] M. G. Cox. The numerical evaluation of b-splines. *IMA Journal of Applied Mathematics*, 10:134–149, 1972.

[20] H. B. Curry and I. J. Schoenberg. On pólia frequency functions iv: The fundamental spline functions and their limits. *Journal d'Analyse Mathématique*, 17:71–107, 1966.

[21] W. Dahmen. Polynomials as linear combination of multivariate b-splines. *Mathematische Zeitschrift*, 169:93–98, 1979.

[22] W. Dahmen. On multivariate b-splines. *Journal of Numerical Analysis*, 17:179–191, 1980.

[23] W. Dahmen. Approximation by linear combinations of multivariate b-splines. *Journal of Approximation Theory*, 31:299–324, 1981.

[24] W. Dahmen and C. A. Micchelli. On the linear independence of multivariate b-splines in triangulations of simploids. *SIAM Journal on Numerical Analysis*, 19:993–1012, 1982.

[25] W. Dahmen, C. A. Micchelli, and H. P. Seidel. Blossoming begets b-spline bases built better by b-patches. *Mathematics of Computation*, 59:97–115, 1992.

[26] O. Davydov and L. L. Schumaker. On stable local bases for bivariate polynomial spline spaces. *Constructive Approximation*, 18:87–116, 1999.

[27] M. de Berg, O. Cheong, M. Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*, chapter Delaunay Triangulations, pages 191–218. Springer-Verlag, 2008.

[28] C. de Boor. On calculating with b-splines. *Journal of Approximation Theory*, 6:50–62, 1972.

[29] C. de Boor. Splines as linear combinations of b-splines. a survey. In G. G. Lorentz, C. K. Chui, and L. L. Schumaker, editors, *Approximation Theory II*, pages 1–47, 1976.

[30] C. de Boor. Efficient computer manipulation of tensor products. *ACM Transactions on Mathematical Software*, 5(2):173–182, 1979.

[31] C. de Boor. B-form basics. In G. Farin, editor, *Geometric modeling: algorithms and new trends*. SIAM, 1987.

[32] C. de Boor. What is a multivariate spline? In J. McKenna and R. Temam, editors, *ICIAM '87: Proceedings of the First International Conference on Industrial and Applied Mathematics*, pages 90–101, 1987.

[33] C. de Boor. *Fundamental Developments of Computer-Aided Geometric Modeling*, chapter B-spline Basics, pages 27–49. Academic Press, New York, 1993.

[34] C. de Boor. Multivariate piecewise polynomials. *Acta Numerica*, 2:65–109, 1993.

[35] C. de Boor. On the evaluation of box splines. *Numerical Algorithms*, 5:5–23, 1993.

[36] C. de Boor. *A Practical Guide to Splines (Revised Edition)*. Springer-Verlag New York, Inc., 2001.

[37] C. de Boor. On interpolation by radial polynomials. *Advances in Computational Mathematics*, 24:143–153, 2006.

[38] C. de Boor and A. Ron. On multivariate polynomial interpolation. *Constructive Approximation*, 6(3):287–302, 1990.

[39] C. de Boor and A. Ron. Polynomial ideals and multivariate splines. In C. Chui, W. Schempp, and K. Zeller, editors, *ISNM 90, Multivariate Approximation Theory IV*, volume 25, pages 31–40, 1991.

[40] C. C. de Visser, Q. P. Chu, and J. A. Mulder. A new approach to linear regression with multivariate splines. *Automatica*, 45(12):2903–2909, 2009.

[41] C. C. de Visser, Q. P. Chu, and J. A. Mulder. Differential constraints for bounded recursive identification with multivariate splines. *Automatica*, 2011. article in press.

[42] C.C. de Visser, J. A. Mulder, and Q. P. Chu. A multidimensional spline based global nonlinear aerodynamic model for the Cessna Citation II. In *AIAA Atmospheric Flight Mechanics Conference*, 2010.

[43] C.C. de Visser, J.A. Mulder, and Q.P. Chu. Global nonlinear aerodynamic model identification with multivariate splines. In *AIAA Atmospheric Flight Mechanics Conference*, 2009.

[44] E. de Weerdt, Q. P. Chu, and J. A. Mulder. Neural network aerodynamic model identification for aerospace reconfiguration. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[45] E. de Weerdt, C. C. de Visser, Q. P. Chu, and J. A. Mulder. Fuzzy simplex splines. In *IFAC SYSID 2009*, 2009.

[46] E. de Weerdt, E. R. van Oort, Q. P. Chu, and J. A. Mulder. Level set method based on interval analysis. In *AIAA GNC Conference*, 2011. Submitted for publication.

[47] T. K. Dey, C. L. Bajaj, and K. Sugihara. On good triangulations in three dimensions. In *SMA '91: Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications*, pages 431–441, New York, NY, USA, 1991. ACM.

[48] S. Drake. *Galileo at Work: His Scientific Biography*. Courier Dover Publications, 2003.

[49] N. Dyn and A. Ron. Radial basis function approximation: from gridded centers to scattered centers. *Proc. London Mathematical Society*, 71:76–108, 1995.

[50] M. Espinoza, J.A.K. Suykens, and B. de Moor. Kernel based partial linear models and nonlinear identification. *IEEE Transactions On Automatic Control*, 50:1602–1606, 2005.

[51] H. Y. Fan, G. S. Dulikravich, and Z. X. Han. Aerodynamic data modeling using support vector machines. *Inverse Problems in Science and Engineering*, 13:261–278, 2005.

[52] G. Farin. Triangular bernstein-bézier patches. *Computer Aided Geometric Design*, 3:83–127, 1986.

[53] G. Farin. *Handbook of Computer Aided Geometric Design*, chapter A History of Curves and Surfaces in CAGD, pages 1–21. Elsevier Science B.V., 2002.

[54] S. Ferrari and R. F. Stengel. Smooth function approximation using neural networks. *IEEE Transactions On Neural Networks*, 16(1):24–38, 2005.

[55] P. Fong and H. P. Seidel. Control points for multivariate b-spline surfaces over arbitrary triangulations. *Computer Graphics Forum*, 10:309–317, 1991.

[56] A. R. Forrest. Interactive interpolation and approximation by bézier polynomials. *Computer Aided Design*, 22(9):527–537, 1990. Reprint of The Computer Journal, 15(1):71-79, 1972.

[57] D. Fowler. The binomial coefficient function. *The American Mathematical Monthly*, 103(1):1–17, 1996.

[58] R. Franke. A critical comparison of some methods for interpolation of scattered data. Technical report, Naval Postgraduate School, 1979.

[59] R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, 38(157):181–200, 1982.

[60] R. Franke. Smooth interpolation of scattered data by local thin plate splines. *Computers & Mathematics with Applications*, 8:273–281, 1982.

[61] R. Franke and G. Nielson. Smooth interpolation of large sets of scattered data. *International Journal for Numerical Methods in Engineering*, 15:1691–1704, 1980.

[62] Galileo Galilei. *Sidereus Nuncius (The Starry Messenger)*. Venice, 1610. English translation by A. van Helden.

[63] W. L. Garrard, D. F. Enns, and S. A. Snell. Nonlinear feedback control of highly manoeuvrable aircraft. *Int. J. Control*, 56:799–812, 1992.

[64] O. H. Gerlach. Determination of performance and stability parameters from non-steady flight test manoeuvres. In *SAE Paper 700236*, 1970.

[65] G. C. Goodwin and R. L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, 1977.

[66] Grandine. The computational cost of simplex spline functions. *SIAM Journal on Numerical Analysis*, 24:887–890, 1987.

[67] H. Hakopian. Multivariate splines functions, b-spline basis and polynomial interpolations. *SIAM Journal on Numerical Analysis*, 19:510–517, 1982.

[68] Y. Han and Y. Kim. Optimal input design for online parameter estimation for aircraft with multiple control surfaces. *Engineering Optimization*, pages 1–22, 2011.

[69] E. Hansen and G.W. Walster. *Global Optimization Using Interval Analysis*. Marcel Dekker, Inc. and Sun Microsystems, Inc., 2004. ISBN 0-8247-4059-9.

[70] M. S. Hansen, B. Glocker, N. Navab, N. Navab, R. Larsen, and R. Larsen. Adaptive parametrization of multivariate b-splines for image registration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[71] R. L. Harder and R. N. Desmarais. Interpolation using surface splines. *AIAA Journal of Aircraft*, 9:189–191, 1972.

[72] S. Haykin. *Neural Networks*. Prentice Hall, Inc., 1999.

[73] N. J. Higham and F. Tisseur. A block algorithm for matrix 1-norm estimation with an application to 1-norm pseudospectra. *SIAM Journal of Matrix Analysis Applications*, 21:1185–1201, 2000.

[74] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[75] X. L. Hu, D. F. Han, and M. J. Lai. Bivariate splines of various degrees for numerical solution of partial differential equations. *SIAM Journal on Scientific Computing*, 29:1338–1354, 2007.

[76] J. Hua, Y. He, and H. Qin. Trivariate simplex splines for inhomogeneous solid modeling in engineering design. *Journal of Computing and Information Science in Engineering*, 5(5):149–157, 2005.

[77] R. B. Hughes and M. R. Anderson. Simplexity of the cube. *Discrete Mathematics*, 158:99–150, 1996.

[78] E. Husserl. *Ideas Pertaining to a Pure Phenomenology and to a Phenomenological Philosophy*. Martinus Nijhoff Publishers, 1913. Translated by F. Kersten (1983).

[79] E. Husserl. *Psychological and Transcendental Phenomenology and the Confrontation with Heidegger*. Springer, 1997.

[80] M. Hwang and J. H. Seinfeld. Observability of nonlinear systems. *Journal of Optimization Theory and Applications*, 10:67–77, 1972.

[81] K. W. Iliff, R.E. Maine, and T.D. Montgomery. Important factors in the maximum likelihood analysis of flight test maneuvers. Technical Paper 1459, NASA, 1979.

[82] K. W. Iliff and L. W. Taylor. Determination of stability derivatives from flight data using a newton-raphson minimization technique. Technical Report TN D-6579, NASA, 1972.

[83] Kockelmans J. J. *Phenomenology; the philosophy of Edmund Husserl and its interpretation. Edited by Joseph J. Kockelmans*. Doubleday, 1967.

[84] R. V. Jategaonkar. *Flight Vehicle System Identification*, volume 216 of *Progress in Astronautics and Aeronautics*. AIAA, 2006.

[85] C. Jauberthie, L. Denis-Vidal, P. Coton, and G. Joly-Blanchard. An optimal input design procedure. *Automatica*, 42:881–884, 2006.

[86] T. Kalbe, T. Koch, and M. Goesele. High-quality rendering of varying isosurfaces with cubic trivariate C^1-continuous splines. In *ISVC 2009*, volume 5875, pages 596–607, 2009.

[87] Takeaki Kariya and Hiroshi Kurata. *Generalized Least Squares*. John Wiley & Sons, Inc., 2004.

[88] V. Klein. Estimation of aircraft aerodynamic parameters from flight data. *Progress in Aerospace Sciences*, 26:1–77, 1989.

[89] V. Klein and J. G. Batterson. Determination of airplane model structure from flight data using splines and stepwise regression. Technical Report 2126, NASA, 1983.

[90] V. Klein, J. G. Batterson, and P. C. Murphy. Determination of aircraft model structure from flight data by using modified stepwise regression. Technical Report 1916, NASA, 1981.

[91] V. Klein and J. R. Schiess. Compatibility check of measured aircraft responses using kinematic equation and extended kalman filter. Technical Report TN D-8514, NASA, 1977.

[92] Vladislav Klein and Eugene A. Morelli. *Aircraft System Identification*. AIAA, 2006.

[93] Elias Kyriakides and Gerald T. Heydt. Calculating confidence intervals in parameter estimation: A case study. *IEEE Transactions on Power Delivery*, 21:508–509, 2006.

[94] M. Laban. *Online Aerodynamic Model Identification*. PhD thesis, Delft University of Technology, 1994.

[95] M. J. Lai. Scattered data interpolation and approximation by using bivariate c^1 piecewise cubic polynomials. *Computer Aided Geometric Design*, 13:81–88, 1996.

[96] M. J. Lai. Geometric interpretation of smoothness conditions of triangular polynomial patches. *Computer Aided Geometric Design*, 14:191–199, 1997.

[97] M. J. Lai. Macro-elements and stable local bases for splines on clough-tocher triangulations. *Numerische Mathematik*, 88:105–119, 2001.

[98] M. J. Lai. The convergence of three l1 spline methods for scattered data interpolation and fitting. *Journal of Approximation Theory*, 145:196–211, 2007.

[99] M. J. Lai. Multivariate splines for data fitting and approximation. In *12th Approximation Theory Conference*, 2007.

[100] M. J. Lai and L. L. Schumaker. Scattered data interpolation using c2 supersplines of degree six. *SIAM Journal on Numerical Analysis*, 34(3):905–921, 1997.

[101] M. J. Lai and L. L. Schumaker. On the approximation power of bivariate splines. *Advances in Computational Mathematics*, 9:251–279, 1998.

[102] M. J. Lai and L. L. Schumaker. On the approximation power of splines on triangulated quadrangulations. *SIAM Journal on Numerical Analysis*, 36:143–159, 1999.

[103] M. J. Lai and L. L. Schumaker. Macro-elements and stable local bases for splines on powell-sabin triangulations. *Mathematics of Computation*, 72(241):335–354, 2003.

[104] M. J. Lai and L. L. Schumaker. *Spline Functions on Triangulations*. Cambridge University Press, 2007.

[105] M. J. Lai and L. L. Schumaker. A domain decomposition method for computing bivariate spline fits of scattered data. *SIAM Journal on Numerical Analysis*, 47:911–928, 2009.

[106] M. J. Lai and P. Wenston. Bivariate spline method for navier-stokes equations: Domain decomposition technique. In C. K. Chui and L. L. Schumaker, editors, *Approximation Theory IX: computational Aspects*, 1998.

[107] M. J. Lai and P. Wenston. Bivariate splines for fluid flows. *Computers and Fluids*, 33:1047–1073, 2004.

[108] M. J. Lai and P. Wenston. L1 spline methods for scattered data interpolation and approximation. *Advances in Computational Mathematics*, 21:293–315, 2004.

[109] M.J. Lai. Some sufficient conditions for convexity of multivariate bernstein-b'ezier polynomials and box spline surfaces. *Studia Scient. Math. Hung.*, 28:363–374, 1990.

[110] C. E. Lan and J. M. Brandon. Development of non-linear aerodynamic models from flight data and evaluation of tabulated aerodynamic models. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2003.

[111] S. H. Lane and R. F. Stengel. Flight control design using non-linear inverse dynamics. *Automatica*, 24:471–483, 1988.

[112] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice Hall, Inc., 1995.

[113] D. T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer and Information Sciences*, 9(3):219–242, 1980.

[114] X. Y. Li, G. Calinescu, P. J. Wan, and Y. Wang. Localized delaunay triangulation with application in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(10):1035–1047, 2003.

[115] D. J. Linse and R. F. Stengel. Identification of aerodynamic coefficients using computational neural networks. *Journal of Guidance, Control, and Dynamics*, 16(6):1018–1025, 1993.

[116] C. M. Linton. *From Eudoxus to Einstein: A History of Mathematical Astronomy*. Cambridge University Press, 2004.

[117] L. Ljung. Recursive identification algorithms. *Circuits Systems Signal Processing*, 21:57–68, 2002.

[118] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, 1983.

[119] Lennart Ljung. *System Identification Theory for the User*. Prentice Hall, Inc., second edition, 1999.

[120] G. E. R. Lloyd. *Aristotle: the Growth and Structure of His Thought*. Cambridge University Press, 1968.

[121] T. J. J. Lombaerts. *Fault Tolerant Flight Control: A Physical Model Approach*. PhD thesis, Delft University of Technology, 2010.

[122] T. J. J. Lombaerts, Q. P. Chu, J. A. Mulder, and D. A. Joosten. Real time damaged aircraft model identification for reconfiguring flight control. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2007.

[123] T. J. J. Lombaerts, H. O. Huisman, Q. P. Chu, J. A. Mulder, and D. A. Joosten. Flight control reconfiguration based on online physical model identification and nonlinear dynamic inversion. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[124] T. J. J. Lombaerts, H. O. Huisman, Q. P. Chu, J. A. Mulder, and D. A. Joosten. Nonlinear reconfiguring flight control based on online physical model identification. *Journal Of Guidance, Control, and Dynamics*, 32:727–748, 2009.

[125] T. J. J. Lombaerts, E. R. Van Oort, Q. P. Chu, J. A. Mulder, and D. A. Joosten. Online aerodynamic model structure selection and parameter estimation for fault-tolerant control. *Journal Of Guidance, Control, and Dynamics*, 33:707, 2010.

[126] B. Lubbers. A model of the experimental fly-by-wire control system for the ph-lab. Master's thesis, Delft University of Technology, 2009.

[127] K. G. Magnusson. Observability of nonlinear systems. *IMA Journal of Mathematical Control & Information*, 1:339–358, 1984.

[128] P. S. Mara. Triangulations for the cube. *Journal of Combinatorial Theory*, 20:170–177, 1976.

[129] S. S. McGaugh. A novel test of the modified newtonian dynamics with gas rich galaxies. *Physical Review Letters*, pages 1–4, 2011. Article in Press.

[130] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *Zeitschrift für Angewandte Mathematik und Physik*, 30:292–304, 1978.

[131] C. A. Micchelli. A constructive approach to kergin interpolation in $\mathbb{R}^k$: Multivariate b-splines and lagrange interpolation. *Rocky Mountain Journal of Mathematics*, 10(3):485–497, 1980.

[132] C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, 1986.

[133] M. Milgrom. A modification of the newtonian dynamics as a possible alternative to the hidden mass hypothesis. *The Astrophysical Journal*, 270:365–370, 1983.

[134] D. T. Mirikitani and N. Nikolaev. Recursive bayesian recurrent neural networks for time-series modeling. *IEEE Transactions on Neural Networks*, 21:262–274, 2010.

[135] S. A. Mitchell and S. A. Vavasis. Quality mesh generation in three dimensions. In *SCG '92: Proceedings of the eighth annual symposium on Computational geometry*, pages 212–221, New York, NY, USA, 1992. ACM.

[136] A.F. Möbius. *Der Barycentrischer Calcul.* Verlag von Johann Ambrosius Barth, Leipzig, 1827.

[137] R.E. Moore. *Interval Analysis*. Prentice-Hall, Inc., 1966.

[138] E. A. Morelli. Global nonlinear aerodynamic modeling using multivariate orthogonal functions. *AIAA Journal of Aircraft*, 32:270–277, 1995.

[139] E. A. Morelli. Piloted parameter identification flight test maneuvers for closed loop modeling of the f-18 high alpha research vehicle (harv). Technical Report 198268, NASA, 1996.

[140] E. A. Morelli. Flight test validation of optimal input design and comparison to conventional inputs. Technical Report 887652, NASA, 1997.

[141] E. A. Morelli. Flight test validation of optimal input design and comparison to conventional inputs. *AIAA Journal of Aircraft*, pages 1–12, 1997.

[142] E. A. Morelli. Global nonlinear parametric modeling with application to f-16 aerodynamics. In *American Control Conference*, 1998.

[143] E. A. Morelli and V. Klein. Optimal input design for aircraft parameter estimation using dynamic programmng principles. In *AIAA AFM Conference*, 1990.

[144] E. A. Morelli and V. Klein. Accuracy of aerodynamic model parameters estimated from flight test data. *Journal of Guidance Control and Dynamics*, 20:74–80, 1997.

[145] T. I. Mueller. *Geometric Modeling with Multivariate B-Splines.* PhD thesis, University of Utah, 1986.

[146] J. A. Mulder. Estimation of the aircraft state in non-steady flight. In *AGARD CP-172 on Methods for Aircraft State and Parameter Identification*, 1974.

[147] J. A. Mulder. *Design and Evaluation of Dynamic Flight Test Maneuvers.* PhD thesis, Delft University of Technology, 1986.

[148] J. A. Mulder. Design and evaluation of dynamic flight test maneuvers. Technical report, Delft University of Technology, 1986.

[149] J. A. Mulder, Q. P. Chu, J. K. Sridhar, J. H. Breeman, and M. Laban. Non-linear aircraft flight path reconstruction review and new advances. *Progress in Aerospace Sciences*, 35:673–726, 1999.

[150] M. Mulder, B. Lubbers, P. M. T. Zaal, M. M. van Paassen, and J. A. Mulder. Aerodynamic hinge moment coefficient estimation using automatic fly-by-wire control inputs. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2009.

[151] M. Neamtu. On discrete simplex splines and subdivision. *Journal of Approximation Theory*, 70:358–374, 1992.

[152] M. Neamtu. Bivariate simplex b-splines: A new paradigm. In R. Durikovic and S. Czanner, editors, *Proc. of Spring Conference on Computer Graphics, IEEE Computer Society*, pages 71–78, 2001.

[153] M. Neamtu. What is the natural generalization of univariate splines to higher dimensions? In T. Lyche and L. L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces*, pages 355–392, 2001.

[154] M. Neamtu. Delaunay configurations and multivariate splines: A generalization of a result of b. n. delaunay. *Transactions of the American Mathematical Society*, 359:2993–3004, 2007.

[155] L. T. Nguyen, M. E. Ogburn, W .P. Gilbert, K. S. Kibler, P. W. Brown, and P. L. Deal. Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability. Technical Report 1538, NASA, 1979.

[156] F. H. Norton. The measurement of the damping in roll on a jn-4h in flight. Technical report, N.A.C.A, 1923.

[157] F. H. Norton. A study of longitudinal dynamic stability in flight. Technical report, N.A.C.A., 1923.

[158] H. Oelker and O. Brieger. Flight test experiences with eurofighter typhoon during high bandwidth pio resistance testing. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2006.

[159] H. Oelker, B. Renzo, G. T. Hoare, and G. Garcia-Mesuro. Experiences on aerodynamic parameters identification for eurofighter at daimler-benz aerospace, alenia, british aerospace and casa. In *RTO SCI symposium on System Identification for Integrated Aircraft Development and Flight Testing*, 1998.

[160] E. Oezger. Investigation on the influence of time shifts of measured input signals on parameter estimation results. In *DGLR Conference Paper*, 2006.

[161] E. Oezger. Aerodynamic model validation of eurofighter aircraft. In *AIAA Conference Paper*, 2007.

[162] E. Oezger and E. Meyer. Aerodynamic model validation at eads military aircraft. In *DGLR Conference*, 2005.

[163] H. Ogawa. An operator pseudo-inversion lemma. *SIAM Journal on Applied Mathematics*, 48(6):1527–1531, 1988.

[164] S. K. Oh, W. D. Kim, W. Pedrycz, and B. J. Park. Polynomial-based radial basis function neural networks (p-rbf nns) realized with the aid of particle swarm optimization. *Fuzzy Sets and Systems*, 2010. Article in Press.

[165] S. K. Oh, W. Pedrycz, and S. B. Roh. Hybrid fuzzy set-based polynomial neural networks and their development with the aid of genetic optimization and information granulation. *Applied Soft Computing*, 9:1068–1089, 2009.

[166] D. Orden and F. Santos. Asymptotically efficient triangulations of the d-cube. *Discrete and Computational Geometry*, 30:509–528, 2003.

[167] R. Panda and B. N. Chatterji. Least squares generalized b-spline signal and image processing. *Signal Processing*, 81:2005–2017, 2001.

[168] B. J. Park, W. Pedrycz, and S. K. Oh. Fuzzy polynomial neural networks: Hybrid architectures of fuzzy modeling. *IEEE Transactions on Fuzzy Systems*, 10:607–621, 2002.

[169] B. J. Park, W. Pedrycz, and S. K. Oh. Polynomial-based radial basis function neural networks and their application to pattern classification. *Applied Intelligence*, 32(1):27–46, 2010.

[170] O. Pedersen. *A Survey of the Almagest*. Springer, 2010.

[171] R. Pfeifle and H.P. Seidel. Fitting triangular b-splines to functional scattered data. *Computer Graphics Forum*, 15:15–23, 1996.

[172] M. J. D. Powell. The uniform convergence of thin plate spline interpolation in two dimensions. *Numerische Mathematik*, 68:107–128, 1994.

[173] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003.

[174] B. Ram. *Computer fundamentals: Architecture and Organization*. New Age International Ltd., third edition, 2000.

[175] L. Ramshaw. Blossoming: A connect-the-dots approach to splines. Technical report, HP Labs, 1987.

[176] L. Ramshaw. Blossoms are polar forms. Technical report, HP Labs, 1989.

[177] C. Radhakrishna Rao. *Linear Statistical Inference and Its Applications*. John Wiley & Sons, Inc., 2002.

[178] J. Reiner, G. J. Balas, and W. L. Garrard. Robust dynamic inversion for control of highly maneuverable aircraft. *Journal of Guidance, Control and Dynamics*, 18:18–24, 1995.

[179] J. Reiner, G. J. Balas, and W. L. Garrard. Flight control using robust dynamic inversion and time-scale separation. *Automatica*, 32:1493–1504, 1996.

[180] J. Richalet. Industrial applications of model based predictive control. *Automatica*, 29(5):1251 – 1274, 1993.

[181] D. Rohlf. Global model approach for x-31 vector system identification. *AIAA Journal of Aircraft*, 42, 2005.

[182] J. Roll, A. Nazin, and L. Ljung. Nonlinear system identification via direct weight optimization. *Automatica*, 41:475–490, 2005.

[183] J. Ruppert. A delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18:548–585, 1995.

[184] J. F. Sallee. A triangulation of the n-cube. *Discrete Mathematics*, 40:81–86, 1982.

[185] R. Schaback. Approximation by radial basis functions with finitely many centers. *Constructive Approximation*, 12(3):331–340, 1996.

[186] R. Schaback. Improved error bounds for scattered data interpolation by radial basis functions. *Mathematics of Computation*, 68(225):201–216, 1999.

[187] I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. *Quarterly Applied Mathematics*, 4:45–99, 112–141, 1946.

[188] I. J. Schoenberg. *Inequalities I*, chapter On Spline Functions, pages 255–291. Academic Press (New York), 1967.

[189] B. Schölkopf, J. Giesen, and S. Spalinger. Kernel methods for implicit surface modeling. In *Advances in Neural Information Processing Systems 17*, pages 1193–1200. MIT Press, 2005.

[190] F. Schröder and P. Robach. Managing the complexity of digital terrain models. *Computers & Graphics*, 18(6):775–783, 1994.

[191] H. P. Seidel. Symmetric recursive algorithms for surfaces: B-patches and the de boor algorithm for polynomials over triangles. *Constructive Approximation*, 7:257–279, 1991.

[192] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22:21–74, 2001.

[193] J. R. Shewchuk. What is a good linear element? interpolation, conditioning, and quality measures. In *Proceedings, 11th International Meshing Roundtable*, pages 115–126, 2002.

[194] J. R. Shewchuk. General-dimensional constrained delaunay and constrained regular triangulations i: Combinatorial properties. *Discrete and Computational Geometry*, 39(1):580–637, 2008.

[195] H. Si and K. Gärtner. An algorithm for three-dimensional constrained delaunay tetrahedralizations. In *Proceeding of the Fourth International Conference on Engineering Computational Technology*, 2004.

[196] H. Si and K. Gärtner. Meshing piecewise linear complexes by constrained delaunay tetrahedralizations. In *Proceeding of the 14th International Meshing Roundtable*, 2005.

[197] D. E. Smith. *History of Modern Mathematics*. John Wiley & Sons, Inc., 1906.

[198] P. L. Smith. Curve fitting and modeling with splines using statistical variable selection techniques. Contractor Report NASA-CR-166034, NASA, 1982.

[199] T. Söderström, L. Ljung, and I. Gustavsson. A theoretical analysis of recursive identification methods. *Automatica*, 14:231–244, 1978.

[200] L. Sonneveldt. Nonlinear flight control design using constrained adaptive backstepping. *Journal Of Guidance, Control, and Dynamics*, 30(2):322–336, 2007.

[201] L. Sonneveldt. Nonlinear adaptive backstepping trajectory control. *Journal Of Guidance, Control, and Dynamics*, 32(1):25–39, 2009.

[202] L. Sonneveldt. *Adaptive Backstepping Flight Control For Modern Fighter Aircraft*. PhD thesis, Delft University of Technology, 2010.

[203] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, Inc., 2003.

[204] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005.

[205] L. W. Taylor, K. W. Iliff, and B. G. Powers. A comparison of newton-raphson and other methods for determining stability derivatives from flight data. *AIAA Paper*, pages 69–315, 1969.

[206] G. J. Toomer. The size of the lunar epicycle according to hipparchus. *Centaurus*, 12:145–150, 1968.

[207] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part i-theory. *IEEE Trans. Signal Processing*, 41:821–833, 1993.

[208] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part ii-efficient design and applications. *IEEE Transactions on Signal Processing*, 41(2):834–848, 1993.

[209] E. van Kampen. *Global Optimization using Interval Analysis*. PhD thesis, Delft University of Technology, 2010.

[210] E. van Kampen, E. de Weerdt, Q.P. Chu, and J.A. Mulder. Applied interval based integer ambiguity resolution. *ION Journal of Navigation*, 56:205–219, 2009.

[211] E. R. van Oort. *Adaptive Backstepping Control And Safety Analysis For Modern Fighter Aircraft*. PhD thesis, Delft University of Technology, 2011.

[212] J. R. Voelkel. *The Composition of Kepler's Astronomia Nova*. Princeton University Press, 2001.

[213] M. von Golitschek, M. J. Lai, and L. L. Schumaker. Error bounds for minimal energy bivariate polynomial splines. *Numerische Mathematik*, 93:314–331, 2002.

[214] B. L. Walcott, M. J. Corless, and S. H. Żak. Comparative study of non-linear state observation techniques. *Int. J. Control*, 45:2109–2132, 1987.

[215] R. T. Whitcomb. A study of the zero-lift drag-rise characteristics of wing-body combinations near the speed of sound. Technical Report 1273, NACA, 1952.

[216] J. Yoon. *Approximation to Scattered Data by Radial Basis Function*. PhD thesis, University of Wisconsin, 1998.

[217] J. Yoon. Computational aspects of approximation to scattered data by using 'shifted' thin plate splines. *Advances in Computational Mathematics*, 14:329–359, 2001.

[218] J. Yoon. Lp-error estimates for 'shifted' surface spline interpolation on sobolev space. *Mathematics of Computation*, 72:1349–1367, 2002.

[219] P. M. T. Zaal, D. M. Pool, M. Mulder, M. M. van Paassen, and J. A. Mulder. Identification of multimodal pilot control behavior in real flight. *Journal Of Guidance, Control, and Dynamics*, 33(5):1527–1538, 2010.

[220] Y. Zhu and X. R. Li. Recursive least squares with linear constraints. *Communications In Information And Systems*, 7:287–312, 2007.

[221] F. Zwicky. On the masses of nebulae and of clusters of nebulae. *The Astronomical Journal*, 86:217–246, 1937.

Index

n-cube, 66

Aerodynamic Model, 7, 261, 286
    Model Structure, 206
AMI, 7, 199
    Procedure, 199, 261

B-coefficients, 56, 262
    Bounds on, 107
    Continuity Body, 111
    Continuity point, 111
    Vertex B-coefficients, 73
B-form, 52
    Global B-form, 96, 98
B-net, 58
    Orientation Rule, 74
    Propagation, 124
B-spline, 23
Barycentric coordinates, 44
Bernstein Polynomial, 49
Bonferroni Method, 181
Bounded Model Extrapolation, 142
Box spline, 32

Continuity
    Structure of Continuity, 111
Continuity Conditions, 108

Data Coverage Problem, 300, 325
Data Sifting Matrix, 97
Degree Raising, 99
Delaunay condition, 70
Differential Constraints, 137
Directional Derivative, 101
    Directional Coordinate, 101
    Matrix Form, 103
Dispersion Matrix, 169
DMS spline, 33
DSM, 97

Edge Facet, 42, 65, 108
Engine Model, 246
Equant, 3
Equations of Motion, 203
    Force Equations, 203
    Moment Equations, 203

First Principles, 1, 3, 5
Fisher Information Matrix, 171
Flight Envelope, 223
Flight Path Reconstruction, 199
Flight Test Design, 203
Force Coefficients, 205

Global Model, 6

Hypercube, 66
Hysteresis, 263

Inertial Measurement Unit, 227, 229, 240
Input signals
  3211, 213
  Doublet, 213
  Step inputs, 213
Instrumentation, 201
  Accelerometers, 227

Kalman Filter, 235
  Extended Kalman Filter, 236
  Iterated Extended Kalman Filter, 236
Karush-Kuhn-Tucker System, 169

Lie Derivative, 245

Mass Model, 247
Model Based Control, 4
Model Extrapolation, 141
Multi-index, 49
Multilinear Combinations, 208

Observability, 244
  Matrix, 245
Out Of Edge Vertices, 109

Partition of Unity, 51
Pascal's Simplex, 50
Pascal's Tetrahedron, 50
Pascal's Triangle, 42
Pentachoron, 42
Phenomenological Model, 1
Polytope, 39
Propagation, 124

Qhull, 68

Region of Validity, 225, 289
Relative Residual RMS, 178

Simplex, 39
  metrics, 62
  Polynomial, 49, 56
  Sliver Simplex, 64
Simplex Star, 75
Smoothness Constraints, 108
Smoothness Matrix, 132
  Full Rank, 134
Specific Forces, 205
Star, 75
State Plane, 209
System Identification, 5, 7, 161, 165, 182

Test Points, 210
Triangulation, 62, 261
  CDT, 76
  Delaunay, 70
  HCI method, 78
  optimization, 76
  Type I/II, 66
TSM, 198, 234
Tuple, 40
Two-Step Method, 198, 234

Valid Extrapolation Region, 141
VER, 141
Vertex, 39

Samenvatting

Globale Non-lineaire Model Identificatie met Multivariate Splines

Op dit moment spelen modelgebaseerde regelaars een essentiële rol in veel aspecten van de moderne samenleving. Gebieden van toepassing van modelgebaseerde regelaars variëren van de levensmiddelenindustrie tot medische scan apparatuur, en van procesbeheersing in olieraffinaderijen tot de besturingssystemen aan boord van moderne vliegtuigen. Het hart van een modelgebaseerde regelaar is een wiskundig model van het fysieke systeem of proces dat wordt gecontroleerd. Het gebied van de wetenschap die zich bezighoudt met de identificatie van modellen van fysische systemen heet systeem identificatie. In dit proefschrift wordt een nieuwe methode voorgesteld voor de identificatie van modellen van niet-lineaire systemen met complexe dynamica met behulp van multivariate simplex splines. Deze nieuwe methode heeft de potentie om de prestaties van elke modelgebaseerde regelaar te verbeteren door de accuratesse van de systeem modellen te verhogen.

Het is een uitdagende taak om systemen te modelleren met non-lineaire dynamica. Op dit moment bestaat er slechts een handvol methodes die correcte modellen kunnen maken van zulke systemen. Van deze methodes zijn de vier meest bekende de neurale netwerken, kernfunctie methodes, polynoom mengmethodes en spline methodes. Al deze methodes kunnen modellen produceren met een willekeurig hoge benaderingsmethode op een globale modelschaal. Tot voor kort hadden deze methodes echter inherente tekortkomingen. Neurale netwerken zijn in wezen *black-box* modellen die globale basisfuncties gebruiken. Dit resulteert in complexe en inefficiënte algoritmes voor hun training en evaluatie. Kernfunctie methodes zijn van nature niet parametrisch, wat betekent dat er in principe net zoveel kernfuncties zijn als datapunten. Dit leidt tot inefficiënte algoritmes voor grote datasets. Polynoom mengmethodes gebruiken *fuzzy logic* technieken om lokale polynoom modellen te vermengen tot één globaal model. De afstemming van de *fuzzy* menging is gebaseerd op de kennis van deskundigen, waardoor het onwaarschijnlijk is dat deze techniek ooit geheel geautomatiseerd kan worden. Polynome spline methodes zijn in het verleden met succes gebruikt om modellen te maken van non-lineaire systemen. Deze spline methodes gebruiken echter multivariate tensor product B-splines, die niet alleen beperkt zijn tot modellering op rechthoekige domeinen, maar daarnaast niet gebruikt kunnen worden om verstrooide data te benaderen.

De nieuwe methode die in dit proefschrift wordt voorgesteld, is gebaseerd op multivariate simplex splines. Dit zijn een nieuw soort multivariate splines die een aantal belangrijke voordelen hebben ten opzichte van de genoemde methodes. Ten eerste hebben simplex splines een lokale polynoom basis. Dit betekent dat slechts kleine deelverzamelingen van parameters en basispolynomen overwogen hoeven worden tijdens de schatting en evaluatie, wat resulteert in efficiënte algoritmes. Ten tweede zijn simplex spline modellen parametrische modellen, die voor een efficiënte benadering zorgen van zeer grote datasets. Ten derde zijn simplex splines lineair in de parameters. Dit houdt in dat lineaire regressiemethodes gebruikt kunnen worden om hun parameters te schatten. Ten vierde worden simplex splines bepaald op niet-rechthoekige domeinen en kunnen ze gebruikt worden om verstrooide data te benaderen. Ten slotte kan de kwaliteit van op simplex splines gebaseerde modellen vastgesteld worden door gebruik te maken van een aantal unieke en krachtige methodes voor de bepaling van de modelkwaliteit.

Multivariate simplex splines bestaan uit polynome basisfuncties, ook wel B-polynomen genoemd. De B-polynomen zijn gedefinieerd op simplices, wat speciale geometrische structuren zijn. Elke simplex draagt één B-polynoom, die weer uit een lineaire combinatie van Bernstein basispolynomen bestaat. Iedere Bernstein basispolynoom wordt geschaald door één coëfficiënt, de zogenaamde B-coëfficiënt. De B-coëfficiënten hebben een bijzondere eigenschap in die zin dat ze een unieke ruimtelijke locatie hebben binnen hun ondersteunende simplex. Deze ruimtelijke structuur, die ook wel bekend staat als het B-net, biedt een aantal unieke mogelijkheden die pleiten voor het gebruik van simplex splines als een instrument voor modelering. Het B-net vereenvoudigt bijvoorbeeld de lokale modificatie van modellen door direct specifieke modelgebieden te relateren aan subsets van B-coëfficiënten die betrokken zijn bij de vorm van het model in die gebieden. Deze bijzondere mogelijkheid kan een belangrijke rol gaan spelen in toekomstige adaptieve modelgebaseerde besturingssystemen. In een dergelijk systeem kan een geïntegreerd simplex spline model lokaal en in *real-time* worden aangepast om veranderingen in de systeemdynamica te op te vangen.

Het benaderingsvermogen van de multivariate simplex splines kan verhoogd worden door een willekeurig aantal simplices samen te voegen in een geometrische structuur. Dit wordt een triangulatie genoemd. Triangulaties komen in vele vormen en maten voor, variërend van configuraties die slechts uit twee simplices bestaan tot configuraties die miljoenen simplices bevatten. Triangulaties kunnen geoptimaliseerd worden door lokaal de resolutie van simplices te verlagen of te verhogen om de lokale complexiteit van het systeem optimaal te kunnen benaderen. In principe wordt het totaal aantal simplices binnen een triangulatie alleen beperkt door de beschikbare rekenmiddelen. In dit proefschrift wordt echter geconstateerd dat er een belangrijke praktische grens is voor de grootte en resolutie van een triangulatie. Deze praktische grens is het gevolg van het feit dat iedere simplex een minimum aan datacontent nodig heeft, welke wordt bepaald door de orde en de continuïteit van de basispolynomen. In dit proefschrift wordt aangetoond dat dit probleem van datadekking een nieuwe aanpak van triangulatie optimalisatie noodzakelijk maakt, aangezien de methodes in de bestaande literatuur de per-simplex datadekking niet beschouwen als een parameter voor optimalisatie. Deze nieuwe voorgestelde methode voor de optimalisatie van de triangulatie produceert triangulaties die speciaal geschikt zijn voor de simplex splines, door ervoor te zorgen dat iedere simplex in een triangulatie een minimum aantal gegevens bevat.

Hoewel multivariate simplex splines in het verleden zijn gebruikt voor het modelleren van verstrooide non-lineaire data in twee en drie dimensies, was er geen schema voor systeemidentificatie beschikbaar op basis van de simplex splines. De unieke eigenschappen van de simplex splines, samen met de hierboven genoemde voordelen ten opzichte van bestaande data approximatoren, maakt ze zeer wenselijk voor gebruik in een dergelijk schema. De belangrijkste doelstelling van dit proefschrift is om een nieuw schema te definiëren voor systeemidentificatie, welke gebaseerd is op de multivariate simplex splines. Deze nieuwe methode omvat de drie hoofdaspecten van systeemidentificatie: modelstructuurselectie, parameterschatting en modelvalidatie. Het aspect van modelstructuurselectie voor de multivariate simplex splines bestaat uit twee delen. Het eerste deel is de geometrische modelstructuurselectie, die bestaat uit de selectie van de dimensies van het spline model en de definitie van een triangulatie in deze set dimensies. Het tweede deel bestaat uit de definiëring van de polynome modelstructuur. Met betrekking tot het aspect van parameterschatting is een nieuwe formulering van het standaard lineaire regressiemodel ontwikkeld. In deze formulering vormen de B-polynomen van de simplex splines de regressoren. Door deze nieuwe formulering kunnen een aantal verschillende technieken van parameterschatting worden toegepast om de B-coëfficiënten van de B-polynomen te schatten. In dit proefschrift worden twee nieuwe methodes voor parameterschatting geïntroduceerd. De eerste methode is een gegeneraliseerde kleinste kwadratenschatter, waarmee B-coëfficiënten op simplices kunnen worden geschat die meetruis bevatten van wisselende magnitude. De tweede methode is een recursieve kleinste kwadratenschatter met differentile randvoorwaardes waarmee in *real-time* de spline modellen gemodificeerd kunnen worden aan de hand van nieuw binnenkomende observaties. Voor het aspect van modelvalidatie werd de kwaliteit van de geschatte spline modellen bepaald door bestaande methodes te gebruiken die gebaseerd zijn op een analyse van modelresiduen en parameter varianties. Daarnaast werd er een aantal geheel nieuwe methodes voor kwaliteitsbepaling mogelijk gemaakt door gebruik te maken van B-polynomen. Zo kunnen bijvoorbeeld de statistische varianties van de B-coëfficiënten worden gelokaliseerd tot specifieke locaties binnen het model. Dit betekent dat gebieden met een hoge parameter variantie geïsoleerd kunnen worden binnen het totale model en vervolgens nader geanalyseerd. Deze unieke en krachtige eigenschappen kunnen nieuwe inzichten geven in systeemidentificatie en parameterschatting, hetgeen mogelijk kan leiden tot nieuwe ontwikkelingen op dit gebied.

Dit proefschrift introduceert drie belangrijke innovaties op het gebied van de multivariate spline theory. Deze innovaties zijn noodzakelijk om een effectieve methode te creëren voor systeemidentificatie met simplex splines. De eerste vernieuwing is de definitie van de differentiële randvoorwaardes, die gebruikt worden om de richtingsafgeleide van de simplex splines te beperken op geselecteerde locaties binnen het spline-domein. De differentiële randvoorwaardes maken begrensde modelextrapolatie mogelijk en beperken de divergentie van polynomen op de grens van het spline-domein. Daarnaast kunnen de differentiële randvoorwaardes toegepast worden om bijvoorbeeld Dirichlet- of Cauchy-randvoorwaarden op te stellen, waardoor het mogelijk wordt om oplossingen van randwaardeproblemen te benaderen met simplex splines. De tweede vernieuwing is de ontwikkeling van een theorie voor de kwantificatie van B-net propagatie, een nieuw effect dat werd waargenomen in triangulaties op grote schaal. B-net propagatie is de verspreiding van lokale verstoringen van het B-net van één simplex naar dat van zijn buren. Er wordt bewezen dat B-net propagatie een simplex spline-functie effectief van een lokale approximator in een globale approximator transformeert als de continuïteit van de spline-functie hoog is ten opzichte van zijn polynoomorde en wanneer deze functie gedefinieerd is op het meest gebruikte type triangulatie. De derde innovatie is een nieuwe formulering van de B-polynomen in globale Cartesische coördinaten in plaats van lokale barycentrische coördinaten. De Bernstein basispolynomen van de simplex splines zijn functies in termen van lokale barycentrische coördinaten. Dit houdt in dat hun globale interpretatie betekenisloos is. De nieuwe formulering van de B-polynomen in globale coördinaten maakt een globale interpretatie mogelijk. Bovendien maakt de nieuwe formulering het mogelijk om triangulaties en B-coëfficiënten in één stap te optimaliseren, waardoor er geen noodzaak meer bestaat voor afzonderlijke triangulatie optimalisatie.

Vliegtuigaerodynamica staart bekend om zijn niet lineaire aard. Dit heeft ten gevolge gehad dat dat de identificatie van nauwkeurige aerodynamische modellen op basis van vluchtdata een uiterst uitdagende taak is gebleken. Aerodynamische modellen zijn van cruciaal belang in de correcte werking van vluchtsimulators en vluchtbesturingssystemen. Hoe hoger de kwaliteit is van een aerodynamisch model, hoe nauwkeuriger de voorspelling van de werkelijke aerodynamische krachten en momenten die op een vliegtuig werken. Voor vluchtsimulatortoepassingen vertaalt dit zich direct in een hogere waarheidsgetrouwheid van de simulatie, met als gevolg een betere trainingsomgeving voor piloten. Voor vluchtbesturingssystemen leiden aerodynamische modellen van hogere kwaliteit tot een betere bestuurbaarheid van het vliegtuig, ook na het optreden van structurele schade. Het uiteindelijke gevolg van het verhogen van de kwaliteit van aerodynamische modellen is het verbeteren van de vliegveiligheid. Deze maatschappelijke relevantie, tezamen met de technische uitdaging van het identificeren van nauwkeurige aerodynamische modellen, biedt de ideale argumenten voor het demonstreren van de nieuwe methode.

Met de nieuwe methode zijn twee identificatie-experimenten uitgevoerd op het gebied van aerodynamische modelidentificatie. Het eerste experiment betrof de identificatie van een aerodynamisch model voor een F-16 gevechtsvliegtuig op basis van een NASA windtunnel model. De interne structuur van dit windtunnel model was bekend en bood daardoor een gecontroleerde omgeving om de nieuwe methode te testen en te valideren.

Bij het tweede identificatie-experiment werd een complete set aerodynamische modellen geïdentificeerd voor het Cessna Citation II laboratorium vliegtuig met behulp van vluchtdata die verkregen werd uit zeven testvluchten die werden uitgevoerd tussen 2006 en 2010. In totaal werden 247 testvluchtmanoeuvres uitgevoerd die een significante dekking boden van de vluchtenveloppe van de Citation II. De volledige dataset voor identificatie bestond uit miljoenen metingen van meer dan zestig vluchtparameters. Voor dit experiment werden de modelstructuurselectie, parameterschatting en modelvalidatie aspecten van de systeemidentificatie behandeld. De geometrische modelstructuurselectie werd uitgevoerd door gebruik te maken van een nieuwe aanpak die gebaseerd is op de aanwezigheid van hysterese in de aerodynamische kracht- en momentcoëfficiënten. Door de hysterese-analysemethode te gebruiken, werden een aantal mogelijke dimensiesets gedefinieerd. Voor iedere mogelijke dimensieset werd een triangulatie van de hyperkubus gecreëerd die de data van de vluchttest omvatte. De polynome modelstructuur werd geselecteerd door de prestaties te vergelijken van een aantal prototype simplex spline-functies van verschillende polynoom- en continuïteit ordes op de hyperkubus triangulatie. Meer dan 2000 prototype spline-modellen werden geïdentificeerd door gebruik te maken van een geoptimaliseerde software-implementatie van het simplex spline identificatiealgoritme. De uiteindelijke geometrische en polynoom modelstructuren werden geselecteerd op basis van een verdere optimalisatie van het best presterende prototypemodel.

De geïdentificeerde aerodynamische modellen zijn fenomenologische modellen, ofwel modellen die direct op observatie gebaseerd zijn. De validatie van de spline modellen wees uit dat de modellen nauwkeurig zijn en ook een gegarandeerde numerieke stabiliteit hebben binnen het spline-domein. De identificatie- en validatieresultaten van de simplex spline modellen werden vergeleken met die van gangbare polynoom modellen, die geïdentificeerd werden met standaard systeem identificatie methodes. Deze resultaten toonden aan dat de aerodynamische modellen die gebaseerd zijn op multivariate simplex splines van een significant hogere kwaliteit waren dan de aerodynamische modellen die gebaseerd waren op gangbare polynomen.

Het onderzoek dat werd uitgevoerd in het kader van dit proefschrift heeft tot drie hoofdaanbevelingen geleid. Ten eerste werd vastgesteld dat de belangrijkste praktische beperking van de toepassing van multivariate simplex splines voor systeem identificatie de per-simplex datadekking is. Om het praktische nut van simplex splines te verhogen, zou er een software-instrument ontwikkeld moeten worden om, in *real-time*, de dekking met data van de triangulatie te analyseren. In het geval van de identificatie van aerodynamische modellen zou een dergelijk software-instrument signalen doorsturen naar de piloten om specifieke manoeuvres uit te voeren. De tweede aanbeveling is dat een algemene optimalisatiemethode voor triangulaties ontwikkeld zou moeten worden die speciaal geschikt is voor systeemidentificatie met simplex splines. Een dergelijke methode zou gebaseerd kunnen zijn op de globale formulering van de B-polynomen die in dit proefschrift gepresenteerd wordt. Dit zou een belangrijk hiaat kunnen dichten in de huidige simplex spline-theorie. De derde aanbeveling is een *real-life* implementatie van een besturingssysteem dat is gebaseerd op een adaptief model waarbij de recursieve B-coëfficiënt schatter wordt toepast die in dit proefschrift is geïntroduceerd. Voor toepassingen in de lucht- en ruimtevaart zou dit resulteren in een fouttolerant vluchtbesturingssysteem met een ingebouwde voorspellingsfunctionaliteit voor de vluchtenveloppe. Een dergelijk besturingssysteem, wanneer geïnstalleerd in een toekomstig vliegtuig, zou de vliegveiligheid verhogen door potentieel fatale gebeurtenissen om te zetten in overleefbare incidenten.

Acknowledgments

This thesis is the result of a voyage of scientific adventure that has lasted almost four years. As with any voyage, there were moments of setback and moments of great reward. I am grateful to the following people for supporting me during moments of setback, and celebrating with me the moments of reward.

First of all, I would like to thank my promoter prof. Bob Mulder for his strategic guidance, and his uncanny ability to make single remarks that materialize into scientific journal papers. Bob, thank you for giving me the opportunity and the freedom to pursue my PhD project, and for letting me realize that lecturing is a true art.

Also, I would like to thank my co-promoter dr. Ping Chu for his scientific guidance and insights that lead to many innovations during my research project.

I would like to express my sincere gratitude to prof. Ming-Jun Lai of the Department of Mathematics of the University of Georgia, U.S.A., for supporting the research in this thesis.

My gratitude also goes out to prof. Max Mulder for recruiting me into the department of Control and Simulation, and for giving me the opportunity to perform an internship at EADS during which the foundation for the research in this thesis was laid.

I would like to thank dr. Erol Özger and dr. Cristophe Oelker for the great times that I experienced at the department of flight testing of EADS in Manching, Germany.

My thanks go out to Elwin de Weerdt. I greatly appreciated our many discussions, brainstorming sessions, and the filling in of patent forms. I am looking forward to continuing our cooperation, in the certain knowledge that the “billions” are steadily marching towards us.

I would also like to thank Erik-Jan van Kampen for our scientific cooperation, but also for the great times we had abroad, such as that one conference in São Paulo, Brazil, where we presented our work in an amphitheater to an audience of professors. I am looking forward to our continued scientific collaboration.

My thanks go to Daan Pool and Peter Zaal, for giving me the opportunity to conduct my sometimes nauseating flight test experiments during their experiment flights.

I would like to thank my squash buddies Joost, Erik-Jan, Peter, and Paul for the many hours of grueling but refreshing exercise on the squash court. It is said that the combination of a healthy mind in a healthy body produces happiness. As for the first part, one cannot pretend that the mind is still healthy after a year of full-time writing, but the squash sure helped with balancing the whole equation.

My thanks also goes out to all the colleagues at the department of control and simulation; Olaf Stroosma, Joost Ellerbroek, Laurens van Eykeren, Wouter Falkena, Jan Comans, Liguo Sun, Bruno Correia Gracio, Deniz Yilmaz, Daan Pool, Rita Valente Pais, Herman Damveld, Paul de Jong, Arjen de Leege, Mark Mulder, Xander in 't Veld, Herman Koolstra, Erik-Jan van Kampen, Hui Yu, Ferdinand Postema, Andries Muis, Clark Borst, Hans Mulder, Thomas Lombaerts, Elwin de Weerdt, Lars Sonneveldt, Joost Venrooij, Peter Zaal, and Eddy van Oort. I would like to thank you for all the high- and low quality times in the coffee corner, around the poker table, on the basketball court, and in places far beyond. And I would also like to thank Bertine Markus, the driving force behind Control & Simulation.

My gratitude goes out to my friends in Bolivia, Ernesto Gomez and Zarina Mendez. I greatly enjoyed our discussions, and are grateful for your patience in trying to teach me Spanish. You helped me gain perspective on aspects of opportunity and freedom that we either take for granted or have lost altogether in our parts of the world.

I would also like to thank my longtime friends Boudewijn, Christoph, Roderique, Alwin, Marijn, Jitske, Ivette, and Iris, for providing both fresh perspectives on my research as well as some much needed social distraction.

My thanks also go to Coen Willemse first for reading my thesis and providing feedback, but also for our always inspirational discussions on all fields of science and technology.

Then, I would like to thank my new family, André, Mieke, Harmen, Anne, Jolien and Sander for your kind support, interest and feedback during these years.

I would like to thank my sisters Marijn and Sanne, and my new brothers Peter and Bernd, for their support, care and interest in my professional and private life. Marijn, thank you for your professional translation of the summary into Dutch!

I also want to express my gratitude towards my parents, Ries and Liedewey. Thank you for your unconditional support, constructive feedback, and care, and for setting the boundary conditions that eventually resulted in this thesis. Our endless discussions on any topic have been, and always will be, a great source of inspiration and guidance for me.

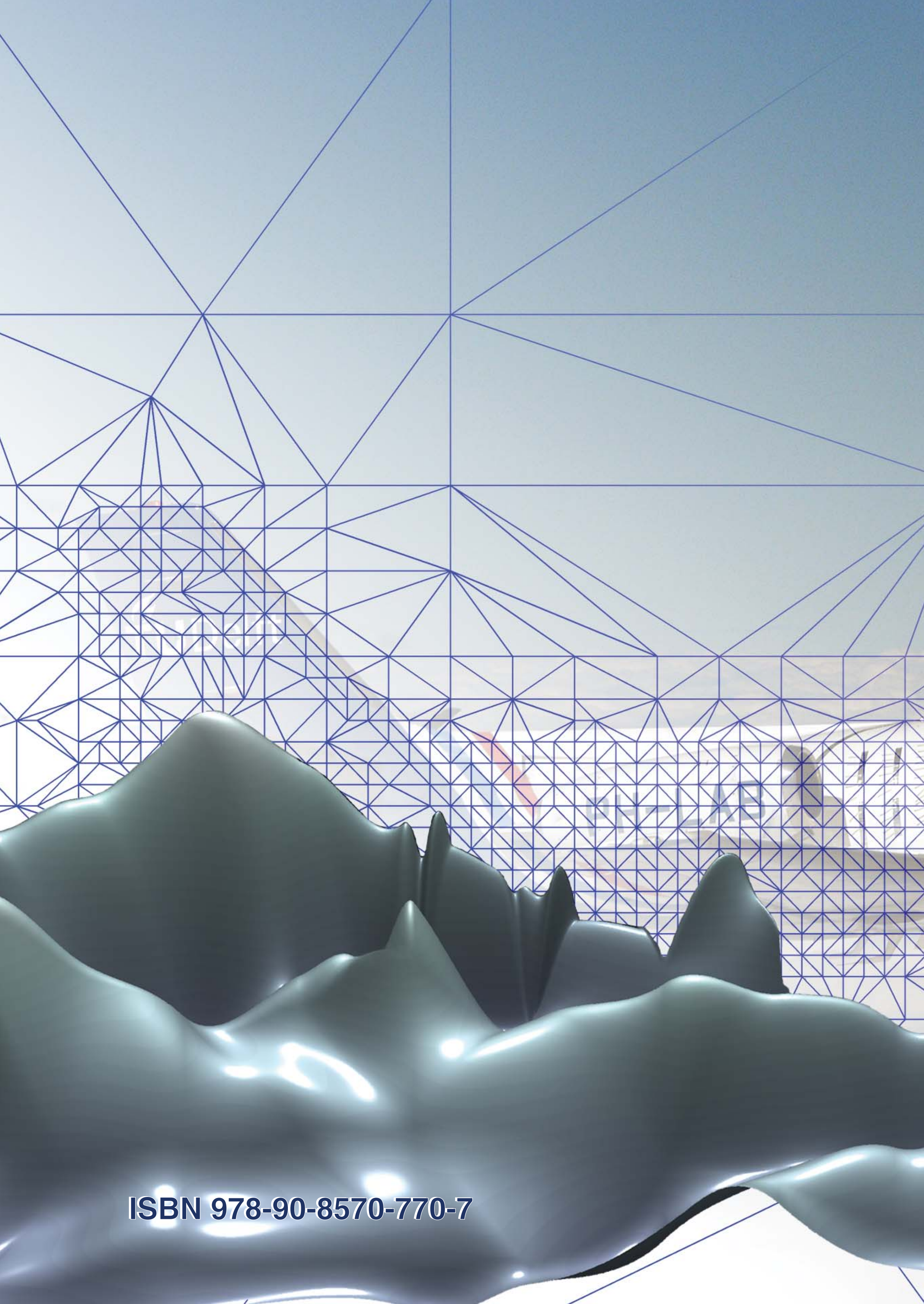
And finally, Rinske, my dearest friend, my most trusted companion, my beloved wife. Without you I would not have been able to finish this thesis. Your unwavering support and ever present energy, love, enthusiasm, optimism, and determination pulled me through. I thank you for all our big adventures, and precious little moments together. Together with you I see a bright future in which nothing is impossible.

Curriculum Vitae

Coen de Visser was born in Groningen, The Netherlands on December 30^{th}, 1979. From 1992 to 1998 he attended Het Wagenings Lyceum in Wageningen, The Netherlands.

In 1998 enrolled as a student at the Faculty of Aerospace Engineering of the Delft University of Technology. From 2001 till 2005 he was employed as a software engineer, first at Floating Point Software, later at Foxnet BV. In June 2005 he performed a six month internship at EADS Military Aircraft (now Cassidian) in Manching, Germany, where he developed a software tool for visualizing aerodynamic data tables of Eurofighter Typhoon. In August 2006, after obtaining his B.Sc. degree, he started his M.Sc. graduation project in cooperation with EADS Military Aircraft, with the topic of developing a multivariate spline method for aerodynamic model validation. After finishing the primary project at EADS in January 2007, he obtained an EADS scholarship that allowed him to continue his research until April 2007. In August 2007 he obtained his M.Sc. degree for his research project 'A Multivariate spline Method for Aerodynamic Model Validation' which he applied to validate modifications to the aerodynamic model of Eurofighter Typhoon.

In September 2007 he started as a Ph.D. student at the department of Control & Operations of the Faculty of Aerospace Engineering of the Delft University of Technology. His research interests are system identification, multivariate spline theory, system theory, flight dynamics, aerodynamics, flight envelope prediction, and fault tolerant control.



ISBN 978-90-8570-770-7