Faculty of Aerospace Engineering

Full Flight Envelope Aerodynamic Modelling of the Cessna Citation II using Physical Splines

F.J.A. Huisman

5 December 2017

TUDelft

CITATION STALL MODELING TUDelft C&S

Full Flight Envelope Aerodynamic Modelling of the Cessna Citation II using Physical Splines

Master of Science Thesis

For obtaining the degree of Master of Science in Aerospace Engineering
at Delft University of Technology

F.J.A. Huisman

5 December 2017

Faculty of Aerospace Engineering · Delft University of Technology

Delft University of Technology

Delft University Of Technology
Department Of
Control and Simulation

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled **"Full Flight Envelope Aerodynamic Modelling of the Cessna Citation II using Physical Splines"** by **F.J.A. Huisman** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 5 December 2017

Readers:

dr.ir. Q. P. Chu

dr.ir. C. C. de Visser

Summary

A new aviation legislation makes it mandatory for air-carrier pilots to go through stall recovery training on simulators. New aerodynamic models are required and because of complex non-linear stall dynamics also more advanced system identification techniques are needed. Therefore a Control and Simulation Department task-force was set-up. In recent years multiple non-linear system identification techniques are researched but none has proven itself to be a true solution. In 2005 a promising true general splines method, the Multivariate Simplex B-Splines was developed. In combination with the recent physical-splines transformation a major shortcoming is solved. The transformation makes it possible to physically interpreted the aerodynamic model and the model structure. In this research project the properties of the new physical-splines are explored. Subsequently the aerodynamic modelling process is updated and the physical-splines formulation is added to this process forming a new toolbox.

The aircraft to be modelled is the Cessna Citation II. Experimental flights are performed, with the PH-LAB the Cessna Citation II co-owned by the Delft University of Technology, creating a flight-test dataset with measurements. The flight-test data is filtered and the flight path is reconstructed. The flight-test dataset will be used to create a full-flight envelope aerodynamic model of the Cessna Citation II.

The process of creating an aerodynamic model using the Multivariate Simplex B-Splines method will be divided in four parts: data selection, the geometric model structure, polynomial model structure selection and the parameter estimation. Creating a geometric model structure is not a trivial process. The amount and distribution of the flight-test will have influence on the outcome and high dimensional spaces are quite large and empty. Also, certain parts of the triangulation domain can require higher approximation power an continuity. The consequence is an overall dense triangulation and high order basis functions. This introduces problems such as over fitting the model and divergent behaviour on triangulation boundaries. The triangulation is created using multi-dimensional hyperrectangles containing simplices.
The polynomial model structure selection is performed in physical space using the Barycentric-Cartesian space transformation. The importance of model terms is expressed in the significance they have in reducing the model error. Using a forward orthogonal stepwise regression method their error reduction ratio can be calculated. Different (stopping) rules can be considered for including or excluding terms in the model while decreasing model storage size and prevent overfitting the data. Physical-Splines make use of a linear transformation that transforms from the barycentric coordinate space to the Cartesian coordinate space, giving the

MSBS a physical interpretation. The physical transformation is introduced to the optimization process in the form of equality and inequality constraints. This way a-priori aerodynamic information can form a bound on the aerodynamic stability derivatives. Promising results show that they are robust, prevent over-fitting, prevent propagation of erroneous data, remove divergent behavior on triangulation boundaries, and that they can be used for extrapolation of sparse datasets.
Finally the model parameters are estimated using a optimisation algorithm. Important for not breaking the geometric model structure is that terms cannot be excluded from regression matrix used during the parameter optimisation process. Hence the optimisation is subjected to constraints. Constraints are subdivided in equality, inequality and/ or bounded constraints. The estimation can be performed using the well known linear ordinary least squares method or the generalised least squares method. Inequality and bounded constraints require iterative methods which can be solved in polynomial time. Overall, the physical constraints make it possible to adjust model approximation power locally and alter the B-net via the physical parameters without breaking them.
This process will be an iterative process and multiple models are created and compared. Their quality will be assessed based on the outcomes of a raw validation by comparing the model output and the measured output using the validation set, a global model inspection of the physical parameters, a model residuals analysis, stability analysis and statistical analysis.

The outcome of this thesis, the new physical-splines formulation, is used to create and aerodynamic model. This model is compared to the current base-line model. In the future the aerodynamic models created, using the physical-splines formulation, can be used on simulators that will train air-carrier pilots in stall recovery. This will result in an increased safety in aviation.

Research question, aims and objectives

In this MSc research project the new formulation of multivariate Physical-Splines is researched and implemented in order to create an aerodynamic model of the full flight envelope of the Cessna Citation II. The recent results of the task force will be used and the new model will be compared with the current baseline-local-linear aerodynamic model. This section introduces the research question, including the sub-questions. Using these questions the project aim is elaborated and the objectives of the project are given. Finally the motivation and feasibility will substantiate the project aim and objectives.

Research Question

The research question of the project is formulated as follows:

How can the Physical-Splines formulation be used to create an aerodynamic model of the Cessna Citation II aircraft?

The research question is supported by the following sub-questions:

1. *How does the new Physical-Splines formulation differ from the Multivariate Simplex B-Splines formulation and what does this mean for the system identification process and the aerodynamic model?*
2. *What improvements can be made to the current non-linear aerodynamic model (DASMAT) and how can the Physical-Splines contribute to this?*
3. *Which geometric and polynomial model structure should be used for aerodynamic modelling and what is the dimension and degree of this model structure and the size of the triangulation?*
4. *What flight-test data is available and how can this data be used to create the Physical-Splines model?*

Project Aim

The project aim is to implement a full-flight envelope aerodynamic model of the Cessna Citation II in the simulator framework using physical-splines. This physical-splines model

will be continues over the full flight-envelope and can be physically interpreted. The stall and post-stall dynamics in the flight envelope can be used on (full-motion) flight simulators to train pilots on stall recovery according to the new aviation legislation. The continuity of the splines makes the aerodynamic model more appropriate for developing (adaptive) model based control allocation methods. Overall the physical-splines aerodynamic model will increase the fidelity of the current baseline aerodynamic model of the Cessna Citation II in terms of approximation power, applicability, continuity, physical interpretation and computational efficiency.

Research Objectives

The main goal of this research project is projected in the objectives listed below. The objective states the main goal of this research project and is subdivided into multiple sub-objectives explaining how this objective will be achieved.

- Create a Physical-Splines formulation of the full-flight envelope aerodynamic model of the Cessna Citation II aircraft, using flight-test data by
    1. comparing the new physical-splines formulation with the Multivariate Simplex B-Splines formulation and explore new possibilities,
    2. creating a new methodology which integrates the physical-splines into the Multivariate Simplex B-Splines aerodynamic modelling process including other newly obtained possibilities,
    3. measuring the performance of physical-splines in terms of model output, computational efficiency, numerical stability, interpretation and continuity, compared to current base-line model.

Motivation and Feasibility

Full flight envelope modelling became more popular in recent years. Non-linear aircraft dynamics and full flight envelope models (including stall and post stall dynamics) require more complex modelling techniques. The increase of computational power makes it possible to process larger datasets and increase model complexity. Different modelling techniques are researched in the previous years, all having certain advantages and disadvantages. None of these methods distinguish itself featuring all advantages. A promising method was introduced in 2005, a general spline formulation called the Multivariate Simplex B-Splines (MSBS). In combination with the physical-splines transformation it is a general function approximator that can fit any scattered non-linear dataset. It is compatible with linear regression methods and the final model coefficients can be physically interpreted. In the previous years many research is performed on the MSBS by the C&S department. This research will continue with the new physical-splines formulation and solve a major disadvantage of the MSBS. The C&S department has an unique expertise in this developing field. Furthermore measurements from the experimental flights are generated by the PH-LAB aircraft co-owned by the TU-Delft. This and the unique combination with the SIMONA Research Simulator (SRS) makes it feasible to create and test an aerodynamic model of the Cessna Citation II. The new high fidelity model can be used to train air-carrier pilots in stall recovery. This contribution to aviation will increase the safety.

Acronyms

AHRS	Attitude and Heading Reference System
c.m.	Centre of Mass
C&S	Control & Simulation
CAD	Computer-Aided Design
DASMAT	Delft University Aircraft Simulation Model and Analysis Tool
DEM	Digital Elevation Model
DUT	Delft University of Technology
ECEF	Earth-Centered, Earth-Fixed
EOM	Equations of Motion
ERR	Total Error Reduction Ratio
err	Individual term Error Reduction Ratio
FMS	Flight Management System
GLS	Generalised Least Squares
HDF5	Hierarchical Data Format
IMU	Inertial Measurement Unit
KKT	Karush-Kuhn-Tucker
MSBS	Multivariate Simplex B-Splines
MSE	Mean Squared Error
NRL	Netherlands Aerospace Centre
OLS	Ordinary Least Squares
P-Splines	Physical-Splines
PDE	Partial Differential Equations
PRESS	Predicted Sum of Squares
PSE	Predicted Square Error
R-MSE	Reduction of Mean Squared Error
RMS	Root Mean Squared Error
SDP	Semidefinite Programming

SRS	SIMONA Research Simulator
SRTM	Shuttle Radar Topographic Mission
TR-MSE	Total Reduction of Mean Squared Error
TSE	Total Squared Error
UKF	Unscented Kalman Filter
VIF	Variance Inflation Factor

List of Symbols

Greek Symbols

α	Angle of Attack
β	Angle of Sideslip
δ	Control surface deflection
γ	Flightpath angle
κ	Multi-index
$\mathbf{\Lambda}$	Barycentric to Cartesian space transformation matrix
λ	bias term
ν	Simplex vertex coordinate in Cartesian space
ρ	Air density
$\mathbf{\Sigma}$	Covariance matrix
σ	Standard Deviation
σ^2	Variance
τ	Grid density
$\boldsymbol{\theta}$	Parameter vector

Roman Symbols

$\mathbf{A}$	Cartesian to barycentric space coordinate transformation matrix
a	Aileron
b	Barycentric coordinate
b	Wingspan
$\mathbf{B}$	Global data location B-matrix

$\bar{c}$	Mean aerodynamic chord
$\mathbf{c}$	Global B-coefficient vector
c_κ	Individual B-coefficient
C_l	Roll Moment Coefficient
C_m	Pitching Moment Coefficient
C_n	Yaw Moment Coefficient
C_X	Longitudinal Force Coefficient
C_Y	Lateral Force Coefficient
C_Z	Vertical Force Coefficient
d	Spline polynomial degree
e	Elevator
$\mathbf{err}$	Error Reduction Vector
$\mathbf{G}$	Constraint matrix in Cartesian space
$\mathbf{g}$	Orthogonal coefficient vector
g	Gravity constant
$\mathbf{H}$	Smoothness/ Continuity Matrix
h	Altitude
$\mathbb{I}$	Inertia Matrix
i, j	General indexers
J	Cost function
k, l	General indexers
L	Combined aerodynamic and thrust moments about the body X-axis
M	Number of measurements
M	Combined aerodynamic and thrust moments about the body Y-axis
m	Mass
N	Combined aerodynamic and thrust moments about the body Z-axis
n	Spline dimension
p	Roll Rate around the body X-axis
$p(x)$	General polynomial in Cartesian coordinates
q	Pitch Rate around the body Y-axis
q, r	General indexers
r	Spline continuity order
r	Yaw Rate around the body Z-axis

r	Rudder
S	Wing Area
$\mathcal{S}_d^r$	Spline space of degree d and continuity r
$\mathcal{T}$	Triangulation formed by a set of simplices
t_j	Single simplex j
u	Lateral airspeed velocity along body X-axis
V	Airspeed
v	Longitudinal airspeed velocity along body Y-axis
$\mathbf{W}$	Orthogonal regression matrix
w	Vertical airspeed velocity along body Z-axis
$\mathbf{X}$	Regression Matrix
x	Position coordinates along the X-axis
X	Combined aerodynamic and thrust forces about the body X-axis
$\mathcal{X}$	Set of points in Cartesian space
$\mathbf{y}$	Measurement vector
y	Position coordinates along the Y-axis
Y	Combined aerodynamic and thrust forces about the body Y-axis
z	Position coordinates along the Z-axis
Z	Combined aerodynamic and thrust forces about the body Z-axis

Contents

Part I

Scientific Paper

Full Flight Envelope Aerodynamic Modeling of the Cessna Citation II using Physical-Splines

F.J.A. Huisman* and C.C. de Visser†
Delft University of Technology, P.O. Box 5058, 2600GB Delft, The Netherlands

A new aviation legislation makes it mandatory for air-carrier pilots to go through stall recovery training on simulators. As a result, new aerodynamic modeling techniques are required to model complex non-linear behavior of the aircraft flight envelope. In 2005, the Multivariate Simplex B-Splines method was developed. MSBS are a true general function approximator and are easily integrated in standard identification routines. Their downside is that the basis functions and B-coefficients, forming the B-net, do not have a straightforward physical interpretation. Also creating the triangulation is not a trivial process. Parts of the triangulation domain can require higher approximation power and continuity. The consequence is an overall dense triangulation and high order basis functions. This introduces problems such as over fitting the model and divergent behavior on triangulation boundaries. Physical-Splines make use of a linear transformation that transforms from the barycentric coordinate space to the Cartesian coordinate space, giving the MSBS a physical interpretation. The physical transformation is introduced to the optimization process in the form of equality and inequality constraints. This way a-priori aerodynamic information can form a bound on the stability derivatives. Promising results show that they are robust, prevent over-fitting, prevent propagation of erroneous data, remove divergent behavior on triangulation boundaries, and that they can be used for extrapolation of sparse datasets. Also a stepwise orthonormalization can create physical model structure constraints and set unimportant physical model terms to zero. Overall, the physical constraints make it possible to adjust model approximation power locally and alter the B-net via the physical parameters without breaking them.

Nomenclature

A_x, A_y, A_z	Specific forces around the body X,Y,Z axis, m/s^2
b	Barycentric coordinate
$\mathbf{B}$	Global data location B-matrix
C	General dimensionless coefficient with aerodynamic forces X, Y, Z and moment l, m, n subscripts
$\bar{c}$,b	Mean aerodynamic chord and wingspan, m
$\mathbf{c}_b$	Global B-coefficient vector
c_κ	Individual B-coefficient
$\mathbf{c}_p$	Global physical coefficient vector
d	Spline polynomial degree
$\hat{d}$	Total number of polynomial terms in basis function
err	Error Reduction Ratio
$F_d^{n,d}$	Kronecker substituted convolution function expanding multinomial polynomial of degree d, dimension n
$\mathbf{F}$	Kronecker Substituted convolution matrix
g	gravity constant, m/s^2
$\mathbf{G}, \mathbf{g}_e$	Cartesian equality constraint matrix and vector
$\mathbf{G}_{l,u}, \mathbf{g}_{l,u}$	Upper and lower bound Cartesian inequality constraint matrix and vector
$\mathbf{H}_e, \mathbf{h}_e$	Barycentric equality constraints matrix and vector
$\mathbf{H}_i, \mathbf{h}_i$	Barycentric inequality constraints matrix and vector
$I_{xx}, I_{yy}, I_{zz}, I_{xz}$	Moment of inertia around the body and cross body axis, $kg \cdot m^2$

*M.Sc. student, Department of Control and Simulation, Faculty of Aerospace Engineering.
†Assistant Professor, Department of Control and Simulation, Faculty of Aerospace Engineering.

i	Complex number
j, k	General indexers
M	Mach number
L, M, N	Combined aerodynamic and thrust moments around the body X,Y,Z axis, $n \cdot m$
m	mass, kg
n	Spline model dimension
$p(b), p(x)$	Polynomial function with barycentric and Cartesian coordinates respectively
p, q, r	roll, pitch and yaw rates around the body X,Y,Z axis, rad/s
r	Spline model continuity
S_d^r	Spline space of degree d and continuity r
$\mathcal{T}$	Triangulation formed by a set of simplices
$\mathbf{T}^1$	Cartesian to barycentric coordinate transformation matrix
t_j	Individual simplex j
V	Airspeed, m/s
$\mathbf{X}_b$	Global data location Barycentric B-matrix
$\mathbf{X}_p$	Global data location Physical B-matrix
X, Y, Z	Combined aerodynamic and thrust forces around the body X,Y,Z axis, F
Symbols	
α, β, γ	Angle of attack, angle of sideslip, and flightpath angle, rad
δ	Control surface deflection with a, e, r aileron, elevator and rudder subscripts, rad
ϵ	Residual vector
$\kappa, \boldsymbol{\kappa}$	Multi-index and multi-index matrix
λ	Bias term
$\mathbf{\Lambda}$	Barycentric to Cartesian coordinate transformation matrix
ρ	Air density, kg/m^3
ν	Cartesian coordinate of simplex vertex

I. Introduction

From 2019, a new aviation legislation makes all air-carrier pilots obliged to go through flight simulator-based stall recovery training [1]. A 'task force' was formed by the division of the Control & Simulation (C&S) group as result of this new aviation legislation. The implication of this legislation is that current flight simulation models must be updated regarding accurate pre-stall, stall, and post-stall dynamics. Recently, a new methodology is derived for high-fidelity aircraft stall behavior modeling and simulation, and a new high-fidelity flight simulation model of the Cessna Citation II laboratory aircraft has been developed. This model is made up of sets of local linear models which must be interpolated in order to adequately cover the flight envelope. The current Delft University Aircraft Simulation Model and Analysis Tool (DASMAT), [2], model will eventually be replaced by this model.

Performing this interpolation is not trivial. For example, naive database interpolation can result in significant discontinuities and other artifacts which may influence simulator fidelity. A promising alternative approach uses Multivariate Simplex B-Splines (MSBS) to accurately model non-linear aerodynamic models, [3] and [4],over the entire flight envelope and are easily integrated into standard identification routines. While the multivariate B-splines are powerful function approximators, their downside has been that their basis functions and B-coefficients do not have a straightforward physical interpretation due to the use of a local coordinate system.

Very recently, a new formulation of the MSBS was developed. These physical B-splines or Physical-Splines are a modification of standard multivariate B-splines in which the B-coefficients have a physical interpretation, i.e. they are transformed from barycentric coordinate space into Cartesian space under a linear coordinate transformation. This has removed the primary disadvantage of the MSBS method, and will allow its much more widespread use as a tool in aircraft system identification.

In this paper the Physical-Splines are elaborated and new possibilities and methods are researched. These methods are thereafter implemented in the aerodynamic modeling process. This paper concludes with an experiment, where flight test data is used to create a Physical-Splines moment C_m model. The experimental outcome measures the performance of the Physical-Splines and delivers a comparison to the MSBS.

II. Multivariate Simplex B-Splines

The MSBS is a general function approximator. It can fit any scattered non-linear dataset, is compatible with linear regression methods and has local basis functions. Furthermore it requires a geometric support structure called a triangulation. A disadvantage is that the B-coefficients, to be estimated, do not have a straightforward physical meaning. In this section the theory of the Multivariate Simplex B-spline is explained briefly by introducing simplices, barycentric coordinates and the barycentric Bernstein basis polynomials.

A. Simplex

A hypercube triangulation is created using simplices. A simplex is a generalization of a point, line, triangle, tetrahedron, etc. An n-dimensional n-simplex will have $n + 1$ vertices. The vertices are connected to each other via so called edge facets. Each simplex forms a convex hull with $n + 1$ independent vertices. Furthermore, a n-simplex is affinely independent $\in \mathbb{R}^n$. A single simplex t_j with $n + 1$ vertices ν is defined in Eq. (1)

$$t_j := \langle \boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_n \rangle \tag{1}$$

This property makes the simplex very popular tool in graphical computer animations and modeling. The number of n-facets in a simplex can be determined using Pascal's triangle making the simplex creation recursively. Aligning the $n - 1$ facets of adjacent simplices will create a triangulation. A Kuhn triangulation, [5], [6], generalizes the creation of the triangulation by adding multiple n-simplices together and aligning their facets. This type-1 triangulation forms a n-dimensional, n-hypercube of J non-overlapping simplices t_j, partitioning a certain domain. Multiple n-hypercubes can be aligned together in each dimension. This will increase the density of the triangulation for this dimension. The triangulation is defined by Eq. 2, with the edge facets of a k-simplex given by $\tilde{t}$ with $0 \leq k \leq n - 1$.

$$\mathcal{T} := \bigcup_{i=1}^{J} t_i, \quad t_i \cap t_j \in \{\emptyset, \tilde{t}\}, \quad \forall t_i, t_j \in \mathcal{T} \tag{2}$$

B. Barycentric Coordinates

Since a n-simplex is affinely independent, an affine coordinate system can be used for the location of a point in a simplex. A point within a simplex is determined relative to the vertices of this simplex. One can say that a point is represented by the "mass" each vertex has, hence the word barycenter which is another word for center of mass. The barycentric coordinates used in MSBS are normalized, i.e.

$$1 = \sum_{i}^{n} b_i \tag{3}$$

Basically, this means that when a point is located on a vertex, the corresponding barycentric coordinate equals one. A relation between the Cartesian coordinates and the barycentric coordinates is determined by the simplex size and location in Cartesian space. This linear transformation is given by

$$\begin{bmatrix} b_1 & b_2 & \ldots & b_n \end{bmatrix}^T = \mathbf{A}_{t_j}^{-1} \cdot \left(\mathbf{x}^T - \boldsymbol{\nu}_0^T \right), \tag{4}$$

with

$$\mathbf{A}_{t_j} = \begin{bmatrix} (\boldsymbol{\nu_1} - \boldsymbol{\nu_0})^T & (\boldsymbol{\nu_2} - \boldsymbol{\nu_0})^T & \ldots & (\boldsymbol{\nu_n} - \boldsymbol{\nu_0})^T \end{bmatrix}, \tag{5}$$

and for b_0

$$b_0 = 1 - \sum_{i=1}^{n} b_i, \tag{6}$$

where $\mathbf{A}_{t_j}$ is the transformation matrix of simplex j transforming the Cartesian coordinates $\mathbf{x} \in \mathbb{R}^n$ to barycentric coordinates $\mathbf{b} \in \mathbb{R}^{n+1}$ using the Cartesian coordinates of the simplex vertices $\boldsymbol{\nu} \in \mathbb{R}^n$. Equations (4), (5), and (6) can be parameterized as an affine function, [7], given by

$$\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} k_0 & a_{01} & a_{02} & \dots & a_{0n} \\ k_1 & a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & & \vdots \\ k_n & a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{or similar} \quad \mathbf{b} = \mathbf{T^1} \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \tag{7}$$

and a simplified notation given by Eq. 8.

$$b = b_{t_j}(x) \;\; \forall x \in t_j \tag{8}$$

For the determination if a data point x is within a simplex t_j a simple rule can be applied using the barycentric coordinates, i.e.

$$\begin{aligned} b_i > 0 &\rightarrow \mathbf{x} \in t_j, \;\; \forall i \\ b_i < 0 &\rightarrow \mathbf{x} \notin t_j, \;\; 0 \leq i \leq n \end{aligned} \tag{9}$$

C. Barycentric Bernstein Basis

For MSBS a barycentric Bernstein basis is used. The Bernstein basis of degree $d \geq 0$ are determined using a multinomial expansion given by Eq. 10. The multinomial theorem is a generalized form of the binomial theorem.

$$\begin{aligned} (b_0 + b_1 + \ldots + b_n)^d &= \sum_{|\kappa|=d} \frac{d!}{\kappa!} b_0^{\kappa_0} \cdot b_1^{\kappa_1} \cdot \ldots \cdot b_n^{\kappa_n} \\ &= \sum_{|\kappa|=d} \frac{d!}{\kappa!} b^{\kappa} \\ &= \sum_{|\kappa|=d} B_{\kappa}^{d}(b) \\ &= 1 \end{aligned} \tag{10}$$

In Eq. (10) the multi-index κ is given by

$$\kappa := (\kappa_0, \kappa_1, \ldots, \kappa_n) \in \mathbb{N}^{n+1}, \tag{11}$$

with properties

$$\begin{aligned} |\kappa| &= \kappa_0 + \kappa_1 + \ldots + \kappa_n, \\ \kappa! &= \kappa_0! \cdot \kappa_1! \cdot \ldots \cdot \kappa_n!, \end{aligned} \tag{12}$$

and elements of κ are sorted lexicographically such that,

$$\kappa \in \{(d, 0, 0, \ldots, 0), (d-1, 1, 0, \ldots, 0), (d-1, 0, 1, \ldots, 0), \ldots, (0, \ldots, 0, 1, d-1), (0, \ldots, 0, 0, d)\}. \tag{13}$$

Total number of permutations valid for κ, and also the number of basis terms, equals

$$\hat{d} = \binom{d+n}{n} = \frac{(d+n)!}{n!d!} \tag{14}$$

The barycentric Bernstein basis has some pleasant properties, including partition of unity, non-negativity, symmetry and, recursively. The partition of unity, i.e. the summation to one, is the result of the normalized barycentric coordinates. Together, these properties make the basis function stable. This implies that in barycentric space all simplices are equal, no matter if they differ in size in Cartesian space. By definition the basis functions are only locally active in a simplex and zero elsewhere, making the barycentric Bernstein basis besides stable also local.

D. B-Form

The Cartesian to barycentric coordinate transformation, Eq. (8), and the barycentric Bernstein basis, Eq. (10), are combined into the so called B-form. De Boor proved, [8], that any polynomial $p(x)$ of degree d can be written in the B-form:

$$p(x) = \sum_{|\kappa|=d} c_{b_\kappa}^{t_j} B_\kappa^d(b_{t_j}(x)) \quad \forall x \in t_j, \tag{15}$$

and by definition

$$p(x) = 0 \quad \forall x \notin t_j. \tag{16}$$

An interesting and very important feature of the B-coefficients c_b is their unique spatial location within a simplex. The barycentric coordinate system makes it possible to ask the question: where is my B-coefficient located? This property will be very important to generate continuity within the spline model but also for validation. Other properties of the B-form include: easy change of formulation to vector notation, the simplex creates end-point values and bounds for the polynomial, the derivative can be expressed in original B-from, the integral is equal to a constant multiplied to the sum of all B-coefficients, variation of diminishing makes root isolation possible, and it is determined by a recursive algorithm. The theory of this section makes it possible to create MSBS models with the spline space defined in [9] by Eq. 17,

$$S_d^r(\mathcal{T}) := \{s \in C^r(\mathcal{T}) : s|_t \in \mathcal{P}_d, \ \forall t \in \mathcal{T}\}, \tag{17}$$

where s is a n-simplex spline function with degree d and continuity of order r on triangulation $\mathcal{T}$. The space of all polynomials with degree d is defined by $\mathcal{P}$. A more comprehensive review of the MSBS is given in [9]

III. Multinomial Expansion using Kronecker Substituted Convolution

Univariate polynomial multiplications can be performed in a numerical fashion using a discrete convolution. Since the multinomial theorem is a polynomial multiplication, d-times with itself, discrete convolution can be used to expand this polynomial. However since the multinomial polynomial is multivariate a Kronecker Substitution is required first, separating the terms sufficiently such that they do not conflict each other during convolution, see [10]. The multinomial theorem was introduced in Eq. 10. The multi-index κ, used in the multinomial, can be written in a matrix notation as follows

$$\boldsymbol{\kappa} = \begin{bmatrix} d & 0 & 0 & \dots & 0 \\ d-1 & 1 & 0 & \dots & 0 \\ d-1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 1 & d-1 \\ 0 & \dots & 0 & 0 & d \end{bmatrix} = \begin{bmatrix} \boldsymbol{\kappa}_{0,\bullet} \\ \vdots \\ \boldsymbol{\kappa}_{\hat{d},\bullet} \end{bmatrix} \in \mathbb{R}^{\hat{d}\times n+1}, \text{ with property } \begin{bmatrix} ||\boldsymbol{\kappa}_{0,\bullet}||_1 \\ \vdots \\ ||\boldsymbol{\kappa}_{\hat{d},\bullet}||_1 \end{bmatrix} = \begin{bmatrix} d \\ \vdots \\ d \end{bmatrix} \tag{18}$$

In the remainder of this section, an alternative approach is introduced to expand the multinomial polynomial using a Kronecker substitution in combination with a discrete convolution. The multinomial polynomial to be expanded is represented in a vector notation. The Kronecker Substitution conditioner is given by G. Multiple convolution methods are available. In Eq. 19 convolution is performed as a multiplication in the frequency domain. The preconditioned multinomial polynomial is transformed to the frequency domain by using a Discrete Fourier Transform (DFT) given by W. The DFT is a linear transformation matrix. Fast algorithms exist for the creation of W, [11] and [12]. The preconditioning and DFT are combined in a single matrix operation F. The Kronecker Substitution and inverse DFT are reversed with F_I. Both matrices only depend on the dimension n and degree d. This combination creates a fast reduced matrix transform.

$$\begin{aligned} (b_0 + \ldots + b_n)^d &= \left(\begin{bmatrix} b_0 & \dots & b_n \end{bmatrix} \mathbf{GW}\right)^{\odot d} \mathbf{W}_I \mathbf{G}_I \\ &= \left(\begin{bmatrix} b_0 & \dots & b_n \end{bmatrix} \mathbf{F}\right)^{\odot d} \mathbf{F}_I \\ &= F_d^{n,d}(b) \end{aligned} \tag{19}$$

This method can also be used to multiply two different multinomial polynomials, for example $\beta_1(b)$ and $\beta_2(b)$ as is shown in Eq. 20.

$$\begin{aligned}(b_0 + \ldots + b_{n_1})^{d_1} \cdot (b_0 + \ldots + b_{n_1} + \ldots + b_{n_2})^{d_2} &= \left(\left(\begin{bmatrix} b_0 & \ldots & b_{n_1} & \ldots & 0 \end{bmatrix} \mathbf{F}\right)^{\odot d_1} \odot \left(\begin{bmatrix} b_0 & \ldots & b_{n_2} \end{bmatrix} \mathbf{F}\right)^{\odot d_2}\right) \mathbf{F}_I \\ &= F_{d_1,d_2}^{n_{\max},d_{\max}}(\beta_1(b), \beta_2(b))\end{aligned} \tag{20}$$

The Kronecker substitution depends on the model degree d and model dimension n and the scheme is given by the function:

$$\gamma(j) = \left\lfloor (2 \cdot d + 2)^{j-1} \right\rfloor. \tag{21}$$

Its reverse, including the preservation of the lexicographical order, can be determined by using a vectorized version of the multi-index κ introduced in Eq. 18.

$$g(\boldsymbol{\kappa}_{j,\bullet}) = \sum_{l=1}^{n} \left(\boldsymbol{\kappa}_{j,l} \cdot \gamma(l)\right) \tag{22}$$

Next the Kronecker substitution is combined with the DFT in matrix form. The matrix is indexed by j for its rows and k for its columns and the elements are determined with Eq. 23.

$$\mathbf{F} = \left(\omega^{\gamma(j)\cdot k}\right)_{j=0,\ldots,n\,,\,k=0,\ldots,N-1} \tag{23}$$

The reverse of the Kronecker substitution is combined with an inverse DFT, where the elements are given by Eq. 24.

$$\mathbf{F}_I = \left(\frac{\overset{*}{\omega}^{g(\boldsymbol{\kappa}_{j,\bullet})\cdot k}}{N}\right)_{j=0,\ldots,\hat{d}-1\,,\,k=0,\ldots,N-1} \tag{24}$$

In equations Eq. (23) and Eq. (24) the primitive N^{th} root of unity, in which $i = \sqrt{-1}$, is given by

$$\omega = e^{-2\pi i/N}$$

and its conjugate

$$\overset{*}{\omega} = e^{2\pi i/N} \tag{25}$$

and

$$N = d \cdot \gamma(n) + 1$$

What becomes clear is that mainly the dimensions n is driving the F-form. When $d \to \infty$ all columns of the multinomial expansions $> \hat{d}$ are equal to zero. The dimension is essential to determine the regrouping of terms to lexicographical order. When the value output of the expansion is of importance only it does not matter which degree or dimensions is set, provided that the dimension and degree are equal or larger to the dimension and degree of the expansion.

IV. Physical-Splines

When using MSBS for Aerodynamic Model Identification, the estimated parameters cannot be physically interpreted. The B-coefficients do not have a straightforward physical interpretation. The model structure of every simplex is in the form of a polynomial in the barycentric coordinates system. The physical parameters, or stability derivatives, form a very important part of the flight dynamics. They are studied extensively and are used to analyze the stability of an aircraft. Therefore a transformation from the barycentric space to physical space will be very advantageous. For example to transform the B-coefficients to P-coefficients.

A. Physical Multinomial Polynomials

In search for the physical equivalent Eq. 19 is substituted first into the B-form of Eq. 15, such that

$$\begin{aligned}p(b) &= \left(\begin{bmatrix} b_0 & \ldots b_n \end{bmatrix} \mathbf{F}\right)^{\odot d} \mathbf{F}_I \, \mathbf{c}_b \\ &= F_d^{n,d}(b) \cdot c_b.\end{aligned} \tag{26}$$

By using the Cartesian to Barycentric coordinates transformation matrix, Eq. 7 a relation between the barycentric B-form and physical B-form is obtained.

$$p(x) = \left(\mathbf{T}^1 \begin{bmatrix} 1 & x_1 & \dots & x_n \end{bmatrix} \mathbf{F}\right)^{\odot d} \mathbf{F}_I \, \mathbf{c}_b \tag{27}$$

The latter equations reveals the multinomial polynomials. Next we are in search for the pseudo physical coefficients $\tilde{c}_p$ such that the transformation is removed from the multinomial expansion and Eq. 27 equals Eq. 28.

$$p(x) = \left(\begin{bmatrix} 1 & x_1 & \dots & x_n \end{bmatrix} \mathbf{F}\right)^{\odot d} \mathbf{F}_I \, \tilde{\mathbf{c}}_p \tag{28}$$

The coefficients are pseudo physical since the the multinomial coefficients are still included in the multinomial expansion. The relation between these pseudo coefficients and physical coefficients is given by

$$\mathbf{c}_p = \mathbf{M}_c \, \tilde{\mathbf{c}}_p \tag{29}$$

where the multinomial coefficient matrix is calculated with

$$\mathbf{M}_c = \begin{bmatrix} \frac{d!}{\boldsymbol{\kappa}_{0,\bullet}!} & 0 & \dots & 0 \\ 0 & \frac{d!}{\boldsymbol{\kappa}_{1,\bullet}!} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \frac{d!}{\boldsymbol{\kappa}_{\hat{d},\bullet}!} \end{bmatrix} \quad \text{and} \quad \mathbf{M}_c^{-1} = \begin{bmatrix} \frac{\boldsymbol{\kappa}_{0,\bullet}!}{d!} & 0 & \dots & 0 \\ 0 & \frac{\boldsymbol{\kappa}_{1,\bullet}!}{d!} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \frac{\boldsymbol{\kappa}_{\hat{d},\bullet}!}{d!} \end{bmatrix} \tag{30}$$

$$\mathbf{M}_c = \operatorname{diag}\left(\left(\begin{bmatrix} 1 & \dots & 1_{n+1} \end{bmatrix} \mathbf{F}\right)^{\odot d} \mathbf{F}_I\right)$$

Hence the true physical B-form is obtained, after substituting Eq. 29 in Eq. 27, resulting in Eq. 31.

$$\begin{aligned} p(x) &= \left(\begin{bmatrix} 1 & x_1 & \dots & x_n \end{bmatrix} \mathbf{F}\right)^{\odot d} \mathbf{F}_I \, \mathbf{M}_c^{-1} \, \mathbf{c}_p \\ &= F_d^{n,d}(x) \cdot c_p \end{aligned} \tag{31}$$

Basically the term C_m^{-1} removes the multinomial coefficient after the multinomial expansions and includes them into the physical coefficients. Equation 31 yields the equivalent physical formulation of the B-form. The physical B-form with physical P-coefficients is now given by

$$\begin{aligned} p(x) &= \sum_{|\kappa|=d} \frac{d!}{\kappa_0! \dots \kappa_n} \tilde{c}_{p_{\kappa'}} 1^{\kappa_0} x_1^{\kappa_1} \dots x_n^{\kappa_n} \\ &= \sum_{|\kappa|=d} c_{p_{\kappa'}} 1^{\kappa_0} x_1^{\kappa_1} \dots x_n^{\kappa_n} \end{aligned}$$

with multi-indexes:

$$\begin{aligned} \kappa &:= (\kappa_0, \kappa_1, \dots, \kappa_n) \in \mathbb{N}^{n+1} \\ \kappa' &:= (\kappa_1, \kappa_2, \dots \kappa_n) \in \mathbb{N}^n \end{aligned} \tag{32}$$

with properties:

$$\begin{aligned} |\kappa| &= \kappa_0 + \kappa_1 + \dots + \kappa_n, \\ \kappa! &= \kappa_0! \cdot \kappa_1! \cdot \dots \cdot \kappa_n!, \end{aligned}$$

B. Barycentric Space to Physical Space Transformation

A simple conversion between the B-form and physical B-form is described in [13]. This reference uses a "Bézier simplex" for root finding and degree reduction and shows that there is a one-to-one relation between the multinomial physical B-form, Eq. (32), and the barycentric B-form, Eq. (15) . However this conversion is only applicable for a

single standard simplex, where the vertices are located along the coordinate axes in a positive direction at a unit distance from the origin. In this section a method for a generalized creation of the physical-splines transformation matrix is suggested. It is given as a matrix operation and makes use of the Kronecker Substituted DFT method introduced in section III. First, the relation between the MSBS and the physical-splines is elaborated. Subsequently we are interested in a linear relation between the B-coefficients c_b and physical coefficients c_p. Starting with the B-form, Eq. 15, and substituting the relation between the physical variables and barycentric variables, the following relation is obtained

$$\begin{aligned} p(b) &= \sum_{|\kappa|=d} \frac{d!}{\kappa_0! \dots \kappa_n!} c_{b_\kappa} b_0^{\kappa_0} \dots b_n^{\kappa_n} \\ p(x) &= \sum_{|\kappa|=d} \frac{d!}{\kappa_0! \dots \kappa_n!} c_{b_\kappa} (k_0 + a_{01}x_1 + \dots + a_{0n}x_n)^{\kappa_0} \dots (k_0 + a_{01}x_1 + \dots + a_{0n}x_n)^{\kappa_n} \\ &= \sum_{|\kappa|=d} \frac{d!}{\kappa_0! \dots \kappa_n!} c_{b_\kappa} F_\kappa^{n,d}(\mathbf{T}^1_{0,\bullet}, \dots, \mathbf{T}^1_{n,\bullet}) \end{aligned} \tag{33}$$

Expanding and regrouping terms in Eq. 33 yields a relation between the physical variables and the B-coefficients. Taking a closer look, Eq. (33) appears to have multiple multinomial expansions within the multinomial expansion itself. The equation can be rewritten to a vector equation using the Kronecker substituted DFT in order to obtain a generalized expression for this linear barycentric to physical space relation.

$$p(x) = \sum_{|\kappa|=d} \frac{d!}{\kappa_0! \dots \kappa_n!} c_{b_\kappa} \left(\left(\mathbf{T}_{0,\bullet} \odot \begin{bmatrix} 1 & x_1 & \dots x_n \end{bmatrix} \mathbf{F} \right)^{\odot \kappa_0} \odot \dots \odot \left(\mathbf{T}_{n,\bullet} \odot \begin{bmatrix} 1 & x_1 & \dots x_n \end{bmatrix} \mathbf{F} \right)^{\odot \kappa_n} \right) \mathbf{F}_I \tag{34}$$

Previous equation is rewritten in such a way that the physical variables are separated from the inner multinomial expansion. The physical variables appear to be independent from the multi-index κ, i.e. they remain the same throughout the main multinomial expansion.

$$\begin{aligned} p(x) &= \sum_{|\kappa|=d} \left(\left(\begin{bmatrix} 1 & x_1 & \dots x_n \end{bmatrix} \mathbf{F} \right)^d \mathbf{F}_I \right) \mathbf{M}_c^{-1} \left((\mathbf{T}_{0,\bullet} \mathbf{F})^{\odot \kappa_0} \odot \dots \odot (\mathbf{T}_{n,\bullet} \mathbf{F})^{\odot \kappa_n} \right) \mathbf{F}_I \frac{d!}{\kappa_0! \dots \kappa_n!} c_{b_\kappa} \\ &= \sum_{|\kappa|=d} F_d^{n,d}(x) \, \Lambda_\kappa^{n,d}(T^1) \, c_{b_\kappa} \end{aligned} \tag{35}$$

After grouping the terms it becomes clear that the transformation is given by $\Lambda_\kappa^{n,d}(T^1)$ and the B-form regression matrix by $F_d^{n,d}(x)$. By defining the multi-index κ as a matrix, the transformation can be rewritten to a single multinomial DFT expansion depending on the dimension, degree, and triangulation only.

$$\begin{aligned} \mathbf{\Lambda}^{n,d} &= \left(\left((\mathbf{1} \ \mathbf{T}_{0,\bullet} \ \mathbf{F})^{\odot \boldsymbol{\kappa}_{\bullet,0}} \odot \dots \odot (\mathbf{1} \ \mathbf{T}_{n,\bullet} \mathbf{F})^{\odot \boldsymbol{\kappa}_{\bullet,n}} \right) \ \mathbf{F}_I \, \mathbf{M}_c \right)^\top \\ &= \Lambda_{\boldsymbol{\kappa}}^{n,d}(T^1) \end{aligned}$$

where

$$\begin{aligned} &\mathbf{1} \in \mathbb{N}^{\hat{d} \times 1} \\ &\mathbf{T}^1 \in \mathbb{R}^{n+1 \times n+1} \\ &\mathbf{F} \in \mathbb{C}^{n+1 \times N} \\ &\mathbf{F}_I \in \mathbb{C}^{\hat{d} \times N} \\ &\mathbf{M}_c \in \mathbb{R}^{\hat{d} \times \hat{d}} \end{aligned} \tag{36}$$

An application of this transformation is, for example, to transform B-coefficients c_b to P-coefficients c_p. This transformation is stated in Eq. (37).

$$\mathbf{c}_{p_{t_j}} = \mathbf{\Lambda}_{t_j} \mathbf{c}_{b_{t_j}}, \tag{37}$$

where $\mathbf{\Lambda}_{t_j} \in \mathbb{R}^{\hat{d} \times \hat{d}}$ is symmetric and non-singular for any non-degenerate simplex. Hence the inverse transform is equal to

Table 1 Longitudinal and Lateral Model

Model	Limited to moderate non-linearities	Significant non-linearities
Longitudinal (X, Z, m)	$M,\ h,\ \delta_e,\ T_e$	$\hat{\alpha},\ \hat{\dot{\alpha}}, \hat{q}$
Lateral (Y, l, n)	$M,\ h,\ \delta_a,\ \delta_r$	$\hat{\beta},\ \hat{\dot{\beta}},\ \hat{p},\ \hat{r}$

$$\mathbf{c}_{b_{t_j}} = \mathbf{\Lambda}_{t_j}^{-1}\mathbf{c}_{p_{t_j}}. \tag{38}$$

The individual transformations $\mathbf{\Lambda}_{t_j}$ can be determined for all simplices and can be combined in a a large sparse transformation matrix valid for the complete triangulation

$$\begin{aligned}\mathbf{c}_p &= \begin{bmatrix} \mathbf{\Lambda}_{t_1} & 0 & 0 & 0 \\ 0 & \mathbf{\Lambda}_{t_2} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{\Lambda}_{t_j} \end{bmatrix} \mathbf{c}_b \\ &= \mathbf{\Lambda}\mathbf{c}_b.\end{aligned} \tag{39}$$

V. Aerodynamic Model Identification with Physical-Splines

Aerodynamic modeling is a system identification process in creating an aerodynamic model of the flight dynamics of an aircraft. In essence, system identification relates to the procedure of giving an input while measuring the output and subsequently determining the model. Modeling certain flight envelopes, for example stall, show high non-linear behavior, requiring advanced modeling technique such as the physicals-splines method. This section explains how the filtered and reconstructed flight-test data can be used in creating an aerodynamic model with physical-splines and how the properties of the physical-splines can contribute to the MSBS method.

A. Equations of Motion

An aerodynamic model of an aircraft is governed by the equations of motion (EOM). The EOM describe the flight dynamics in the form of aerodynamic force and moments. In this section the final EOM are stated. For a more extensive derivation, [14] is recommended. An assumption is made that they can be described using analytical polynomial functions. A general definition in aerodynamic modeling is that the aerodynamic forces and moments are a function of the state variables and the control variables δ. The variables used for modeling the six different dimensionless force and moment equations are given by (40). In this equation M is the Mach number, h the altitude, α the angle of attack, β the side-slip angle and p, q, r the rotational rates in the body-frame. δ_a, δ_e, δ_r are the aileron, elevator and rudder deflections respectively. The dot indicates a time derivative.

$$C_i = \left(M,\ h,\ \alpha,\ \hat{\dot{\alpha}},\ \beta,\ \hat{\dot{\beta}},\ \hat{p},\ \hat{q},\ \hat{r},\ \delta\right) \quad \text{for } i = X,\ Y,\ Z,\ l,\ m,\ n \quad \text{and} \quad \delta = (\delta_a,\ \delta_e,\ \delta_r) \tag{40}$$

Furthermore, it is assumed that the aerodynamic forces and moment are decoupled in longitudinal and lateral direction. This decoupling is given in table 1, where for the different states the expected (non-)linearities of a model is given.

For consistency and better general understanding of the estimated parameters, the state variables are made dimensionless (indicated with a hat symbol) by using the definition

$$M := \frac{V}{a_0},\ \hat{p} := \frac{pb}{V},\ \hat{q} := \frac{q\bar{c}}{V},\ \hat{r} := \frac{rb}{V},\ \hat{\dot{\alpha}} := \frac{\dot{\alpha}\bar{c}}{V},\ \hat{\dot{\beta}} := \frac{\dot{\beta}b}{V} \tag{41}$$

The dimensionless forces and moments are given by Eqs. 42 - 44

$$C_X = \frac{X}{\frac{1}{2}\rho V^2 S} = \frac{m(A_X - \lambda_{a_x}) - T_x}{\frac{1}{2}\rho V^2 S}, \tag{42}$$

$$C_Y = \frac{Y}{\frac{1}{2}\rho V^2 S} = \frac{m(A_Y - \lambda_{a_y})}{\frac{1}{2}\rho V^2 S}, \tag{43}$$

$$C_Z = \frac{Z}{\frac{1}{2}\rho V^2 S} = \frac{m(A_Z - \lambda_{a_z})}{\frac{1}{2}\rho V^2 S}, \tag{44}$$

and the dimensionless moments are given by Eqs. 45 - 47

$$C_l = \frac{L}{\frac{1}{2}\rho V^2 S b} = \frac{\dot{p} I_{xx} + (q - \lambda_p)(r - \lambda_r)\left(I_{zz} - I_{yy}\right) - \left((p - \lambda_p)(q - \lambda_q) + \dot{r}\right) I_{xz}}{\frac{1}{2}\rho V^2 S b}, \tag{45}$$

$$C_m = \frac{M}{\frac{1}{2}\rho V^2 S \bar{c}} = \frac{\dot{q} I_{yy} + (r - \lambda_r)(p - \lambda_p)\left(I_{xx} - I_{zz}\right) + \left((p - \lambda_p))^2 - (r - \lambda_r)^2\right) I_{xz} - M_T}{\frac{1}{2}\rho V^2 S \bar{c}}, \tag{46}$$

$$C_n = \frac{N}{\frac{1}{2}\rho V^2 S b} = \frac{\dot{r} I_{zz} + (p - \lambda_p)(q - \lambda_q)\left(I_{yy} - I_{xx}\right) + \left((q - \lambda_q)(r - \lambda_r) - \dot{p}\right) I_{xz}}{\frac{1}{2}\rho V^2 S b}. \tag{47}$$

A bias term is obtained during the flight path reconstruction using an earlier toolbox developed by the task-force [15]. This bias term is indicated by λ in Eqs. 42 - 47.

B. Constraint Least Squares Estimator

Since the sparse B-regression matrix, Eq. (48), in barycentric space is normalized, and therefore reducing its condition number, the identification will take place in barycentric space. The estimated B-coefficients are transformed to P-coefficients afterwards. Transforming this regression matrix to physical space can introduce numerical problems during identification and will therefore be used only for a model structure selection. Equation 48 holds both the B-regression matrix, [16], of the triangulation and the to be optimized B-coefficients.

$$\mathbf{X}_b = \begin{bmatrix} \mathbf{B}^d(\mathbf{b}_{t_1}(x_{t_j})) & 0 & 0 & 0 \\ 0 & \mathbf{B}^d(\mathbf{b}_{t_2}(x_{t_2})) & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{B}^d(\mathbf{b}_{t_j}(x_{t_j})) \end{bmatrix}, \quad \mathbf{c}_b = \begin{bmatrix} \mathbf{c}_b^{t_1} \\ \mathbf{c}_b^{t_2} \\ \vdots \\ \mathbf{c}_b^{t_j} \end{bmatrix} \tag{48}$$

Each element in Eq. (48) represents a simplex and can be determined with vector-notation of the B-form stated in Eq. (15).

$$\mathbf{B}^d(\mathbf{b}_{t_j}(x)) = \begin{bmatrix} B^d_{d,0,0}(b_{t_j}(x_{1_{t_j}})) & B^d_{d-1,1,0}(b_{t_j}(x_{1_{t_j}})) & \dots & B^d_{0,1,d-1}(b_{t_j}(x_{1_{t_j}})) & B^d_{0,0,d}(b_{t_j}(x_{1_{t_j}})) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ B^d_{d,0,0}(b_{t_j}(x_{M_{t_j}})) & B^d_{d-1,1,0}(b_{t_j}(x_{M_{t_j}})) & \dots & B^d_{0,1,d-1}(b_{t_j}(x_{M_{t_j}})) & B^d_{0,0,d}(b_{t_j}(x_{M_{t_j}})) \end{bmatrix}, \tag{49}$$

where x indicates the M number of flight-test data measurements, belonging to simplex t_j. The B-coefficient vector to be estimated is given by

$$\mathbf{c}_b^{t_j} = \begin{bmatrix} c^{t_j}_{d,0,0} & c^{t_j}_{d-1,1,0} & \dots & c^{t_j}_{0,1,d-1} & c^{t_j}_{0,0,d} \end{bmatrix}^T. \tag{50}$$

Together they form the linear model to be optimized, with respect to the B-coefficients, by minimizing the residuals ϵ. In Eq. 51 y is the vector of the measured output and the hât indicates an estimation.

$$\mathbf{y} = \mathbf{X}_b \hat{\mathbf{c}}_b + \boldsymbol{\epsilon} \tag{51}$$

The function approximator is compatible with linear regression methods such as the Ordinary Least Squares (OLS) [17] and the Generalized Least Squares (GLS) estimators [18]. Very important for the MSBS is the so called B-net structure formed by the B-coefficients. The B-net structure is unique for a MSBS model and can only be adjusted by changing the model dimension and model degree. B-coefficients cannot be deleted, hence removing columns from the regression matrix Eq. 49 will break the B-net structure! However, in physical space it is possible to influence the model structure, since the coefficients do not have a spatial location. A desired physical model structure can be imposed in the form of equality and inequity constraints and transformed to the barycentric space using the transformation matrix. Thereafter the constraints are subjected to the optimization function creating a constraint optimization. This way it is possible to interact with model optimization process, besides adjusting the dimension and degree only, without breaking the B-net structure. The optimization function, subjected to constraints, is summarized in Eqs. 52 - 56. They are known as the KKT-conditions, [19], [20].

$$\underset{c_b}{\textbf{argmin}} \quad J = ||\mathbf{y} - \mathbf{X}_b\hat{\mathbf{c}}_b||_2 \qquad \text{Optimization Function} \tag{52}$$

$$\textbf{subjected to} \quad H(c_b) = h_e \qquad \text{Barycentric equality constraints} \tag{53}$$

$$G(c_p) = g_e \qquad \text{Physical equality constraints} \tag{54}$$

$$G_u(c_p) \leq g_u \qquad \text{Physical inequality constraints, upper bound} \tag{55}$$

$$G_l(c_p) \geq g_l \qquad \text{Physical inequality constraints, lower bound} \tag{56}$$

In the remainder of this section a method is introduced that solves optimization functions subjected to constraints and useful physical constraints are introduced. These constraints are created in both barycentric space and physical space and are equality constraints or bounded inequality constraints. Barycentric constraints are indicated with the letter H and physical constraints with the letter G. Subscripts e and i indicate equality and inequality constraints respectively. The upper and lower bound are indicate by subscripts u and l in the given order. The lower bound inequality constraints is transformed in a virtual lower bound constraint by multiplying the constraint with -1. The constraints are summarized in Eq. 57.

$$\mathbf{H}_e = \begin{bmatrix} \mathbf{H} \\ \mathbf{G}\boldsymbol{\Lambda}^{-1} \end{bmatrix}, \quad \mathbf{h}_e = \begin{bmatrix} \mathbf{h}_e \\ \mathbf{g}_e \end{bmatrix}, \quad \mathbf{H}_i = \begin{bmatrix} \mathbf{G}_u\boldsymbol{\Lambda}^{-1} \\ -\mathbf{G}_l\boldsymbol{\Lambda}^{-1} \end{bmatrix}, \quad \mathbf{h}_i = \begin{bmatrix} \mathbf{g}_u \\ -\mathbf{g}_l \end{bmatrix}, \tag{57}$$

such that

$$\mathbf{H}_e\mathbf{c}_b = \mathbf{h}_e \quad \text{and} \quad \mathbf{H}_i\mathbf{c}_b \leq \mathbf{h}_i. \tag{58}$$

Different methods are possible to solve the constraint optimization, among them are the active-set method and interior-point methods. A promising solver is the Mehrotra predictor-corrector method [21], which is a specific interior-point method. Initially this method was designed to solve linear programming, but it can be very easily rewritten to solve a quadratic programming optimization such as Eq. (52). The KKT-conditions are transformed to Lagrangian form given in Eq. 59 with λ the Lagrangian multipliers and s slack variables transforming inequality constraints into equality constraints.

$$\begin{aligned} \nabla\mathbb{J} = \left(\mathbf{X}_b^T\mathbf{X}_b\right)\mathbf{c}_b - \mathbf{X}_b^T\mathbf{y}\mathbf{c}_b - \mathbf{H}_e^T\boldsymbol{\lambda}_1 - \mathbf{H}_i^T\boldsymbol{\lambda}_2 &= 0 \\ \mathbf{H}_e\mathbf{c}_b - \mathbf{h}_e &= 0 \\ \mathbf{H}_i\mathbf{c}_b - \mathbf{h}_i - \mathbf{s} &= 0 \\ \mathbf{S}\boldsymbol{\lambda}_2 &= 0 \\ \mathbf{s} &\geq 0 \\ \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 &\geq 0 \end{aligned} \tag{59}$$

The idea of the Mehrotra predictor-corrector method is that the search direction is updated by a predictor based on solving the Newton-Raphson step update of Eq. (60). Equation (60) can be rewritten to a symmetrical matrix and therefore a Cholesky decomposition. This simplifies the computation time required to determine the inverse required for the Newton step. In order to maintain a "centered position" [22] a centering step is calculated. The corrector step is required to satisfy the complementarity conditions r_c. After the corrector step, the step size used can be adjusted to find

a better step. This makes it possible to reuse the Cholesky decomposition again reducing the required iterations to solve the optimization and decreasing the computational cost.

$$\begin{bmatrix} \mathbf{X}_b^T\mathbf{X}_b & -\mathbf{H}_e^T & -\mathbf{H}_i^T & 0 \\ \mathbf{H}_e & 0 & 0 & 0 \\ \mathbf{H}_i & 0 & 0 & \mathbf{I} \\ 0 & 0 & \mathbf{S} & \boldsymbol{\lambda}_2 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{c}_b \\ \Delta\boldsymbol{\lambda}_1 \\ \Delta\boldsymbol{\lambda_2} \\ \Delta\mathbf{s} \end{bmatrix} = - \begin{bmatrix} \left(\mathbf{X}_b^T\mathbf{X}_b\right)\mathbf{c}_b - \mathbf{X}_b^T\mathbf{y}\mathbf{c}_b - \mathbf{H}_e^T\boldsymbol{\lambda}_1 - \mathbf{H}_i^T\boldsymbol{\lambda}_2 \\ \mathbf{H}_e\mathbf{c}_b - \mathbf{h}_e \\ \mathbf{H}_i\mathbf{c}_b - \mathbf{h}_i - \mathbf{s} \\ \mathbf{S}\boldsymbol{\lambda}_2 \end{bmatrix} = \begin{bmatrix} r_d \\ r_e \\ r_i \\ r_c \end{bmatrix} \tag{60}$$

C. Smoothness and Continuity Conditions

The B-coefficients have a unique spatial location within each simplex. This spatial location makes it possible to generate continuity constraints between the simplices and make the splines-model continuous. The continuity of order C^r between the simplex polynomials can be important for several reasons. First zero order continuity makes sure no inner "gaps" in the model between simplices exist. When the aerodynamic model is used in a full-motion simulator this will prevent excessive and possibly damaging movements. Higher order splines-models can be required in applying control allocation methods. Also increasing the continuity will let the simplices influence each other and makes the splines-model more smooth. Hence the barycentric parameter variances decreases, see [16]. However, the parameters variances are propagate to the boundary of the triangulation and can introduce divergent behavior at these boundaries. Also poor performing simplices are likely to influence other surrounding simplices. The equality constraints for r^{th} order continuity are created using Eq. (61) and are in Barycentric space coordinate system, [23].

$$\begin{aligned} c^{t_j}_{\kappa_0,m,\kappa_1} &= \sum_{|\gamma|=m} c^{t_i}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(\nu_*), 0 \le m \le r \\ &\kappa_0 + m + \kappa_1 = d \\ &(\kappa_0 + m + \kappa_1) + (\gamma_0 + \gamma_1 + \gamma_2) = d \end{aligned} \tag{61}$$

In this equation the order of the continuity is smaller than the degree, i.e. $r < d$. With a multi-index γ independent of κ and ν_* the out-of-edge vertex of t_j.

D. Physical Model Structure Selection Constraints

The initial basis polynomial model structure of the MSBS in barycentric space can be determined by the degree only. Using the multinomial theorem the basis model is expanded to individual terms. Increasing the degree and as a matter of fact the number of terms, the approximation power of the model increases and the model error decreases. However, by doing so the possibility of over-fitting increases, i.e. instrumentation noise and measured turbulence will be modeled. Also some simplices can require a more difficult model structure because of local non-linearities. In physical space it is possible to determine an optimal model structure or to decouple the model variables on a per-simplex domain. Deleting physical model terms is accomplished by creating physical equality constraints and transform them to the barycentric space afterwards. The importance of physical model terms can be determined per simplex. Transforming the regression matrix to physical space yields

$$\mathbf{X}_p = \mathbf{X}_b\boldsymbol{\Lambda}^{-1}. \tag{62}$$

When the columns of $\mathbf{X}_p$ are independent J orthogonal bases (for each simplex) $\mathbf{W}_{t_j}$ exists. This orthogonal basis makes it possible to calculate the so-called error reduction ratio (err) for individual terms. The err determines the significance of a term in reducing the model error. In order to take the influence of the order of appearance of terms into account an orthogonal regression procedure based on stepwise regression algorithm is used, [24] and [25].

In the initial stage all terms $k = (1, 2, \ldots, \hat{d})$ in the regression matrix $\mathbf{X}_{p_{t_j}}$ are considered possible candidates for the orthogonal matrix $\mathbf{W}_{t_j}$. For each term k the err is calculated according to Eq. 63

$$\text{err}_1 = \max_{k=1}^{\hat{d}} \left(\frac{\hat{g}_k^2\, \mathbf{w}_k^T\, \mathbf{w}_k}{\mathbf{y}^T\, \mathbf{y}} \right) \quad \text{where} \quad \mathbf{w}_k = \mathbf{x}_{p_k} \quad \text{and} \quad \hat{g}_k = \frac{\mathbf{w}_k^T\, \mathbf{y}}{\mathbf{w}_k^T\, \mathbf{w}_k}. \tag{63}$$

The first term to select is $k_1 = k_{\max}$ with the highest err, i.e. $\text{err}_1 = \max\{\text{err}_k \ , \ 1 \leq k \leq \hat{d}\}$ and in combination with $\mathbf{w}_{k1} \ , \ g_{k1}$ the following relation holds

$$\mathbf{y}_{k1} = \mathbf{w}_{k1}\, \hat{g}_{k1} + \boldsymbol{\epsilon}_{k1}. \tag{64}$$

For the second iteration all terms are considered as possible candidates again except k_1 from the first iteration, i.e. $k = (1, 2, \ldots, \hat{d}) \ \forall \ k \neq k_1$. First the candidate column of $\mathbf{X}_p$ is orthogonalized with respect to $\mathbf{w}_{k1}$ using the Gram-Schmidt process given by

$$\mathbf{w}_k = x_{p_k} - \alpha_{k1\,k}\, \mathbf{w}_{k1} \quad \text{where} \quad \alpha_{k1\,2} = \frac{\mathbf{w}_{k1}^T\, \mathbf{x}_{p_k}}{\mathbf{w}_{k1}^T\, \mathbf{w}_{k1}}. \tag{65}$$

Similar as in Eq. 64 the subsequent err for $\mathbf{w}_k$ is calculated. Again select term $k_2 = k_{\max}$ with the highest err, i.e. $\text{err}_2 = \max\{\text{err}_k \ , \ 1 \leq k \leq \hat{d} \ , \ \forall \ k \neq k_1\}$ with $\mathbf{w}_{k2} = \mathbf{x}_{k2} - \alpha_{k1\,k2}\, \mathbf{w}_{k1}$ and g_{k2} and err_2
Equation (64) is updated by adding the sequential term k_2

$$\mathbf{y} = \mathbf{w}_{k1}\, \hat{g}_{k1} + \mathbf{w}_{k2}\, \hat{g}_{k2} + \boldsymbol{\epsilon}_{k1} + \boldsymbol{\epsilon}_{k2}. \tag{66}$$

Because of the orthogonalization process not only are the terms orthogonal to each other, but also the individual residual terms $\boldsymbol{\epsilon}$. Hence $\boldsymbol{\epsilon}_{k1}$ does not change when $\boldsymbol{\epsilon}_{k2}$ is added. This is similar for the orthogonal parameters. When g_{k2} is added g_{k1} does not change. This process in equations continues until the err of last column $k_{\hat{d}}$ is determined and

$$\mathbf{y} = \mathbf{w}_{k1}\, \hat{g}_{k1} + \mathbf{w}_{k2}\, \hat{g}_{k2} + \cdots + \mathbf{w}_{k\hat{d}}\, \hat{g}_{k\hat{d}} + \boldsymbol{\epsilon}_{k1} + \boldsymbol{\epsilon}_{k2} + \cdots + \boldsymbol{\epsilon}_{k\hat{d}} \tag{67}$$

At this point the order of appearance is saved and the final individual term significance is known. This process is repeated for all simplices and is summarized in the **err** vector for each simplex t_j given by (68). The sum of the ERR_{t_j} is between 0 and 1 and tells something about the quality of the polynomial structure for a simplex t_j. Higher sums are better.

$$\mathbf{err}_{t_j} = \begin{bmatrix} \text{err}_1 & \text{err}_2 & \ldots & \text{err}_{\hat{d}} \end{bmatrix}. \tag{68}$$

The Error Reduction Ratio $\mathbf{err}_{t_j}$ vector in (68) can be used to create physical model constraints in order to reduce the model size, constraint certain terms per simplex and prevent model over-fitting. Per-simplex physical polynomial model structure constraints can be created using the $\mathbf{err}_{t_j}$ using for example a stepwise introduction of terms, based on their error reduction ratio and order of appearance, see [26] and [27]. Terms are added to the model until the validation set reaches its minimum, i.e. adding more terms does no decrease the validation error but increases it. When this point k_p is reached all the remaining terms, not included in the model, are hard constraint to zero with the rule in Eq. 69

$$G_{e_{\text{err}}}(c_p) = 0 \ \forall \ \{k \mid k \in (1, 2, \ldots, \hat{d}) \ , \ k > k_p\} \tag{69}$$

E. Physical Stability Derivative Bounded Inequality Constraints

The well known aircraft stability derivatives, also known as control derivatives, are based on first order partial differential equations. In aeronautical engineering these derivatives are studied extensively. Signs and bounds are often known for an aircraft [28]. These stability derivatives appear mostly as constants, because a linearized first order Taylor expansion of the equation of motions is used as model structure. In order to model non-linear dynamical behavior, while physically interpreting the derivatives, a scaling table is often used. For this method multiple aerodynamic models are estimated and all stability derivatives are stored in large lookup tables. The models are interpolated after estimation. Physical-Splines does not make use of these lookup tables. Also one of the considerations is to increase the degree of the model structure. By doing so the stability derivatives do not have their familiar "engineering" interpretation anymore but are now formed by a function depending on one variable or multiple variables when physical parameters are not decoupled per simplex domain. However, the stability derivatives can still be studied and play an important role in the Physical-Splines method. The stability derivatives can be determined throughout a simplex domain using a grid. This approach has some similarities to the first order Taylor expansion linearizion. The first order $m = 1$ physical derivatives with respect to all physical variables are created using a degree reduction algorithm called the de Casteljau algorithm. The directional derivative for a physical function $p(\mathbf{x})$ with respect to term i is defined by

$$\nabla_{x_i}\ p(x_1, x_2, \ldots, x_n)\ ,\ 1 \leq i \leq n \tag{70}$$

In order to maintain the linear optimization, constraints are required to be linear in the parameters as well. Therefore, the stability derivative inequality constraints are applied to grid points. Since the triangulation exists of simplices it is very easy to create points within the domain of these simplices. This uniform per-simplex grid $\mathbf{x}_{\text{grid}}$ can be created for a certain density τ using Eq. 71, where ν indicates the vertices of a simplex t_j.

$$\begin{aligned} \mathbf{x}_{\text{grid}} &= \bigcup_{|\beta|=\tau} \left(\frac{1}{\tau}\beta_0\, \boldsymbol{\nu}_{0_{t_j}} + \sum_{i=1}^{n} \frac{1}{\tau}\beta_i\, \boldsymbol{\nu}_{i_{t_j}} \right) \\ \text{where } \nu &= \langle \nu_0, \nu_1, \ldots, \nu_n \rangle \in \mathbb{R}^n \\ \text{where } \beta &= (\beta_0, \beta_1, \ldots, \beta_n) \in \mathbb{R}^{n+1} \text{ is the multi-index} \\ \text{with property } |\beta| &= (\beta_0 + \beta_1 + \cdots + \beta_n) \end{aligned} \tag{71}$$

The inequality constraints that are created are now linear in parameters. For the bound two pairs of inequality constraints are created, one for the lower bound l and one for the upper bound u. They are expressed in form given in Eq. 72.

$$\mathbf{H}_{i_u}\, \mathbf{c}_b \leq \mathbf{h}_{i_u} \quad \text{and} \quad -\mathbf{H}_{i_l}\, \mathbf{c}_b \leq -\mathbf{h}_{i_l} \tag{72}$$

In Eq. 72 $\mathbf{H}_i$ is a block diagonal matrix with M differential constraints. The vector $\mathbf{c}_b$ is the B-coefficients vector and $\mathbf{d}_i \in \mathbb{R}^{M\times 1}$ is the vector with the values to constraint the differentials to. The matrix $\mathbf{H}_i$ can be formed for all simplices t_j using the differential constraints theorem of [29], given in Eq. 73.

$$\mathbf{H}_i = \text{diag}\left(\mathbf{D}_m^{t_j}(\mathbf{u}, \mathbf{x}_{\text{grid}}), 1 \leq j \leq J \right) \in \mathbb{R}^{M\times J\cdot \hat{d}} \tag{73}$$

where $\mathbf{D}_m^{t_j}(\mathbf{u}, \mathbf{x}_{\text{grid}})$ is defined as

$$\mathbf{D}_m^{t_j}(\mathbf{u}, \mathbf{x}_{\text{grid}}) := \frac{d!}{(d-m)!}\mathbf{B}^{d-m}(b(\mathcal{X}_{\text{grid}})\mathbf{P}^{d,d-m}(b(\mathbf{u})). \tag{74}$$

In Eq. 74 $\mathbf{P}^{d,d-m}(\mathbf{u})$ is the degree reduction matrix using the de Casteljau algorithm, d the degree, m the order of the differential constraints, $\mathbf{u}$ is the physical normalized directional coordinate vector. For example when $n = 2$ and $i = 1$ the direction derivative coordinate is equal to $[1\ 0]$. The constraint direction is applied for the physical coordinates in $\mathbf{x}_{\text{grid}}$. The directional coordinates and location are transformed to barycentric space and therefore the constraints do not have to be transformed afterwards.

VI. Experimental Setup

Prototyping and aerodynamic modeling with physical-splines is divided in four phases. In Fig. 1 a flow diagram of the aerodynamic modeling process, using physical-splines, is given with each phase indicated. The first phase is to collect and reconstruct flight test data, select candidate state and control variables and determine the aerodynamic model dimension. After the physical-splines initialization (gray color) where the degree, continuity triangulation density is set, the second phase starts. This phase (green color) generates the geometric support structure, also known as the triangulation and the corresponding physical transformation matrix. During the third phase (blue color) the physical polynomial model structure is determined and physical constraints and continuity conditions constraints are created. In the fourth and final phase (red color), the aerodynamic model is estimated and validated. This process is repeated with different model settings until an optimal or adequate model is found. The remainder of this section elaborates each of the different phases and explains the experimental setup. The outcome of the experiment will be the moment model C_m of the Cessna Citation II aircraft using the aforementioned methods and an elaboration how physical-splines perform and influence the aerodynamic modeling process. The experimental results are compared to the ordinary MSBS and a simple linear polynomial model.

A. Cessna Citation II

The method presented in this publication is applied to flight test data collected with the Cessna Citation II aircraft. The aircraft aerodynamic model is governed by the Equations of Motion (EOM) of Sec. A. They are used to identify the aerodynamic model describing the dynamics of the aircraft. Experimental flights are performed and during these flights different types of maneuvers are executed, such as the 3211 and doublet maneuvers. Reference [30] gives a comprehensive review on these maneuvers. The flight test data is recorded and stored. In Fig. 2 the flight envelope of the Cessna Citation II is plotted including the available flight test data, note the sparse regions in the data. The Cessna Citation II, sometimes called the Cessna 550, is a small business jet developed in the 70s and upgraded in the 90s. Almost 1000 Citation II are built. The Cessna 550 used for the test flights has the registration PH-LAB and is co-owned by Delft University of Technology and the Netherlands Aerospace Centre. The aircraft was built in 1993 and is converted to a flying laboratory. It has a modern glass cockpit and fly-by-wire flight controls. Flight test data is measured using special high-fidelity flight test instruments such as vertical gyroscopes for measuring axis-rotation, rate gyroscopes measuring axis rotational rates, IMU measuring accelerations, static-probe for the altitude, pitot-static probes for the true airspeed, AHRS, GPS for location and ground-speed, alpha vane for the angle of attack and other aircraft systems via the FMS. Furthermore a boom is available for better measurements of the angle of attack and the side-slip angle. The instruments including their specifications are listed in Tab. 11. The properties and performance of the PH-LAB are tabulated in Tab. 10.

B. Flightpath Reconstruction and Flight-Test Data

Experimental test flights are performed and measurements are made. The flight test measurements are stored in a HDF5 data format and are converted to a MATLAB data file, a .mat file, using the especially developed software HEFTIG. An earlier developed toolbox by the task-force is used to reconstruct the flightpath using these measurements. This toolbox pre-processes the data first. Tab. 11 shows the different instruments and indicates that the instruments have different sampling rates. Hence re-sampling is required for unification. Estimations of the inertia tensors I_{xx}, I_{yy}, I_{zz} are taken into account and also the aircraft empty mass, fuel mass, passenger (pax) mass and cargo mass. For the fuel mass it is important to know the rate of fuel burn, including the fuel tank dimensions and location. Moments caused by pax and cargo have to be corrected. For the flightpath reconstruction an Unscented Kalman Filter (UKF) is used. The UKF yields good results for systems with moderate to high non-linearities. The states estimates are improved in terms of accuracy and robustness by using this filter. The author refers to [15] for more background information and results about the flightpath reconstruction and the UKF.

At the moment of writing flight test data of multiple experimental flights is available, including a test flight with 38 stalls. In order to select the flight test data a candidate set is created. A candidate set includes the state and control variables that have a significant meaning to the specific forces or moments aerodynamic model of the Cessna Citation II to be created. Hence it should make sense to use them for the system identification process. A candidate set and therefore the initial aerodynamic model dimension for the Cessna Citation II was created in [31] and [32]. The candidates are determined using a method called the occurrence of hysteresis. For this experiment the candidate set for the pitching moment model C_m equals to

$$C_m(\alpha, \hat{q}, \delta_e, M) \tag{75}$$

For the pre-data selection a toolbox was developed. This toolbox uses a triangulation based clustering algorithm which can detect clusters and outlying data points. Outliers are mostly sparse data points situated in more extreme regions of the flight envelope. Sometimes these measurements are erroneous. These extremes can stretch the triangulation domain too much influencing the data distribution in simplices negatively. The toolbox also checks for "islands of simplices". Islands are single or multiple simplices separated and not connected to other "islands". This is the effect of empty simplices in the triangulation caused by missing data. Earlier research showed the minimal influence of the Mach number M to the aerodynamic model. Therefore it was decided to introduce this term as a global variable "outside" the splines model. The splines model has now a model dimension of $n = 3$.

C. Initialization

The initialization, indicated by the gray color in the flow diagram 1, sets the model degree d, triangulation density $\mathcal{T}$(nTRI), continuity order r and the stability derivative bounds. The tested models will have a triangulation density of $\mathcal{T}(1,1,1), \mathcal{T}(2,2,2), \mathcal{T}(3,3,3)$ and degrees of $d = 2$, $d = 3$, $d = 4$ and $d = 5$. The tested continuities C^r are: no continuity (C^{-1}), C^0 and C^1. Also when $d - r - 1 < 1$ the model losses the majority of its approximation power and will be skipped. With this information the B-form for a single simplex can be determined. The B-form terms include

the barycentric variables and their, to be estimated, B-coefficients. The spatial location of the B-coefficients form the B-net. The total size of B-net depends on the triangulation and number of simplices.

D. Geometric Model Structure Selection

The process of the geometric model generation is given by the green box in flow diagram 1. The creation of an optimal triangulation is a non-linear process. For this experiment it is assumed that the triangulation only exists of type-1 simplices, forming together a hypercube. These hypercubes are created with the Kuhn triangulation algorithm. The density of the triangulation can be influenced by appending multiple hypercubes in such a way that adjacent simplex faces are aligned correctly. This action can be performed in each dimension individually. The size/ domain of the triangulation is determined by the extremes of the identification dataset. This ensures that all data points are in the convex hull of the triangulation. A challenge for high dimensional triangulations is that they can let the model complexity explode. A citation by Tarentol [33]: *"There is one problem with large-dimensional spaces that is easy to underestimate: they tend to be terribly empty."* High dimensional models, i.e. when many state and control variables are included, will introduce voids and empty simplices in the triangulation. Simplices count as empty when no data points or insufficient ($\mathcal{X}_{t_j} < 2 \cdot \hat{d}$) data points are available within the simplex domain. Insufficient data will cause rank deficiency and high condition numbers. Removing simplices results in a non-rectangular domain and requires extrapolation. For this triangulation the final B-net and therefore the barycentric parameters (basis functions including the to be estimated B-coefficients) are created.

The triangulation has a corresponding physical transformation matrix ans is created with a Kronecker substituted convolution algorithm of Sec. B. With this transformation it is possible to transform from the barycentric coordinate space to the Cartesian coordinate space, for example to determine the physical-coefficients or the equivalent physical model structure of of the B-form. The transformation is also very important for the generation of physical constraints.

E. Physical Constraints and Continuity Conditions

During this phase constraints are created as indicated by the blue box in flow diagram 1. Continuity conditions between simplices are created in the form of barycentric constraints, see Sec. C. Continuity is often an initial model requirement, based on the prospective model use. However, this requirement reduces the overall approximation power of the splines model and for that reason often an increase of the degree is necessary.

The B-net and barycentric parameters are considered as a base for the model structure selection. For this set the B-form regression matrix is transformed to the physical space. In the physical space the regression matrix is normalized and the significance of individual model parameters is determined per-simplex. The stepwise orthonormalization process and determination of the error reduction ratio creates a favorable order of appearance of the terms per simplex as clarified in Sec. D. The acquired order is used in the subsequent step where per-simplex the terms are introduced one-by-one to a single simplex model structure. The MSE of the validation set is monitored after a term is introduced and compared to the MSE of the previous step. When the MSE increases after a term is introduced the process is stopped. An increasing MSE indicates the possibility of over-fitting the model, i.e. adding too many terms. Physical Model Structure equality constraints are created for the terms not introduced to model. These constraints make sure the P-coefficients of these terms are equal to zero by hard constraining them during the optimization. The removal of the physical terms is only possible after optimization in order to prevent breaking the B-net structure.

Physical information of the Cessna Citation II aircraft is introduced to the model as physical bounded inequality constraints. The physical information is based on the stability derivatives acquired by the Citation II stability characteristics and earlier identified aerodynamic models. In Eq. 76 the bounds utilized for this experiment are given. Preliminary results indicate that the bound for C_{m_q} was too tight and is therefore widened.

$$-2 \leq C_{m_\alpha} \leq 0 \quad , \quad -22 \leq C_{m_q} \leq -1.5 \quad , \quad -3 \leq C_{m_{\delta_e}} \leq -0.5 \tag{76}$$

In following sections the Model Structure equality Constraints are abbreviated to MS constraints and similar the Stability Derivative Physical Bounded inequality constraints are abbreviated to PB constraints. The two mentioned constraints are physical constraints and is what makes the Multivariate Simplex B-Splines a Physical-Splines. In addition to the MSBS models, without constraints other tested models with physical constraints will be: MS constraints and PB and MS constraints combined. The final, best performing physical-splines, model is compared to a MSBS and a simple linear polynomial model.

F. Estimation

After setting the KKT-conditions, the parameters are estimated with the Mehrotra predictor-corrector method. To prevent an infeasible solution only model terms higher than degree one are considered in the MS constraints. Also a maximum number of terms to be constraint was set. When the solution turns out to be infeasible it is decided to deleted terms with coupled variables only. Optimization is performed in barycentric space, hence all physical constraints are transformed with the transformation matrix. The normalized barycentric coordinate system can prevent numerical issues and lowers the regression matrix condition number. The models are compared to each other using their residual Root Mean Squared (RMS) error, the coefficient of determination $R2$, and the maximum residual error found. These statistical metrics are calculated for both the identification set and validation set. Also the percentage of deleted model terms are compared. A stability analysis between the linear model, MSBS and physical-splines is performed based on the per-simplex stability derivative values based on grid points. The grid point method has some similarities to a first order Taylor expansions or linearizion relative to that point. Finally, model slice plots will test the stability and divergent behavior at boundaries and the model output is compared for a global quality assessment. Models including PB constraints will be extrapolated and compared to the non-extrapolated models. The extrapolated simplices are given PB constraints to avoid rank deficiency and continuity constraints in order to maintain the same order of continuity with the interpolated simplices.

VII. Results and Discussion

The results of the experiment are presented with a similar structure as the flow diagram of 1. Each phase discusses and elaborates the influence of the physical-splines on the aerodynamic modeling process. For this experiment multiple physical-splines models are created using the aerodynamic modeling process. For all estimated models the Root Mean Squared (RMS) Error, the coefficient of determination $R2$ and the absolute maximum RMS error is determined for both the identification set and the validation set. The best performing physical-splines model, $S_4^1(1,1,1)$, is selected and analyzed. A thorough analysis is performed by comparing the physical-splines model with a MSBS and simple linear least squares model in the sub section estimation and validation.

A. Flight-Test Data

A pre-selection of the available flight test data includes data where only maneuvers are executed on the elevator. A three dimensional dense triangulation is created, and plotted in a matrix graph, on top of this flight test data in Fig. 3. The diagonal histograms show the number of data points and the distribution of these data points within the domain of a state/ control variable. The dimension of this domain is determined by the maximum and minimum value of the measured data in this dimension. In Fig. 3 erroneous data is obtained for the angle of attack α. Unrealistic large negative values are measured causing a very sparse and large range domain for the angle of attack. Including these wrong measurement in the physical-splines model will create large voids in the triangulation and a poor distributions of the data points between the simplices. The first step in the clustering algorithm is removing all empty simplices, see Fig. 4. The second step, in Fig. 5, is to delete all simplices with a low data density of $\mathcal{X}_{t_j} < 2\hat{d}$. These are clearly the remote data point as indicated by the distribution plots and histograms. A distinct large "main island" of simplices is obtained, see Fig. 6. The majority of the data points are located within the convex hull of this cluster. The domain and data within this cluster will serve the aerodynamic modeling process for physical-splines. The clustering process results in a dataset of 127,102 measurements. This is 98.5% of the initial data before clustering. Finally The dataset is randomly divided in an identification set (60%) and validation set (40%).

B. Initialization

During the initialization the model degree, triangulation density, continuity order and stability derivatives upper and lower bound are set. With the candidates, spline dimension and model degree the B-form basis functions for each simplex can be created. The B-form also has an equivalent physical representation that is obtained with the physical transformation matrix. As an example the physical B-form of the C_m pitching moment model for a degree $d = 4$ model is stated in Eq. 77. Each line indicates a "degree group". The model term coefficients have the letter m as subscript combined with the model variable degrees of that specific term. Each model term is numbered for clarification and

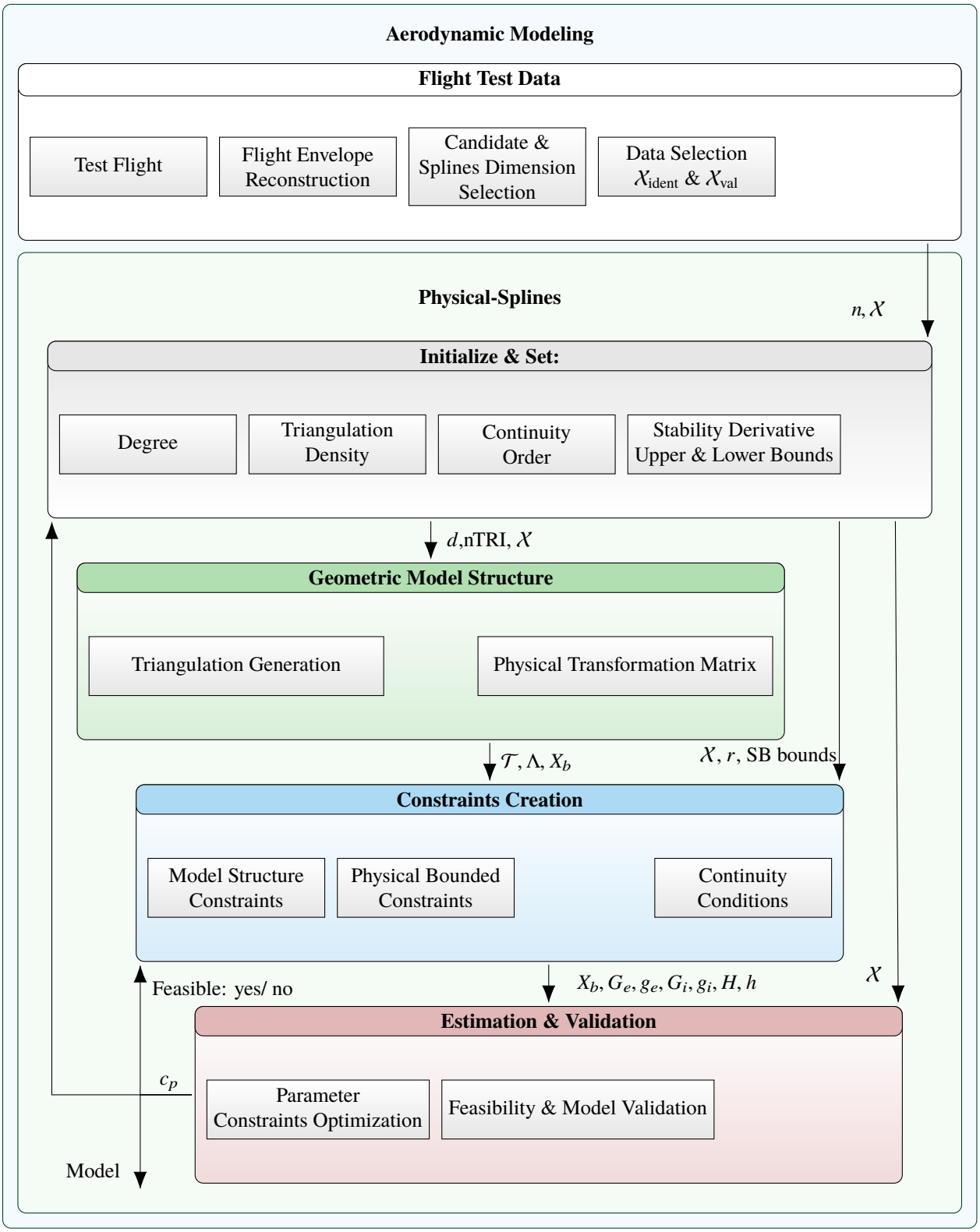

Aerodynamic Modeling
Flight Test Data
Test Flight
Flight Envelope Reconstruction
Candidate & Splines Dimension Selection
Data Selection $\mathcal{X}_{\text{ident}}$ & $\mathcal{X}_{\text{val}}$
Physical-Splines
$n, \mathcal{X}$
Initialize & Set:
Degree
Triangulation Density
Continuity Order
Stability Derivative Upper & Lower Bounds
d,nTRI, $\mathcal{X}$
Geometric Model Structure
Triangulation Generation
Physical Transformation Matrix
$\mathcal{T}, \Lambda, X_b$
$\mathcal{X}$, r, SB bounds
Constraints Creation
Model Structure Constraints
Physical Bounded Constraints
Continuity Conditions
Feasible: yes/ no
$X_b, G_e, g_e, G_i, g_i, H, h$
$\mathcal{X}$
Estimation & Validation
c_p
Parameter Constraints Optimization
Feasibility & Model Validation
Model

Fig. 1

uncoupled model terms are boxed.

$$
\begin{aligned}
Cm(\alpha, q, \delta_e) = \\
&\underset{1}{C_{m_0}} + \\
&\underset{2}{\boxed{C_{m_{100}}\alpha}} + \underset{3}{\boxed{C_{m_{010}}q}} + \underset{4}{\boxed{C_{m_{001}}\delta_e}} + \\
&\underset{5}{\boxed{C_{m_{200}}\alpha^2}} + \underset{6}{C_{m_{110}}\alpha q} + \underset{7}{C_{m_{101}}\alpha\delta_e} + \underset{8}{\boxed{C_{m_{020}}q^2}} + \underset{9}{C_{m_{011}}q\delta_e} + \underset{10}{\boxed{C_{m_{002}}\delta_e^2}} + \\
&\underset{11}{\boxed{C_{m_{300}}\alpha^3}} + \underset{12}{C_{m_{210}}\alpha^2 q} + \underset{13}{C_{m_{201}}\alpha^2\delta_e} + \underset{14}{C_{m_{120}}\alpha q^2} + \underset{15}{C_{m_{111}}\alpha q\delta_e} + \underset{16}{C_{m_{102}}\alpha\delta_e^2} + \underset{17}{\boxed{C_{m_{030}}q^3}} + \underset{18}{C_{m_{021}}q^2\delta_e} + \underset{19}{C_{m_{012}}q\delta_e^2} + \underset{20}{\boxed{C_{m_{003}}\delta_e^3}} + \\
&\underset{21}{\boxed{C_{m_{400}}\alpha^4}} + \underset{22}{C_{m_{310}}\alpha^3 q} + \underset{23}{C_{m_{301}}\alpha^3\delta_e} + \underset{24}{C_{m_{220}}\alpha^2 q^2} + \underset{25}{C_{m_{211}}\alpha^2 q\delta_e} + \underset{26}{C_{m_{202}}\alpha^2\delta_e^2} + \underset{27}{C_{m_{130}}\alpha q^3} + \underset{28}{C_{m_{121}}\alpha q^2\delta_e} + \\
&\quad \underset{29}{C_{m_{112}}\alpha q\delta_e^2} + \underset{30}{C_{m_{103}}\alpha\delta_e^3} + \underset{31}{\boxed{C_{m_{040}}q^4}} + \underset{32}{C_{m_{031}}q^3\delta_e} + \underset{33}{C_{m_{022}}q^2\delta_e^2} + \underset{34}{C_{m_{013}}q\delta_e^3} + \underset{35}{\boxed{C_{m_{004}}\delta_e^4}}
\end{aligned}
\tag{77}
$$

In Sec. E a slightly different interpretation of a stability derivative is deemed necessary. Otherwise the well known individual per term coefficients become very difficult to interpreted. Higher degree models will result in many individual physical coefficients for each simplex. This makes the interpretation a tedious process. Also a universal link between different model degrees is missing. A better and more appropriate approach of interpreting the physical-splines coefficients is obtained by taking the splines model partial derivative with respect to the stability derivative variable. The m^{th} derivative of a physical-splines model, using Eq. 77, is a physical-splines model itself, with degree $d - m$. The model is expressed in the already estimated parameters. This stability derivative splines model can be monitored for different locations within the triangulation domain. The output of these locations can be analyzed and tell something about the aircraft stability and characteristics for these parts of the flight envelope. This way a true generalized and easy to interpreted physical interpretation is given to physical-splines. An example of a stability derivative physical-splines model is given in Eq. 78. The first order partial derivative with respect to the angle of attack α is calculated yielding the stability derivative C_{m_α}. Note that all physical-coefficients shift up, depending on the derivative order.

$$
\begin{aligned}
C_{m_\alpha}(\alpha, q, \delta_e) = \frac{\partial Cm}{\partial \alpha} = \\
&\underset{1}{C_{m_{100}}} + \\
&2\underset{2}{C_{m_{200}}}\alpha + \underset{3}{\boxed{C_{m_{110}}q}} + \underset{4}{\boxed{C_{m_{101}}\delta_e}} + \\
&\underset{5}{\boxed{3C_{m_{300}}\alpha^2}} + 2\underset{6}{C_{m_{210}}}\alpha q + 2\underset{7}{C_{m_{201}}}\alpha\delta_e + \underset{8}{\boxed{C_{m_{120}}q^2}} + \underset{9}{C_{m_{111}}}q\delta_e + \underset{10}{\boxed{C_{m_{102}}\delta_e^2}} + \\
&\underset{11}{\boxed{4C_{m_{400}}\alpha^3}} + 3\underset{12}{C_{m_{310}}}\alpha^2 q + 3\underset{13}{C_{m_{301}}}\alpha^2\delta_e + 2\underset{14}{C_{m_{220}}}\alpha q^2 + 2\underset{15}{C_{m_{211}}}\alpha q\delta_e + 2\underset{16}{C_{m_{202}}}\alpha\delta_e^2 + \underset{17}{\boxed{C_{m_{130}}q^3}} + \underset{18}{C_{m_{121}}}q^2\delta_e + \underset{19}{C_{m_{112}}}q\delta_e^2 + \underset{20}{\boxed{C_{m_{103}}\delta_e^3}}
\end{aligned}
\tag{78}
$$

C. Geometric Model Structure

1. Triangulation

Remote areas in the triangulation show poor data distribution and (near) empty simplices, see Fig. 6. Insufficient data coverage in the hypercube triangulation domain makes it necessary to extrapolate the model. In Tab. 2 the percentage of simplices to be extrapolated, in combination with the flight test data, per triangulation and degree is tabulated. Increasing the triangulation density and degree clearly shows an increase in the percentage of simplices to be extrapolated. The $S_4^1(2,2,2)$ model has 36 extrapolated simplices. This is 25% on a total of 48 simplices. The simplices are extrapolated at the same time when the physical-splines model is optimized. One can conclude from Tab. 6 that the effect of the extrapolated simplices on the interpolated simplices is minimal.

d = 4	**d = 2**	**d = 3**	**d = 4**	**d = 5**
$\mathcal{T}(\mathbf{1},\mathbf{1},\mathbf{1})$	0%	0%	0%	0%
$\mathcal{T}(\mathbf{2},\mathbf{2},\mathbf{2})$	17%	23%	25%	29%
$\mathcal{T}(\mathbf{3},\mathbf{3},\mathbf{3})$	36%	48%	52%	60%

Table 2 Percentage of simplices that require extrapolation.

Extrapolated simplices can also include measurements. They are deleted, because the number of measurements was too low. Table 3 lists the percentage of data points in extrapolated simplices. In contrast to the number of deleted simplices this is minimal. Hence the deleted simplices almost hold no data using the flight test dataset.

	d = 2	**d = 3**	**d = 4**	**d = 5**
$\mathcal{T}(\mathbf{1},\mathbf{1},\mathbf{1})$	0%	0%	0%	0%
$\mathcal{T}(\mathbf{2},\mathbf{2},\mathbf{2})$	0%	0%	0%	1%
$\mathcal{T}(\mathbf{3},\mathbf{3},\mathbf{3})$	0%	1%	2%	3%

Table 3 Percentage of deleted data points as a result of (near) empty simplices

.

2. Physical Transformation

Convolution can be implemented in different manners. For this publication, a script was build that creates the linear physical transformation matrix described in Sec. B. Instead of using a DFT matrix, convolution is performed by multiplying entries while shifting over two vectors. This type of convolution is more numerical stable compared to the DFT. The DFT matrix can yield complex remainders due to rounding errors. The different algorithms are tested in terms of computational time required while determining the transformation for different dimensions and degrees. The first "algorithm 1" is the algorithm in use, at the moment of writing, in the MSBS toolbox. "Algorithm 2" is an algorithm that uses the available de Casteljau algorithm to determine degree reduction matrices. "Algorithm 3" uses convolution in combination with a Kronecker Substitution to determine the physical transformation matrix. The results are tabulated in table 4. The results show that all algorithms perform well for lower degrees. For higher degrees However, there is a significant difference in computational time. The convolution algorithm is much faster and creates a $n = 4$ and $d = 6$ transformation matrix in 2.1 seconds. The current algorithm takes 156 seconds to calculate the same matrix. Overall the best performance is given by algorithm 3 using convolution. Further improvement is possible by creating an algorithm that can convolute sparse vectors. High dimensions Kronecker Substituted vectors become very sparse and many multiplications during convolution can be skipped.

Table 4 Difference between the current algorithm 1 determining the physical transformation matrix and algorithm 2 and the new algorithm 3 using convolution.

	d=2	d=3	d=4	d=5	d=6
Algorithm 1 , n=2	0.0074 s	0.0026 s	0.0076 s	0.0240 s	0.0841 s
Algorithm 2 , n=2	0.0109 s	0.0181 s	0.0496 s	0.1027 s	0.2189 s
Algorithm 3 - n=2	0.0031 s	0.0012 s	9.15×10^{-4} s	0.0012 s	0.0019 s
Algorithm 1 , n=3	0.0025 s	0.0017 s	0.1103 s	0.6882 s	4.1054 s
Algorithm 2 , n=3	0.0170 s	0.1069 s	0.4209 s	1.3376 s	3.5667 s
Algorithm 3 , n=3	8.38×10^{-4} s	0.0018 s	0.0044 s	0.0098 s	0.0258 s
Algorithm 1 , n=4	0.0194 s	0.2068 s	2.0822 s	18.5848 s	156.1977 s
Algorithm 2 , n=4	0.1215 s	1.0856 s	5.8026 s	23.6154 s	83.0631 s
Algorithm 3 , n=4	0.0062 s	0.0352 s	0.1521 s	0.5881 s	2.1196 s

D. Physical Constraints Creation

1. Model Structure Constraints

For the normalized dataset and triangulation the optimal order of appearance is determined by calculating the error reduction ratio of individual terms per simplex. The order of appearance in combination with an stepwise orthogonalization process resulted in insignificant terms to be hard constraint from the physical-splines model. In the histogram of Fig. 7 is the percentage of each term given, indicating how often they are constraint to zero, i.e. are deleted from the physical-splines model afterwards. The color blue is assigned to linear terms, yellow to second order terms, green third order terms and red fourth order terms. The darker colors indicated coupled model terms. The model term relates to a number stated in Eq. 77. The terms hard constraint the most often are

$$\underset{14}{C_{m_{120}}\alpha q^2}\ ,\ \underset{31}{C_{m_{040}}q^4}\ ,\ \underset{24}{C_{m_{220}}\alpha^2 q^2}\ ,\ \underset{27}{C_{m_{130}}\alpha q^3}\ ,\ \underset{25}{C_{m_{211}}\alpha^2 q\delta_e}\ ,\ \underset{23}{C_{m_{301}}\alpha^3\delta_e}\ ,\ \underset{15}{C_{m_{111}}\alpha q\delta_e}\ ,\ \underset{6}{C_{m_{110}}\alpha q}. \tag{79}$$

The majority of the terms in Eq. 79 are coupled model terms and higher order model terms that includes the pitch rate q and angle of attack α. The terms the least hard constraint are

$$\underset{1}{C_{m_0}}\ ,\ \underset{7}{C_{m_{101}}\alpha\delta_e}\ ,\ \underset{20}{C_{m_{003}}\delta_e^3}\ ,\ \underset{35}{C_{m_{004}}\delta_e^4}\ ,\ \underset{21}{C_{m_{400}}\alpha^4}\ ,\ \underset{10}{C_{m_{002}}\delta_e^2}\ ,\ \underset{2}{C_{m_{100}}\alpha}\ ,\ \underset{4}{C_{m_{001}}\delta_e}\ ,\ \underset{5}{C_{m_{200}}\alpha^2}. \tag{80}$$

All of the terms, except number 7, are decoupled model terms. The decoupled model terms are the most meaningful to the physical model. They often have the highest significance in increasing the approximation power and the prevention in over fitting. In figure 8 the percentage of deleted model terms via hard constraints is plotted for different physical-splines models in multiple plots for different degrees. Each column plots a triangulation density and each row different order of continuity conditions. A diamond symbol indicated when coupled terms are hard constraints only. A distinct increase of percentage deleted terms is observed for higher triangulation densities and degrees. For the $\mathcal{T}(1,1,1)$ triangulation very few of the terms are deleted. Interesting for this triangulation is the higher percentage of deleted terms for degree three. Apparently the C_m model is not fitted well with third order terms. Also despite the percentage of deleted physical terms the physical-splines model yields slightly better statistical metrics compared to the MSBS model and therefore prevents over fitting the model when higher degrees are selected.

2. Physical Bounded Constraints

The PB constraints are created on a grid with the density equal to the degree, i.e. the grid points Cartesian coordinates is equivalent to the spatial locations of the B-coefficients.

Table 5 The percentage of stability derivatives out of bound including the percentage of these out of bound stability derivatives located on the boundary.

	Model d = 4, $\mathcal{T}(2,2,2)$, C^1	Out of Bound	Out of Bound on Boundary
C_{m_α}	**MSBS**	48%	91%
	MS Constraints	29%	93%
C_{m_q}	**MSBS**	84%	84%
	MS Constraints	76%	84%
$C_{m_{\delta_e}}$	**MSBS**	59%	91%
	MS Constraints	49%	98%

The PB constraints have a positive performance on the identification process. They remove divergent behavior and deal very well with sparse and poorly distributed data. A slice plot confirms the extreme divergent behavior present in MSBS, Fig. 18. The same slice plot is created for a physical-splines model, Fig. 20. PB constraints completely removes the divergent behavior in the triangulation.

A stability analysis is performed using the stability derivatives of C_{m_α}, C_{m_q}, and $C_{m_{\delta_e}}$ of the C_m moment model to asses the effect of the physical MS and PB constraints on the physical-splines model. For each of the stated stability derivatives the outcome is plotted respectively in Figures 9, 10, and 11. Each figure represents the output of a stability derivative on multiple grid points per-simplex, for the $S_4^1(2,2,2)$ MSBS model without constraints and Physical-Splines models with MS, and MS and PB constraints combined. In these plots divergent behavior of the stability derivatives become apparent. The figures clearly show a divergent behavior for the models without PB constraints. For example in Fig. 9 poor performing simplices include simplex number 9, 26, and 30. However, MS constraints indicate some over all improvement in reducing the divergence. The physical bound set for C_{m_α} is appropriate. The bottom graph in Fig. 9 shows that most of the stability derivative points stay well within the bound, especially the lower bound. This is also true for stability derivative $C_{m_{\delta_e}}$ in Figure 11. Stability derivative C_{m_q} is more troublesome. The bottom graph in 10, indicates that most of the simplices have stability derivatives located on the upper and lower bounds. This can be caused by numerical errors, since the measured values for q are relatively small.

In order to identify the location and the simplices containing out of bound, and therefor divergent, stability derivatives the locations and values of stability derivatives is plotted in the three dimensional ($n = 3$) triangulation. The stability derivatives are determined for the $S_4^1(2,2,2)$ MSBS model without constraints. This plot is created for each stability derivative, C_{m_α}, C_{m_q}, and $C_{m_{\delta_e}}$, in Figures 12, 13, and 14 respectively. The green color indicates that a stability derivative at that specific location is within its preset bound, the color red indicates that the stability derivative is outside the bound. Figure 13 confirms the poor performing stability derivative C_{m_q}. For the other two stability derivatives C_{m_α} and $C_{m_{\delta_e}}$, in Figures 12 and 13 respectively, the poor performing parts of the triangulation are clearly the boundary areas. This divergent behavior at boundaries is also known as Runge's Phenomenon. Oscillation/ divergent behavior occur at the boundary of the domain for higher degree polynomials. This effect is also amplified by the continuity constraints where this behavior of other simplices are propagated to the boundaries of the triangulation. Table 5 summarizes for the $S_4^1(2,2,2)$ models the percentage of stability derivative grid points outside the bound set and what percentage of these points are located on the boundary of the triangulation. For C_{m_α} and $C_{m_{\delta_e}}$ these results clearly confirm that the majority of the out of bound points are located on the triangulation boundary.

Overall one can conclude that the PB constraints performance is very good. They remove the divergent effect of the tested stability derivatives completely and play an important role in extrapolation.

E. Estimation and Validation

The model with the highest statistical quality assessments are the models with $\mathcal{T}(2,2,2)$ triangulation. The best performing physical-splines model is $S_4^1(2,2,2)$. The calculation time for the physical model equals 1 second. The extrapolated model required 5.6 seconds and includes approximately 10,000 PB constraints to the optimization. This model is compared to the MSBS (with equal degree, triangulation and continuity) and a simple linear model.

The statistical metrics of the pitching moment C_m for all three different types of modeling techniques, in Tab. 6 show

Type	Constraints	RMS(ϵ)	R2	max(|RMS(ϵ)|)
Linear Model	-	8.3×10^{-3}	0.83	0.09
$S_4^1(2,2,2)$ MSBS	-	7×10^{-3}	0.88	0.06
$S_4^1(2,2,2)$ Physical-Splines	**MS**	6.4×10^{-3}	0.89	0.06
	MS and PB	6.7×10^{-3}	0.88	0.08
	MS and PB Extrapolated	6.9×10^{-3}	0.88	0.08

Table 6 Validation results of models with $S_4^1(2,2,2)$ property. The MSBS model is compared to the Physical-Splines model and the extrapolated Physical-Splines models.

Linear LS	$\bar{\theta}$
C_{m_0}	-0.04
C_{m_α}	-0.40
C_{m_q}	-10.30
$C_{m_{\delta_e}}$	-1.06
$C_{m_{M_g}}$	0.17

Table 7 Stability derivative coefficients statistics for the simple linear leasts squares model.

adequate outcomes. The MSBS and physical-splines models yield better results than the linear LS model, especially for the coefficient of determination. The MS constraints has the capability to prevent over-fitting of higher degree models by hard constraining insignificant physical model terms from the physical-splines model. This is confirmed by the lower RMS validation error even tough the number of terms is decreased by 15% compared to the MSBS model. The PB constraints and extrapolation slightly increases the RMS error, but is still less compared to the linear LS and MSBS model.

The average stability derivatives of the C_m moment model are tabulated in 7, 8, and 9 for the linear LS, MSBS and physical-splines models respectively. The B-coefficient of the MSBS are transformed to the physical space for comparison only. No other constraints were applied to the MSBS model. The splines models include the minimum and maximum stability derivatives obtained in the interpolated triangulation domain and the corresponding mean and standard deviations of the these stability derivatives. Again the immense divergent behavior on the boundaries in the MSBS model reveals itself. The obtained values in Tab. 8 are very unrealistic and deprive the possibility to physically interpreted the stability derivatives at all. The physical-splines show a much more stable and robust behavior. They outperform the MSBS model in realism and physical interpretation. The linear LS model, Tab. 7, compared to the physical-splines model reassembles similar outcomes. However, the difference in stability derivative mean of C_{m_α} and the standard deviation of the physical-splines model, Tab. 9, hints to possible non-linearities in the flight test data and therefore the physical-splines model.

These non-linearities are confirmed by the slice plots fixing two of the three dimensions. In these figures the linear LS model, the physical-splines model, and the validation data is plotted in blue, green, and orange respectively. Especially in Fig. 15 the angle of attack validation data and physical-splines model demonstrated non-linear behavior, between 0.08 and 0.12 radians, obviously missed by the linear model. Figure 16, plotting the pitch rate q, indicates on some non-linear behavior near the domain boundaries. The presence of the simplex splines is noticeable, the physical-splines model clearly follows the trend of local data points. Whereas the linear model identifies global trends. The elevator deflection δ_e in Fig. 17 show the control variable has an evident linear influence on the pitching moment in the measured part of the flight envelope. The output of the linear model, physical-splines mode, and validation data are close together.

A two dimensional slice plot of the angle of attack, elevator deflection and model output C_m calculated with the linear model is given by Fig. 19 and the physical-splines model by Fig. 20. The latter figure is characterized by the small amount of non-linearities. To visualize the actual non-linearities the two graphs are subtracted from each other to concede the differences, see Fig. 21. The deficiency of the linear model is very slim and mostly exist in more extreme, i.e. larger values of the angle of attack and elevator deflection, regions of the flight envelope.

MSBS	$\bar{\theta}$	$\theta_{\min}$	$\theta_{\max}$	$\sigma(\theta)$
C_{m_0}	9.38	-18.16	128.36	29.96
C_{m_α}	0.91	-857.35	556.99	52.55
C_{m_q}	57.94	-1.16×10^4	2.23×10^4	1.43×10^3
$C_{m_{\delta e}}$	7.14	-907.46	4.42×10^3	178.57
$C_{m_{M_g}}$	0.11	0.11	0.11	0

Table 8 Stability derivative coefficients statistics for the MSBS model $S_4^1(2, 2, 2)$.

Physical-Splines	$\bar{\theta}$	$\theta_{\min}$	$\theta_{\max}$	$\sigma(\theta)$
C_{m_0}	8×10^{-3}	0.16	-0.72	0.14
C_{m_α}	-0.60	-1.95	-0.02	0.4
C_{m_q}	-10.59	-21.91	-1.52	5.89
$C_{m_{\delta e}}$	-1.24	-3.00	-0.51	0.57
$C_{m_{M_g}}$	0.19	0.19	0.19	0

Table 9 Stability derivative coefficients statistics for the extrapolated Physical-Splines model $S_4^1(2, 2, 2)$.

VIII. Conclusion

The advent of MSBS in system identification, and specific advanced aerodynamic modeling, opens up a world of new possibilities. Experiments demonstrate that the Physical-Splines addition is a more robust method for aerodynamic modeling compared to the MSBS. The Physical Bounded inequality constraints introduce a-priori physical information to the optimization process. In combination with the physical Model Structure constraints they reduce the model size, prevent over-fitting and propagation of erroneous data, and remove inherent MSBS divergent behavior at triangulation boundaries. The divergent behavior is mostly caused by the sparsity in the flight-test data, resulting in empty simplices and simplices with poorly distributed data and therefore problems with the estimation of remote B-coefficients. However, too many constraints can make the optimization infeasible. A good practice is to limit the number of terms to be constraint based on the $d - r - 1$ rule. Furthermore, it is preferable not to constraint any linear model terms. If the optimization is found to be infeasible, removing coupled model terms from the constraints will most likely solve the in-feasibility problem. Important is to verify if the bounds set on the PB constraints are not too tight. Creating grid points, for the PB constraints, equal to the B-net density is sufficient to make these constraints work. Increasing the grid density has little effect. Applying continuity constraints and PB constraints to the simplices to be extrapolate makes it possible to add the extrapolation process to the initial optimization process. By doing so, the effect of extrapolation on the interpolated simplices is minimal. This way, a complete hypercube triangulation domain is created, which is much easier to interpreted. The pitching moment C_m model is very linear and applying physical-splines improves the model marginally. However, promising results can be expected for modeling extreme regions of the flight envelope with highly non-linear dynamics.

Appendix

A. Cessna Citation II

Table 10 Aircraft details of PH-LAB

PH-LAB	
Manufacturer	Cessna
Type	Citation II Model 550
Dimensions	
Length	14.39 m
Wingspan b	15.90 m
Wing Area S	30.00 m^2
Wing Chord Length $\bar{c}$	2.06 m
Height	4.57 m
Mass Properties	
Empty Weight	3,906 kg
Maximum Payload	1,400 kg
Maximum Fuel Load	2,250 kg
Maximum Takeoff Weight	6,715 kg
$I_{xx,\text{empty}}$	12,392 kg·m^2
$I_{yy,\text{empty}}$	31,501 kg·m^2
$I_{zz,\text{empty}}$	41,908 kg·m^2
$I_{xz,\text{empty}}$	2,252 kg·m^2
Performance	
Engine	2× Pratt & Whitney JT15D-4 turbofan
Maximum Thrust	2×11.1 kN
Maximum Cruise Speed	710 km/h
Maximum Operating Altitude	13,000 m

Table 11 Instrumentation of the PH-LAB

Parameter	Unit	F_s [Hz]	1σ std	Source
X_b-axis rotation	[rad]	1	8.70×10^{-3}	Sperry Vertical Gyro
Y_b-axis rotation	[rad]	50	8.70×10^{-3}	Sperry Vertical Gyro
Z_b-axis rotation	[rad]	50	1.73×10^{-3}	Gryosyn Compass
X_b-axis rotational rate	[rad/s]	100	2.00×10^{-3}	LITEF μFORS Rate Gyro
Y_b-axis rotational rate	[rad/s]	100	2.00×10^{-3}	LITEF μFORS Rate Gyro
Z_b-axis rotational rate	[rad/s]	100	5.00×10^{-3}	LITEF μFORS Rate Gyro
X_b-axis linear acceleration	$[m/s^2]$	100	2.00×10^{-2}	Q-Flex 3100 Accelerometer
Y_b-axis linear acceleration	$[m/s^2]$	100	2.00×10^{-2}	Q-Flex 3100 Accelerometer
Z_b-axis linear acceleration	$[m/s^2]$	100	3.00×10^{-2}	Q-Flex 3100 Accelerometer
Altitude	[m]	1	3.00×10^{-1}	Static Probe
True Airspeed	[m/s]	50	1.00×10^{-1}	Pitot-Static Probe
Angle of Attack	[rad]	1000	3.50×10^{-3}	Alpha Vane
Angle of Sideslip	[rad]	1000	3.50×10^{-3}	Beta Vane

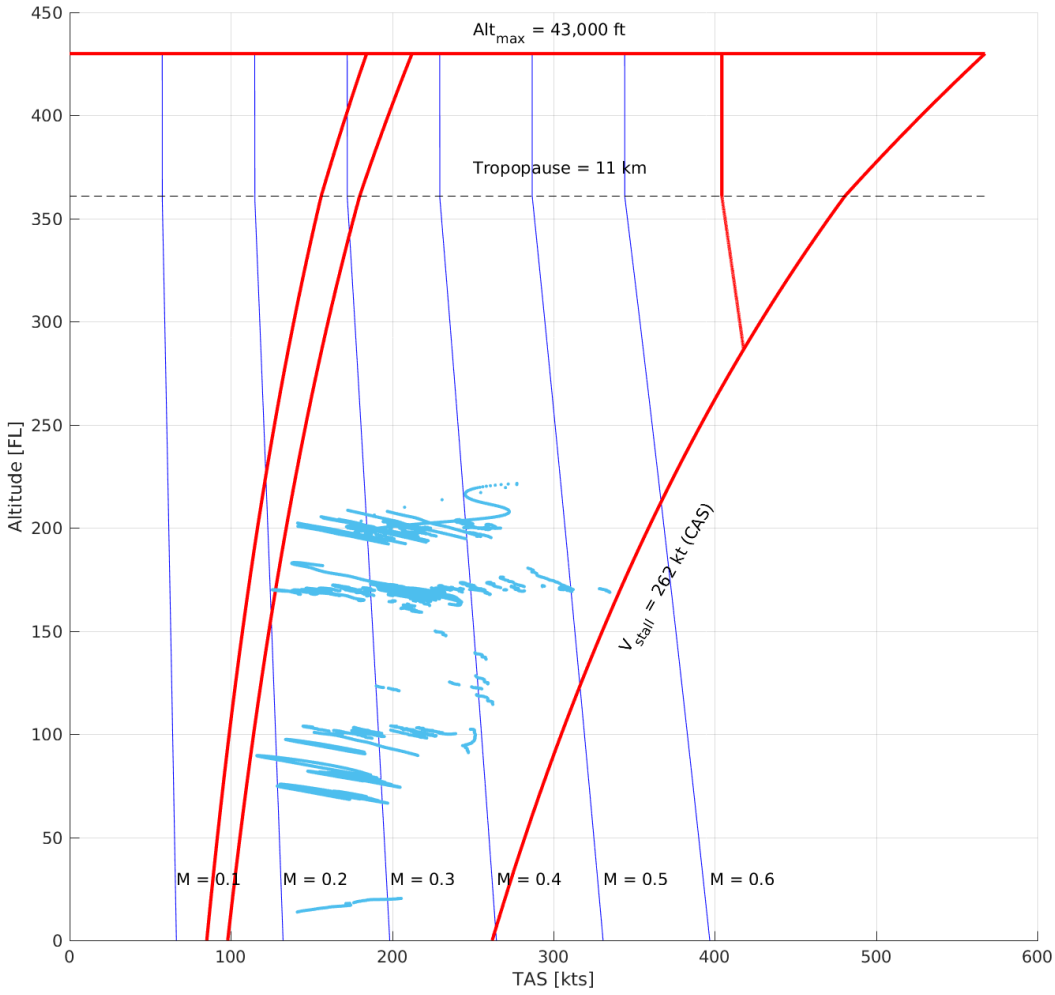


Fig. 2 The Flight Envelope of the Cessna Citation II showing the the TAS and altitude where flight test measurements were performed

B. Flight Test Data

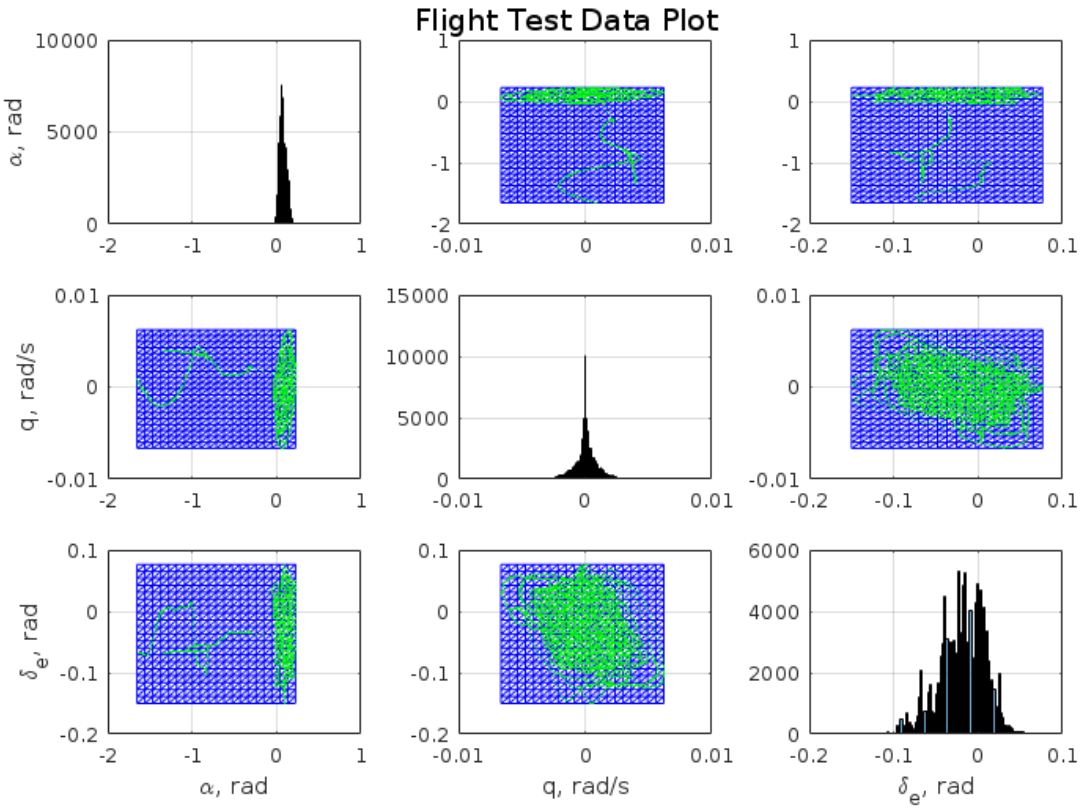


Fig. 3 Clustering of the measured flight test data in a matrix plot. The data distribution is visualized by the green dots. The blue triangulation indicates the domain of the data. The plots on the diagonal are histograms plots of the number of data points.

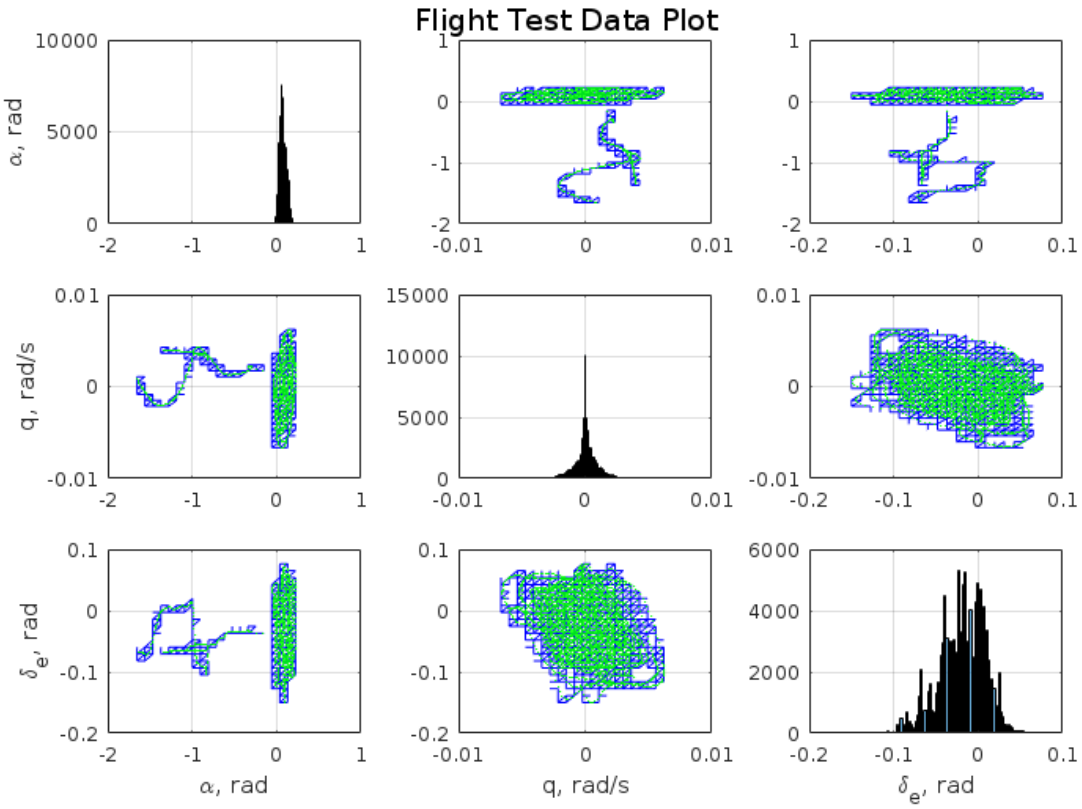


Fig. 4 Cluster plot after deleting the empty simplices in the triangulation.

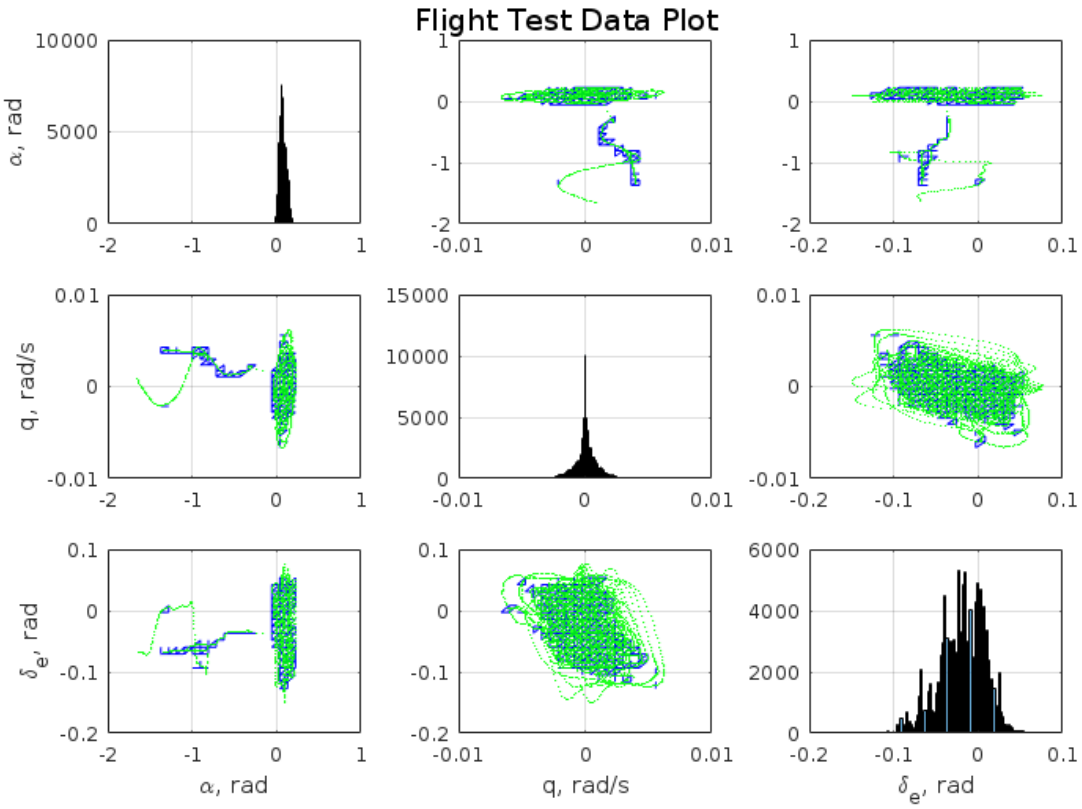


Fig. 5 Cluster plot after deleting near empty simplices from the triangulation.

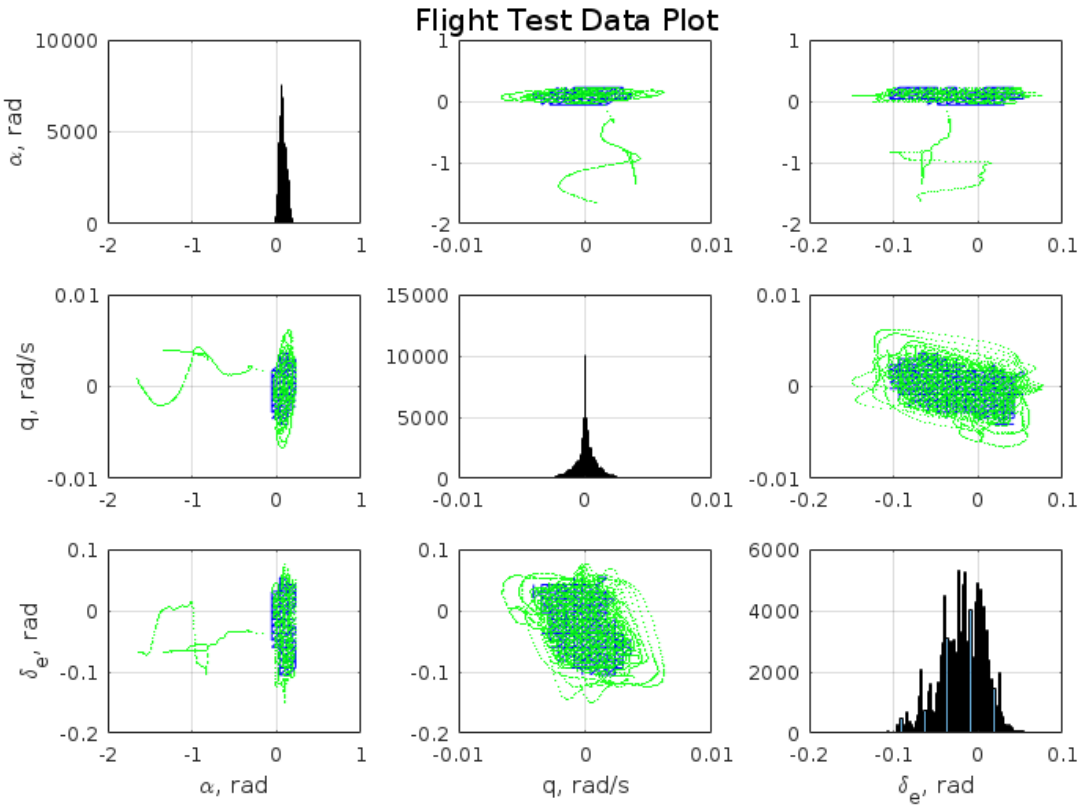


Fig. 6 The largest data triangulation cluster is obtained and data within the convex hull will be used in the aerodynamic modeling process.

C. Model Structure

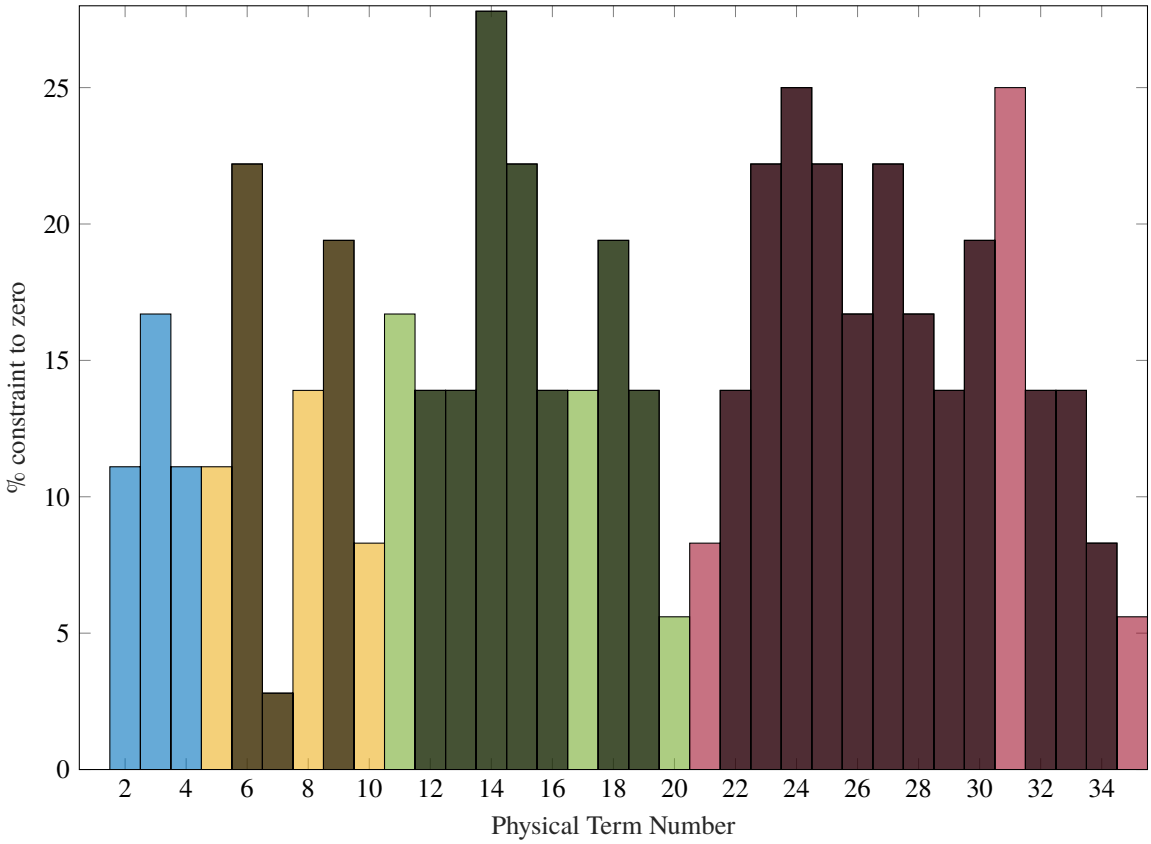
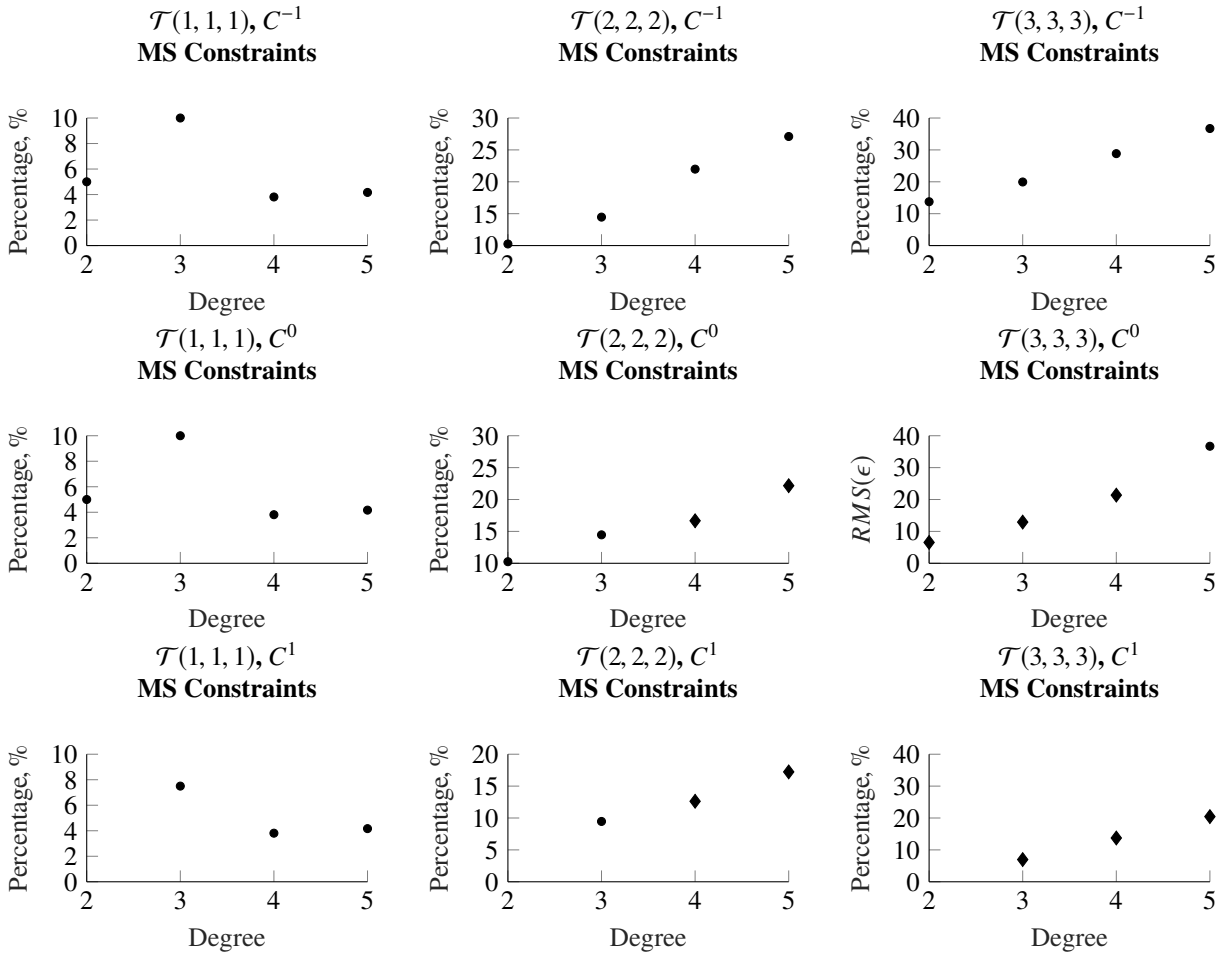


Fig. 7 Histogram of the percentage of individual terms constraint to zero from the physical-splines model for the entire triangulation.

Fig. 8 The percentage of deleted physical model structure terms of the total number of terms. Plotted for triangulations $\mathcal{T}(1,1,1)$, $\mathcal{T}(2,2,2)$, and $\mathcal{T}(3,3,3)$ with different degrees and continuity.



D. Stability Derivatives Analysis Figures

Fig. 9 Per simplex output of the stability derivative C_{m_α} using a grid with data points.

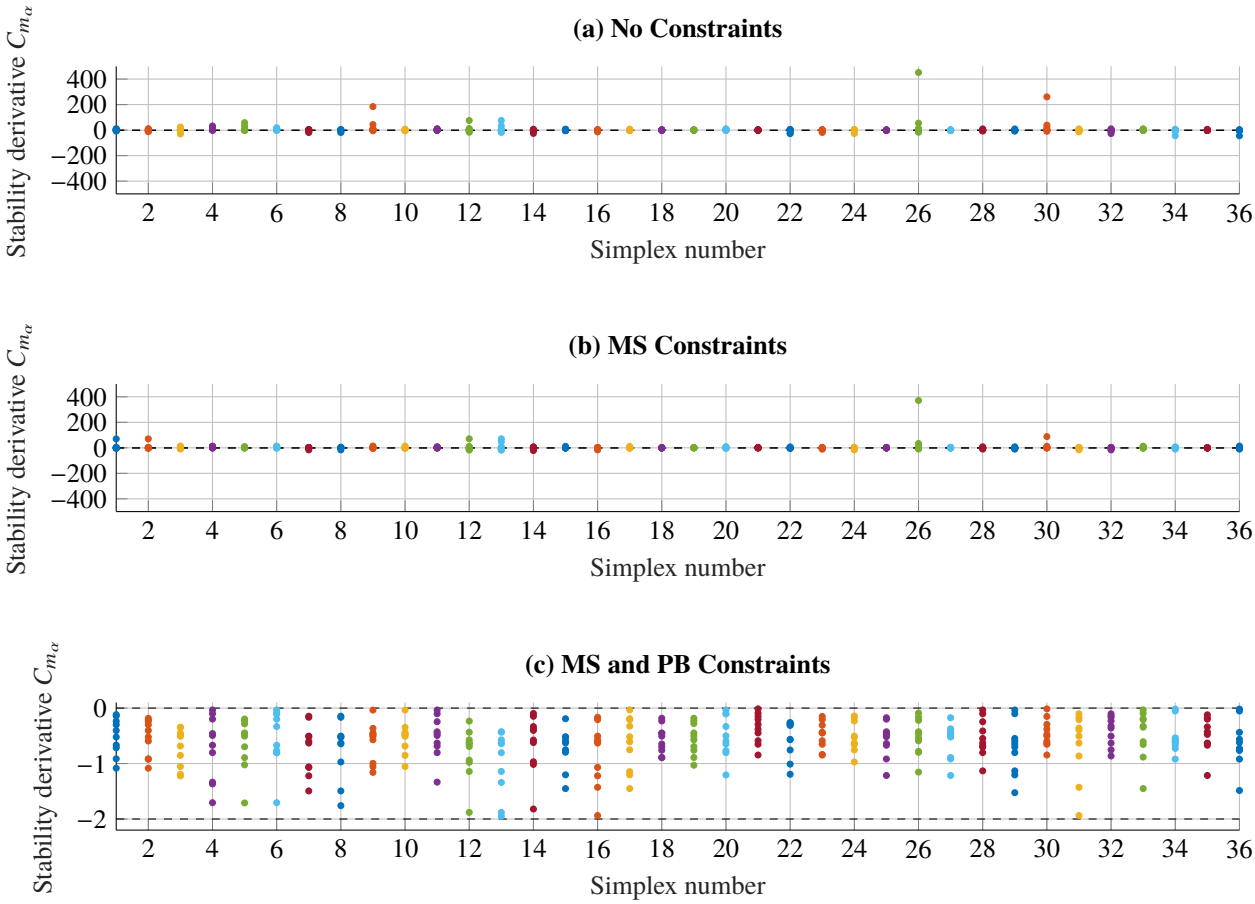


Fig. 10 Per simplex output of the stability derivative C_{m_q} using a grid with data points.

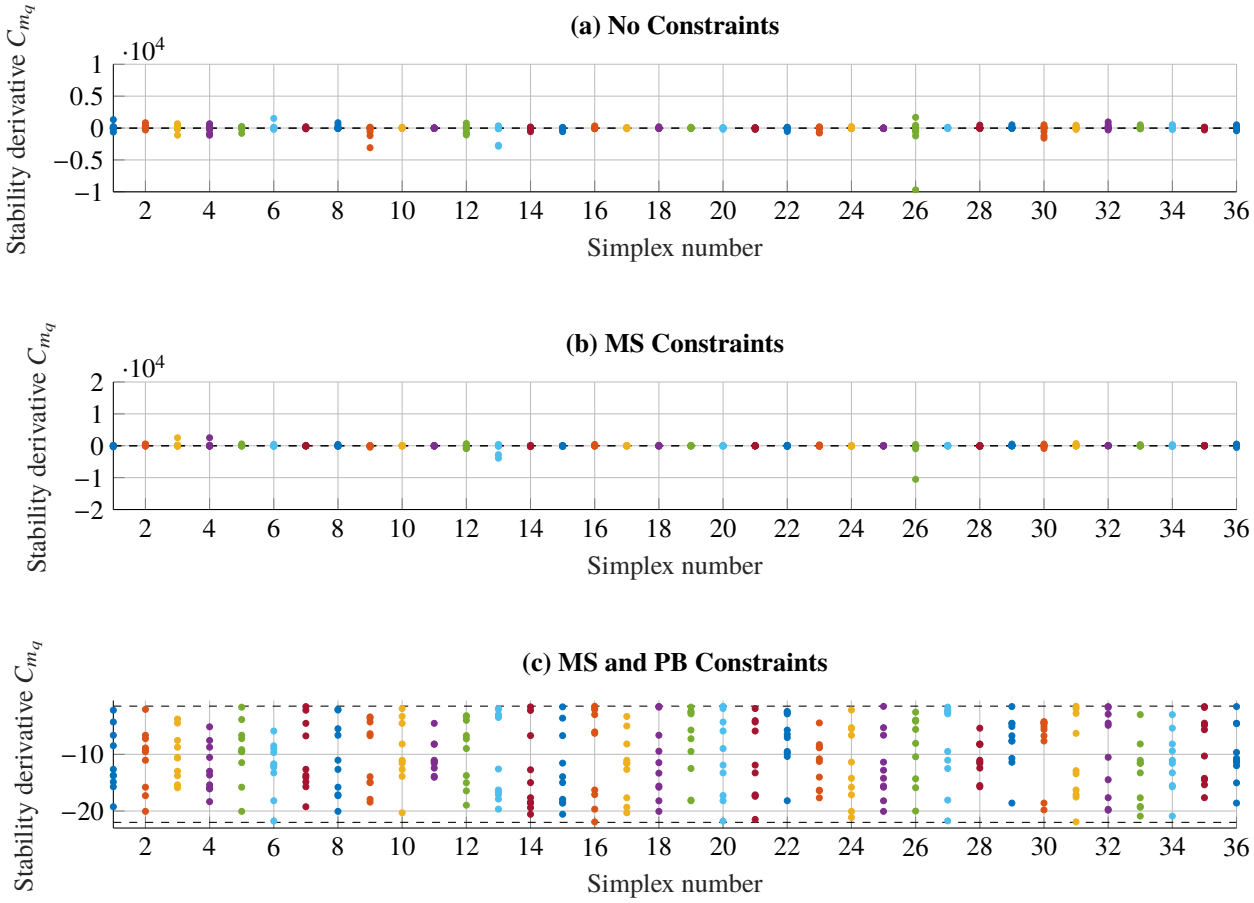
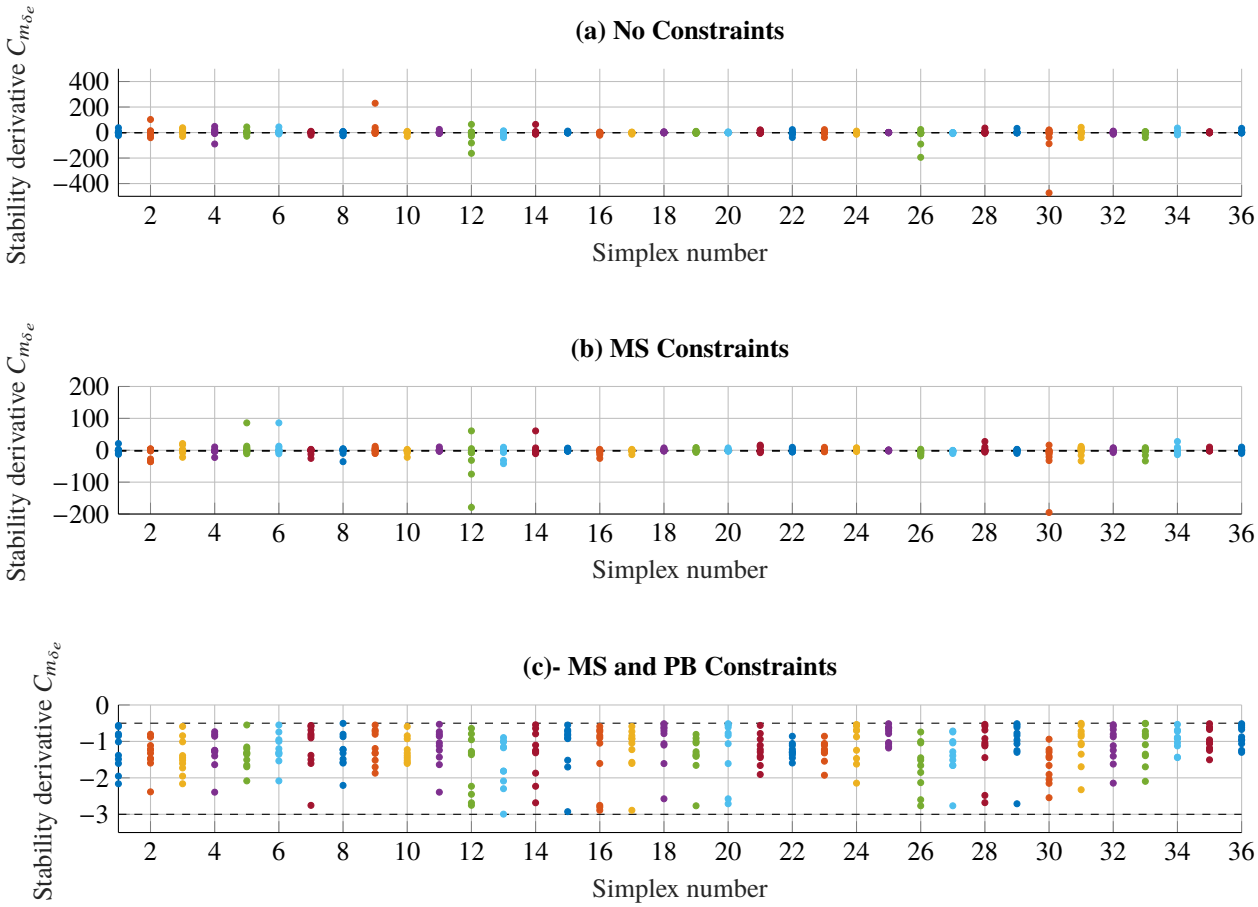


Fig. 11 Per simplex output of the stability derivative $C_{m_{\delta_e}}$ using a grid with data points.



E. Triangulation Stability Derivative Figures

Fig. 12 Triangulation plot of the $S_4^1(2,2,2)$ MSBS model with tested stability derivative C_{m_α} grid points. Green dots indicate that the stability derivative is within the preset bounds, red indicates the stability derivative is out of bound.

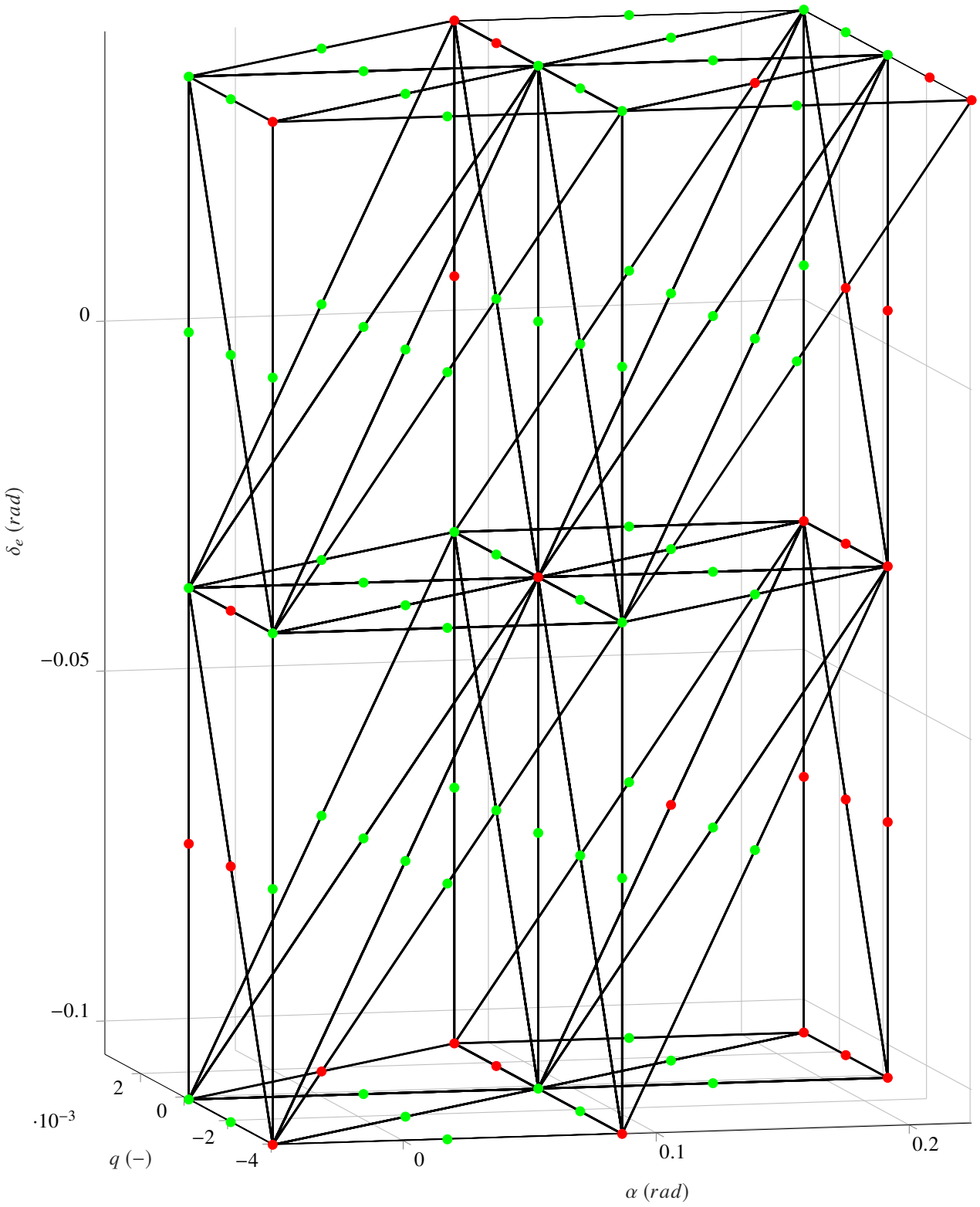


Fig. 13 Triangulation plot of the $S_4^1(2,2,2)$ MSBS model with tested stability derivative C_{m_q} grid points. Green dots indicate that the stability derivative is within the preset bounds, red indicates the stability derivative is out of bound.

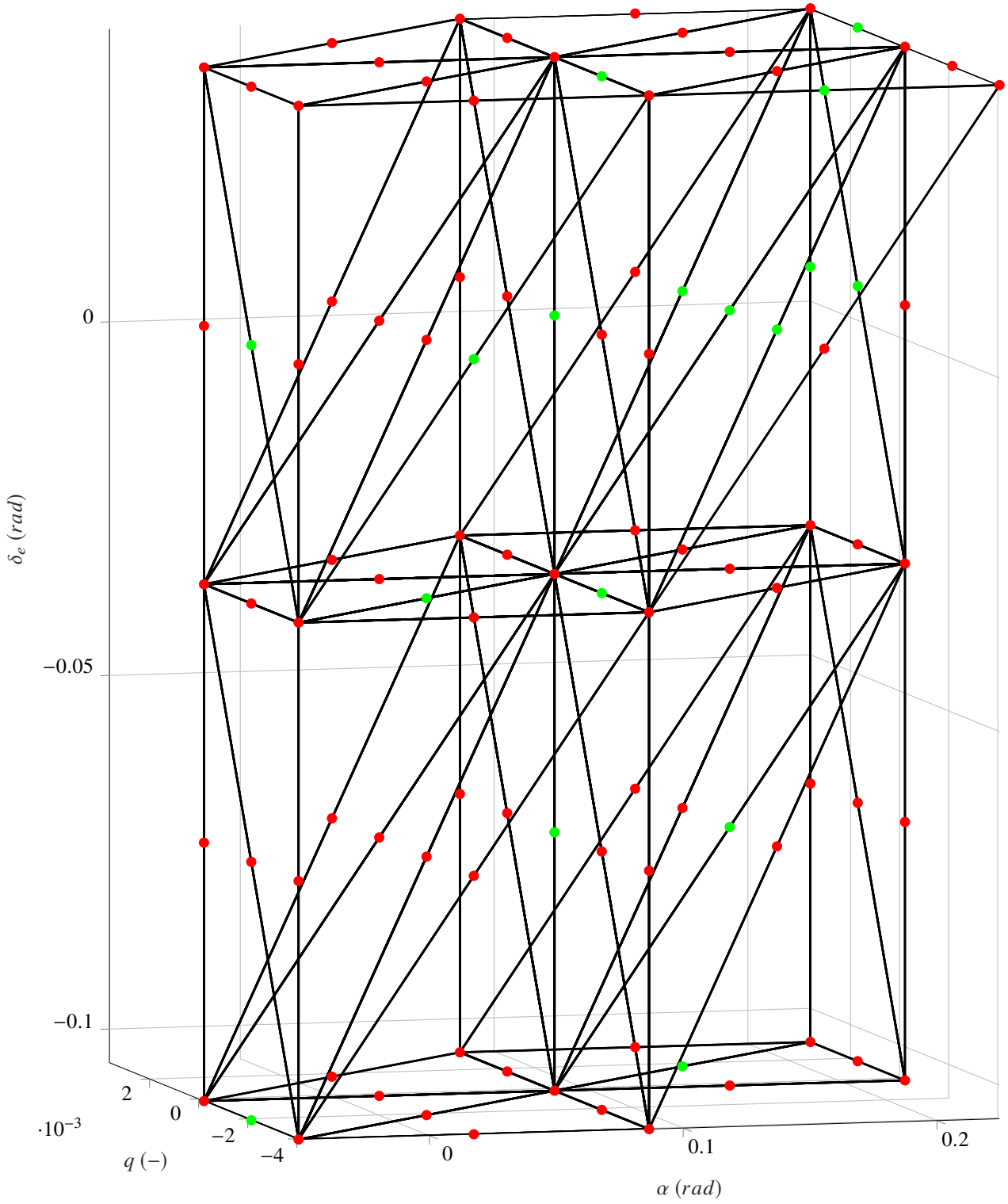
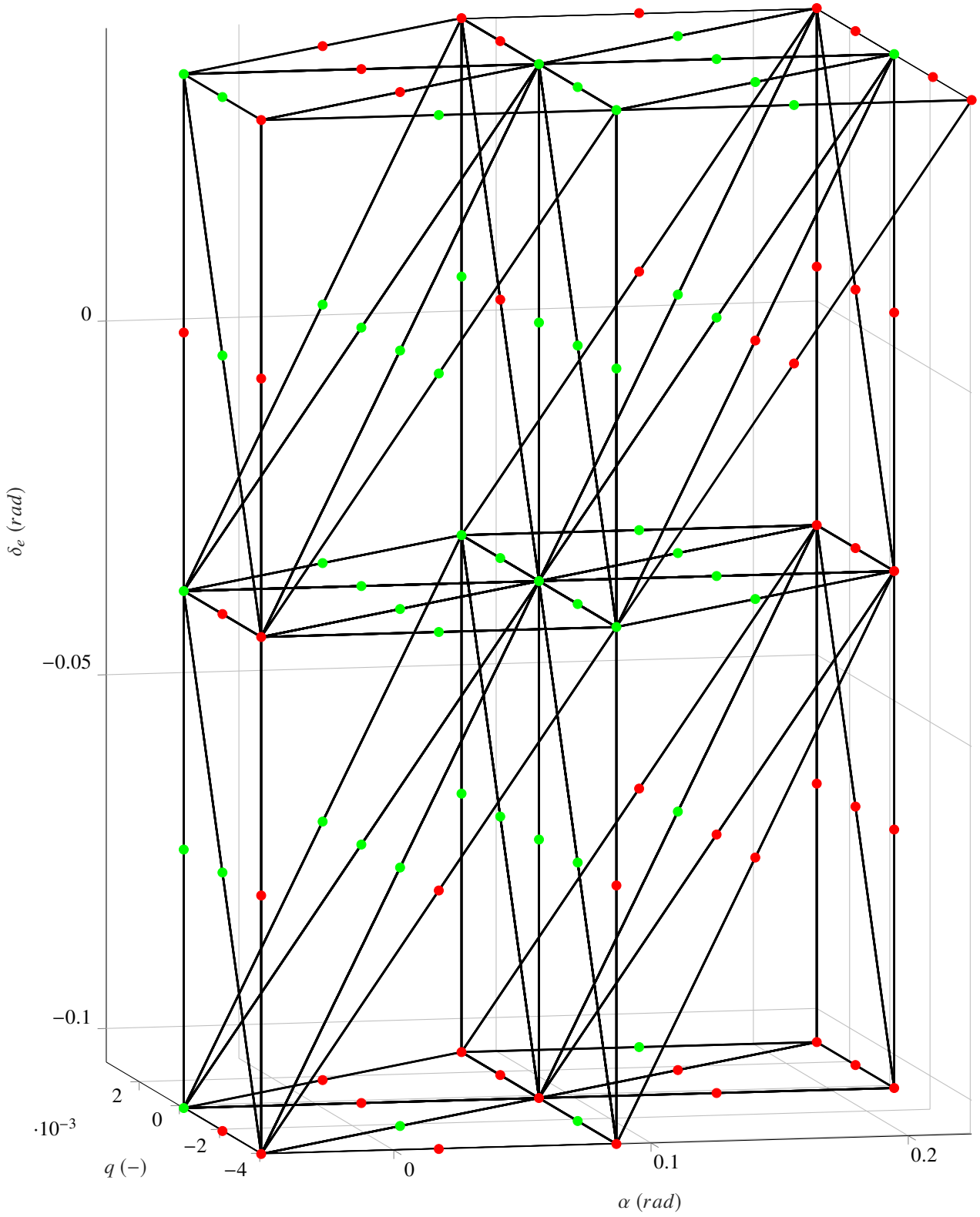


Fig. 14 Triangulation plot of the $S_4^1(2,2,2)$ MSBS model with tested stability derivative $C_{m_{\delta_e}}$ grid points. Green dots indicate that the stability derivative is within the preset bounds, red indicates the stability derivative is out of bound.



F. Slice Plots

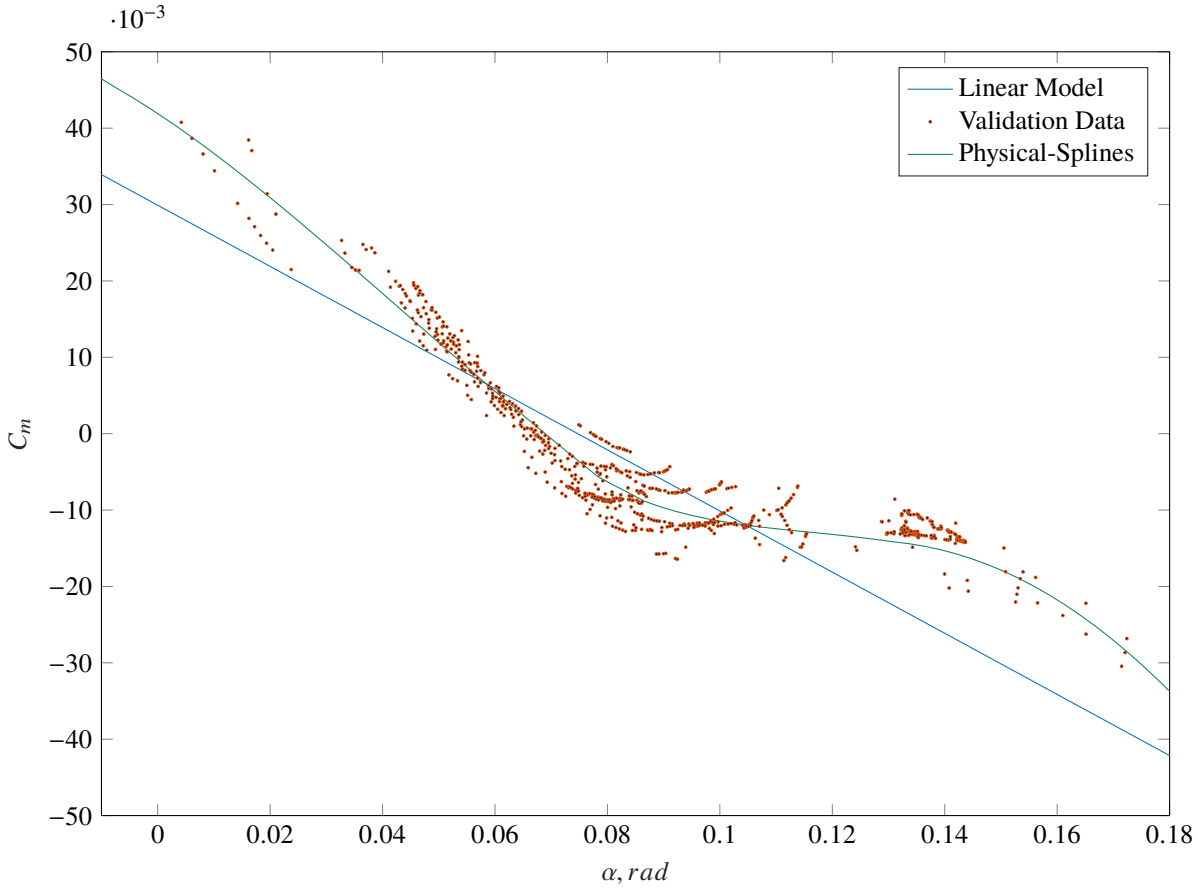


Fig. 15 Angle of attack $\times$ Cm. The blue line plots the output of the simple linear model, the green line the output of the physical-splines $S_4^1(2,2,2)$ model and the orange dots are output of the validation dataset. $q = 1 \times 10^{-3}$ **–,** $\delta_e = -0.03\,rad$ **and** $M = 0.3$**.**

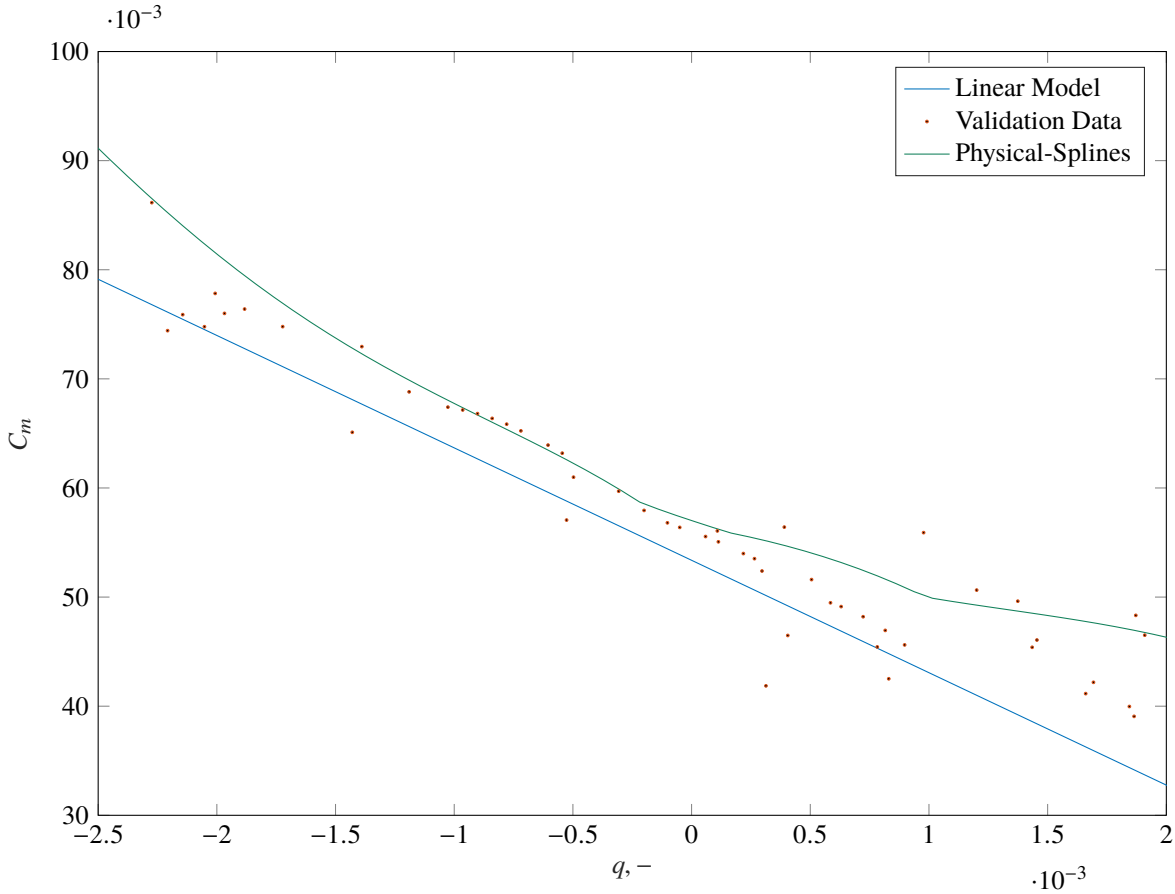


Fig. 16 Pitch rate $\times$ Cm. The blue line plots the output of the simple linear model, the green line the output of the physical-splines $S_4^1(2,2,2)$ model and the orange dots are output of the validation dataset. $\alpha = 0.1\,rad$, $\delta_e = -0.08\,rad$ and $M = 0.3$.

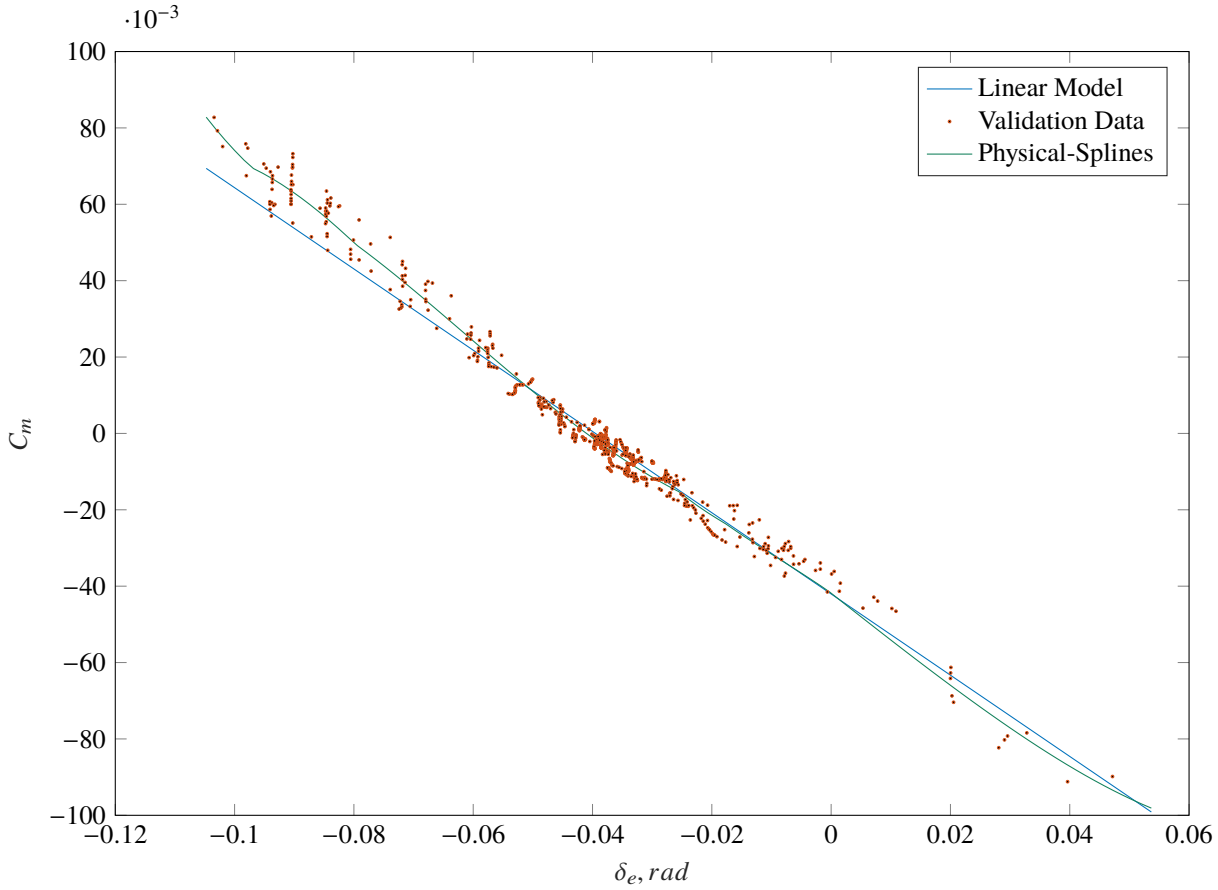


Fig. 17 Elevator deflection $\times$ Cm. The blue line plots the output of the simple linear model, the green line the output of the physical-splines $S_4^1(2,2,2)$ model and the orange dots are output of the validation dataset. $\alpha = 0.1\,rad$, $q = 1 \times 10^{-3}$ –and $M = 0.3$.

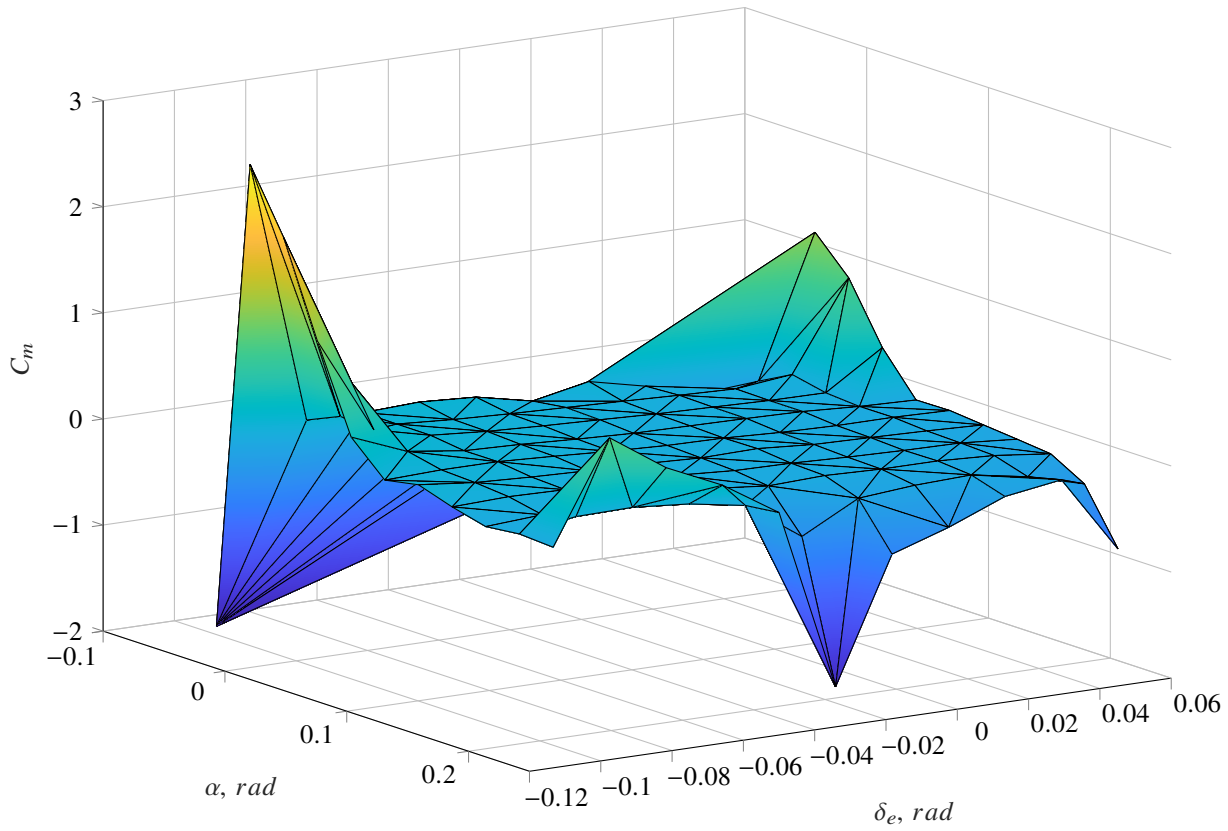


Fig. 18 Slice plot of a MSBS $S_4^1(2,2,2)$, no constraints, angle of attack $\times$ elevator deflection. $q = 0-$ **and** $M = 0.3$**.**

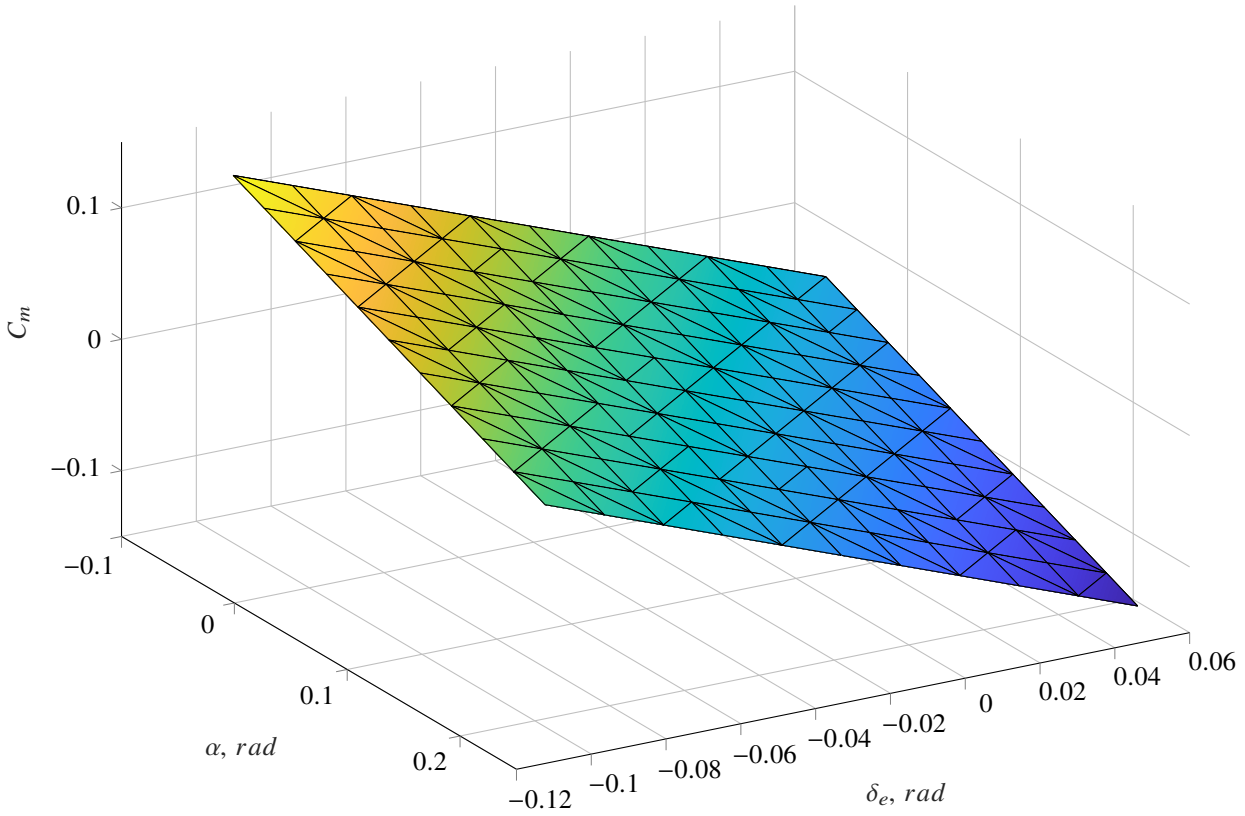


Fig. 19 Slice plot of a simple linear model, angle of attack × elevator deflection. $q = 0-$ **and** $M = 0.3$**.**

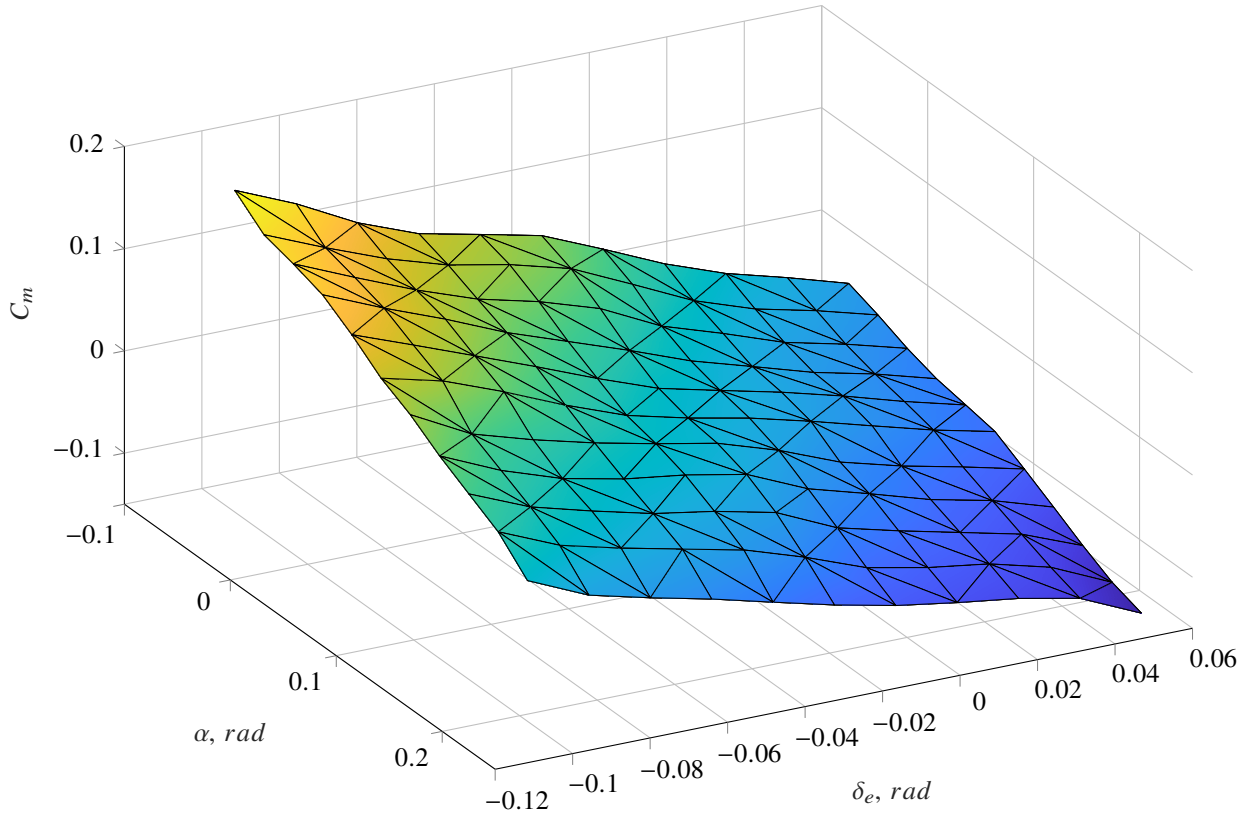


Fig. 20 Slice plot of a physical-splines $S_4^1(2,2,2)$ model with MS and BD constraints, angle of attack $\times$ elevator deflection. $q = 0-$ and $M = 0.3$.

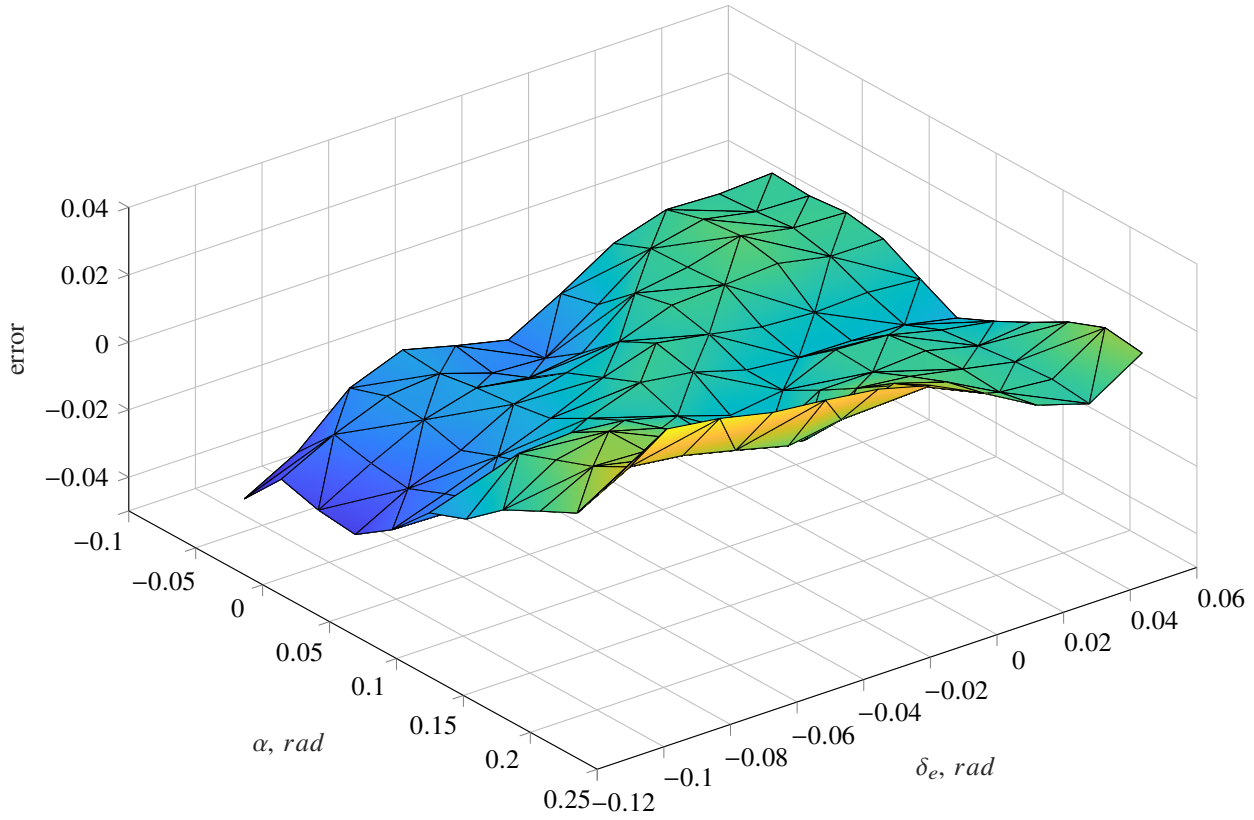


Fig. 21 Slice plot of the difference between the simple linear model and physical-splines $S_4^1(2,2,2)$ model, angle of attack $\times$ elevator deflection. $q = 0-$ **and** $M = 0.3$**.**

References

[1] Administration, F. A., "Qualification, Service, and Use of Crewmembers and Aircraft Dispatchers," Tech. rep., U.S. Department of Transportation, 2013.

[2] van der Linden, C., *DASMAT-Delft University Aircraft Simulation Model and Analysis Tool: A Matlab, Simulink Environment for Flight Dynamics and Control Analysis*, Coronet Books Inc., 1998.

[3] Tol, H., De Visser, C., Van Kampen, E., and Chu, Q., "Nonlinear multivariate spline-based control allocation for high-performance aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1840–1862.

[4] Tol, H. J., Visser, C. C. D., Sun, L. G., Kampen, E. V., and Chu, Q. P., "Multivariate Spline-Based Adaptive Control of High-Performance Aircraft with Aerodynamic Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 4, 2016, pp. 781–800. doi:10.2514/1.G001079.

[5] Leroy, R., "Certificats de positivité et minimisation polynomiale dans la base de Bernstein multivariée," Ph.D. thesis, Université Rennes 1, 2008.

[6] Kuhn, H. W., "Some combinatorial lemmas in topology," *IBM Journal of research and development*, Vol. 4, No. 5, 1960, pp. 518–524.

[7] Kuindersma, S., Permenter, F., and Tedrake, R., "An efficiently solvable quadratic program for stabilizing dynamic locomotion," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 2589–2594.

[8] de Boor, C., "B-form Basics," *Geometric modeling: algorithms and new trends*, edited by G. Farin, SIAM, 1987.

[9] de Visser, C. C., "Global Nonlinear Model Identification with Multivariate Splines," Ph.D. thesis, Delft University of Technology, 2011.

[10] Harvey, D., "Faster polynomial multiplication via multipoint Kronecker substitution," *Journal of Symbolic Computation*, Vol. 44, No. 10, 2009, pp. 1502–1510.

[11] Wang, Z., "Fast algorithms for the discrete W transform and for the discrete Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 32, No. 4, 1984, pp. 803–816.

[12] Winograd, S., "On computing the discrete Fourier transform," *Mathematics of computation*, Vol. 32, No. 141, 1978, pp. 175–199.

[13] Lodha, S., and Warren, J., "Degree reduction of Bézier simplexes," *Computer-Aided Design*, Vol. 26, No. 10, 1994, pp. 735–746.

[14] Mulder, e. a., J.A., *Flight Dynamics*, TU Delft, 2013.

[15] van den Hoek, M. A., de Visser, C. C., and Pool, D. M., "Identification of a Cessna Citation II Model Based on Flight Test Data," *Proceedings of the 4th CEAS Specialist Conference on Guidance, Navigation & Control, Warsaw, Poland*, 2017.

[16] de Visser, C. C., Chu, Q., and Mulder, J., "A new approach to linear regression with multivariate splines," *Automatica*, Vol. 45, No. 12, 2009, pp. 2903–2909.

[17] Klein, V., "Estimation of aircraft aerodynamic parameters from flight data," *Progress in Aerospace Sciences*, Vol. 26, No. 1, 1989, pp. 1–77.

[18] Aitken, A. C., "IV.—On Least Squares and Linear Combination of Observations." *Proceedings of the Royal Society of Edinburgh*, Vol. 55, 1936, pp. 42–48.

[19] Kuhn, H. W., "Nonlinear programming: a historical view," *Traces and emergence of nonlinear programming*, Springer, 2014, pp. 393–414.

[20] Kuhn, H. W., and Tucker, A., "Nonlinear Programming," *Proceedings of the Second Symposium on Mathematical Statistics and Probability*, , No. x, 1951, pp. 481–492. doi:10.1007/BF01582292, URL `http://projecteuclid.org/euclid.bsmsp/1200500249`.

[21] Mehrotra, S., "On the implementation of a primal-dual interior point method," *SIAM Journal on optimization*, Vol. 2, No. 4, 1992, pp. 575–601.

[22] Nocedal, J., and Wright, S. J., *Sequential quadratic programming*, Springer, 2006.

[23] Lai, M.-J., and Schumaker, L. L., *Spline functions on triangulations*, 110, Cambridge University Press, 2007.

[24] Billings, S., and Voon, W., "A prediction-error and stepwise-regression estimation algorithm for non-linear systems," *International Journal of Control*, Vol. 44, No. 3, 1986, pp. 803–822.

[25] Billings, S., Korenberg, M., and Chen, S., "Identification of non-linear output-affine systems using an orthogonal least-squares algorithm," *International Journal of Systems Science*, Vol. 19, No. 8, 1988, pp. 1559–1568.

[26] Klein, V., and Morelli, E. A., *Aircraft System Identification*, AIAA, 2006.

[27] Lombaerts, T., Chu, Q., Mulder, J., and Joosten, D., "Real time damaged aircraft model identification for reconfiguring flight control," *AIAA atmospheric flight mechanics conference and exhibit*, 2007, p. 6717.

[28] Nelson, R. C., *Flight stability and automatic control*, Vol. 2, WCB/McGraw Hill New York, 1998.

[29] de Visser, C. C., Chu, Q., and Mulder, J., "Differential constraints for bounded recursive identification with multivariate splines," *Automatica*, Vol. 47, No. 9, 2011, pp. 2059–2066.

[30] Mulder, J. A., "Design and evaluation of dynamic flight test manoeuvres," Ph.D. thesis, Technische Universiteit Delft, 1986.

[31] de Visser, C., Mulder, J. A., and Chu, Q. P., "A Multidimensional Spline Based Global Nonlinear Aerodynamic Model for the Cessna Citation II," *AIAA Atmospheric Flight Mechanics Conference*, 2010.

[32] de Visser, C., Mulder, J., and Chu, Q., "Global Nonlinear Aerodynamic Model Identification with Multivariate Splines," *AIAA Atmospheric Flight Mechanics Conference*, 2009.

[33] Tarantola, A., *Inverse problem theory and methods for model parameter estimation*, SIAM, 2005.

Part II

Preliminary Thesis

Chapter 1

Introduction

From 2019 a new aviation legislation makes all air-carrier pilots obliged to go through flight simulator-based stall recovery training (Federal Aviation Administration, n.d.). A 'task force' was formed by the division of the Control & Simulation (C&S) group as result of this new aviation legislation. The implication of this legislation is that current flight simulation models must be updated regarding accurate pre-stall, stall, and post-stall dynamics. Recently a new methodology is derived for high-fidelity aircraft stall behaviour modelling and simulation and a new high-fidelity flight simulation model of the Cessna Citation II laboratory aircraft has been developed. This model is made up of sets of local linear models which must be interpolated in order to adequately cover the flight envelope. The current Delft University Aircraft Simulation Model and Analysis Tool (DASMAT) model will eventually be replaced by this model.

Performing this interpolation is not trivial. For example, naive database interpolation can result in significant discontinuities and other artefacts which may influence simulator fidelity. A promising alternative approach uses MSBS to accurately model nonlinear aerodynamic models over the entire flight envelope and are easily integrated into standard identification routines. While the multivariate B-splines are powerful function approximators, their downside has been that their basis functions and B-coefficients do not have a straightforward physical interpretation due to the use of local coordinate systems.

Very recently, a new formulation of the MSBS was developed. These physical B-splines or Physical-Splines (P-Splines) are a modification of standard multivariate B-splines in which the B-coefficients have a physical interpretation, i.e. they are transformed from barycentric coordinate space into Cartesian space under a linear coordinate transformation. This has removed the primary disadvantage of the multivariate spline method, and will allow its much more widespread use as a tool in aircraft system identification.

Chapter 2

Physical-Splines and back again

In the early days, before the time of computers, splines were used as drafting tools to draw curves. This gave drafts men and woman special capabilities. Especially in the in the engineering field of shipbuilding splines were used in the form of flexible, long and thin strips of wood. Often balsa wood. While drawing ship plans the crafts tperson gave form to the spline on the paper and weights were attached to keep the spline in place. The weights are called ducks, because of their shape. With the ducks in place, the splines are secured. Now a line with great curvature can be drawn accurately.

Besides ships, curved lines are very important for another famous streamlined object, the aircraft. The aforementioned thin wooden strips are used again to create these lines. In the field of aviation this drafting technique is known as lofting. This technique is later replaced by mathematical tables, used, among other aircraft, for designing the P-51 Mustang, and eventually replaced by computers.

Curved objects have always played an important and fascinating role for mankind, whether this is for wheels, streamlined design of ancient Roman ships or modern aircraft. It can be no surprise mathematicians got interested by splines. One of the first reference of splines in mathematics was made by the Romanian born American Iscaac Jacob Schoenberg. After the Second World War he published a paper (Schoenberg, 1946). In this paper he describes the spline as a piecewise smooth polynomial approximation. Figure 2-2 shows a spline function with five piecewise linear polynomials.

With computer technology getting better, faster and more advanced the potentials in Computer-Aided Design (CAD) became clear. CAD software gives the possibility to design interactively and create for example curves displayed graphically on a monitor. In the automotive industry a lot of research was performed on splines. Among them was Citroën, one the first companies to see the importance of splines. Researchers as de Casteljau and Boor played an important role in developing splines methods in search of a true generalised multivariate form of splines functions. Methods developed and used for CAD proved to be very promising in other fields and nowadays they are used in for example data fitting and system identification tasks. In the remainder of this chapter an introduction to different splines

Figure 2-1: Two man lofting, (*Cecil Beaton, Ministry of Information, 1943*)

methods is given in order to get a better understanding what let to the MSBS. During this introduction some attention is given to the very important and beautiful mathematical basis they share. The introduction will started with the physical wooden splines and ends with the formulation of the Physical MSBS or P-Splines.

2-1 Bernstein Polynomials

This section introduces a polynomial basis. Because of the mathematic expression the splines are becoming from this point onwards less physical. The polynomials used are called the Bernstein Polynomials. These polynomials are named after the Russian Sergei Natanovich Bernstein. The univariate Bernstein polynomials are defined by (2-1)

$$b_{\nu,d}(x) = \begin{pmatrix} d \\ \nu \end{pmatrix} = x^{\nu} (1-x)^{d-\nu}, \quad \nu = 0, \ldots, d \tag{2-1}$$

where the binomial coefficient is given by

$$\begin{pmatrix} d \\ \nu \end{pmatrix} = \frac{d}{\nu!(d-\nu)!} \tag{2-2}$$

This equation produces $d+1$ Bernstein polynomials of degree d. The binomial coefficient can look familiar since it produces the famous Pascal's triangle. The univariate formulation of the Bernstein Polynomial's in (2-1) will play an important role in the MSBS. Later on a general formulation is given, which can be seen as a general multivariate formulation of the Bernstein polynomials and therefore for the Pascal's triangle.

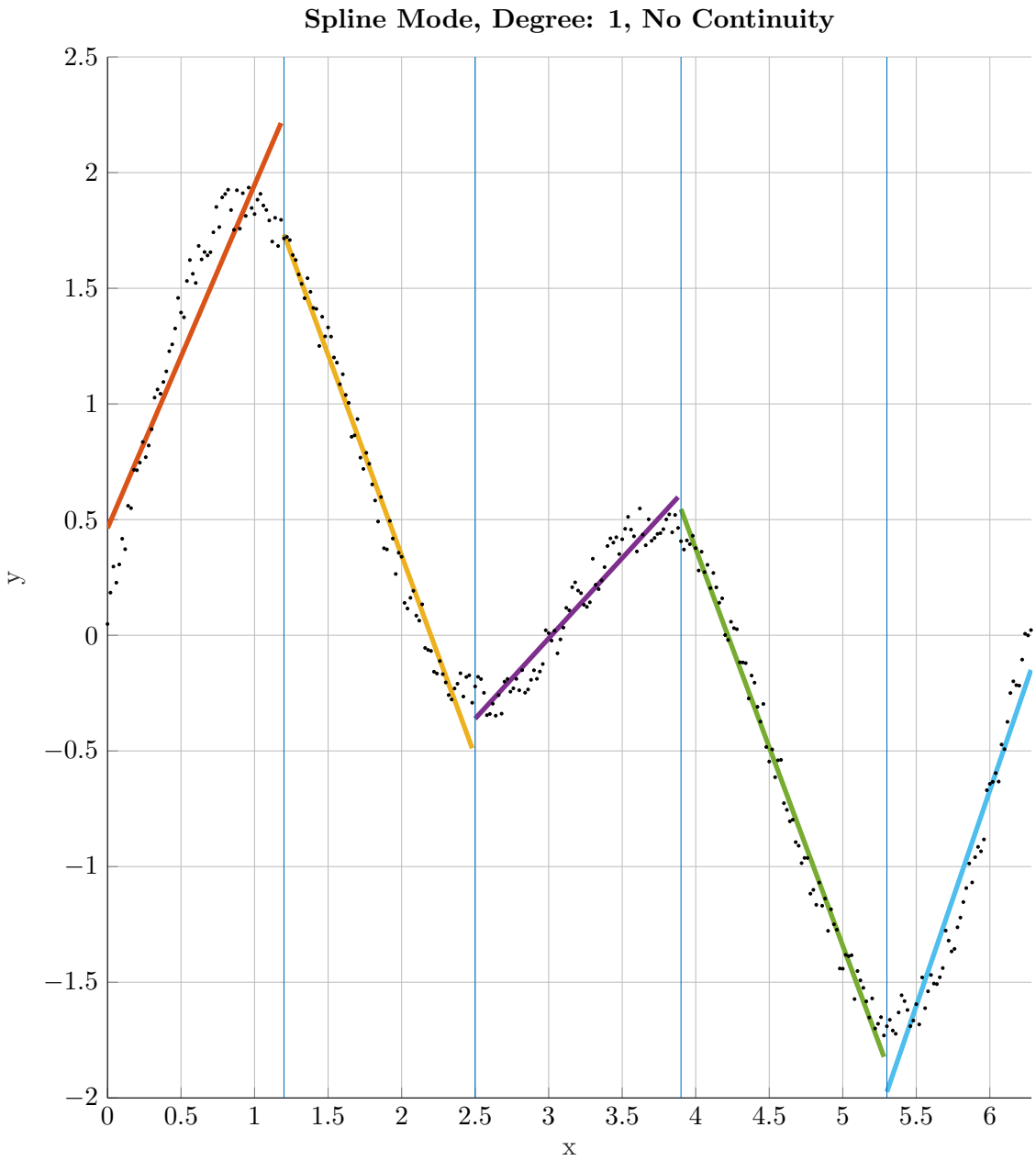


Figure 2-2: One dimensional linear spline function with no continuity, approximating non-linear data.

2-2 Bézier Curves

One of the first curve drawing methods where the Bernstein Polynomials are used are the Bézier Curves. In this section an example how the Bézier Curves works is given. Figure 2-3 shows a cubic Bézier curve S with four control points P_0, P_1, P_2, P_3. The curve S is defined as a function of t where t by definition equals $0 \leq t \leq 1$. In other words a point on the curve is given by function $S(t)$. In vector notation a point on the curve is expressed as $\bar{s}_0$. For a better understanding how a Bézier curve works the vector $\bar{s}_0$ is rewritten in a weighted combination of $\bar{r}_0$ and $\bar{r}_1$.

$$\bar{s}_0 = (1-t)\bar{r}_0 + t\bar{r}_1 \tag{2-3}$$

In a similar fashion $\bar{r}_0$ and $\bar{r}_1$ are rewritten in a weighted combination of $\bar{q}_0$, $\bar{q}_1$ and $\bar{q}_2$.

$$\begin{aligned} \bar{r}_0 &= (1-t)\bar{q}_0 + t\bar{q}_1 \\ \bar{r}_1 &= (1-t)\bar{q}_1 + t\bar{q}_2 \end{aligned} \tag{2-4}$$

and once more $\bar{q}_0$, $\bar{q}_1$ and $\bar{q}_2$ are rewritten in a weighted combination of the control points $\bar{p}_0$, $\bar{p}_1$, $\bar{p}_2$, $\bar{p}_3$.

$$\begin{aligned} \bar{q}_0 &= (1-t)\bar{p}_0 + t\bar{p}_1 \\ \bar{q}_1 &= (1-t)\bar{p}_1 + t\bar{p}_2 \\ \bar{q}_0 &= (1-t)\bar{p}_2 + t\bar{p}_3 \end{aligned} \tag{2-5}$$

To find a relation between the curve and the control points equations (2-4) and (2-5) are substituted in (2-4) and after simplifying (2-6) is obtained

$$\bar{s}_0 = (1-t)^3\,\bar{p}_0 + 3(1-t)^2\,t\,\bar{p}_1 + 3(1-t)\,t^2\,\bar{p}_2 + t^3\,\bar{p}_3 \tag{2-6}$$

Note that in equation (2-6) the curve $\bar{s}_0$ is a relation between Bernstein polynomials of degree three and the control points. The degree three Bernstein polynomials are

$$((1-t)+t)^3 = (1-t)^3 + 3(1-t)^2\,t + 3(1-t)\,t^2 + t^3 = 1 \tag{2-7}$$

The Bernstein polynomials have the nice property that they sum to 1 and are positive for $0 \leq t \leq 1$. In Figure 2-3 the control points P_0, P_1, P_2, P_3 are vertices of a quadrilateral. The curve S is in this convex combination of P_0, P_1, P_2, P_3 and a point on this curve is the Centre of Mass (c.m.) of the quadrilateral where the weights are determined by the Bernstein polynomials. In other words when $t = 0$ all the mass is at control point P_0 when t increases the mass of the other control points increases and the mass of control point P_0 decreases. When $t = 1$ all the mass is at control point P_3.

To evaluate the Bézier curves Paul de Casteljau developed a recursive algorithm. At that time, 1959, he was working at the car manufacturer Citroën. These astounding results were kept

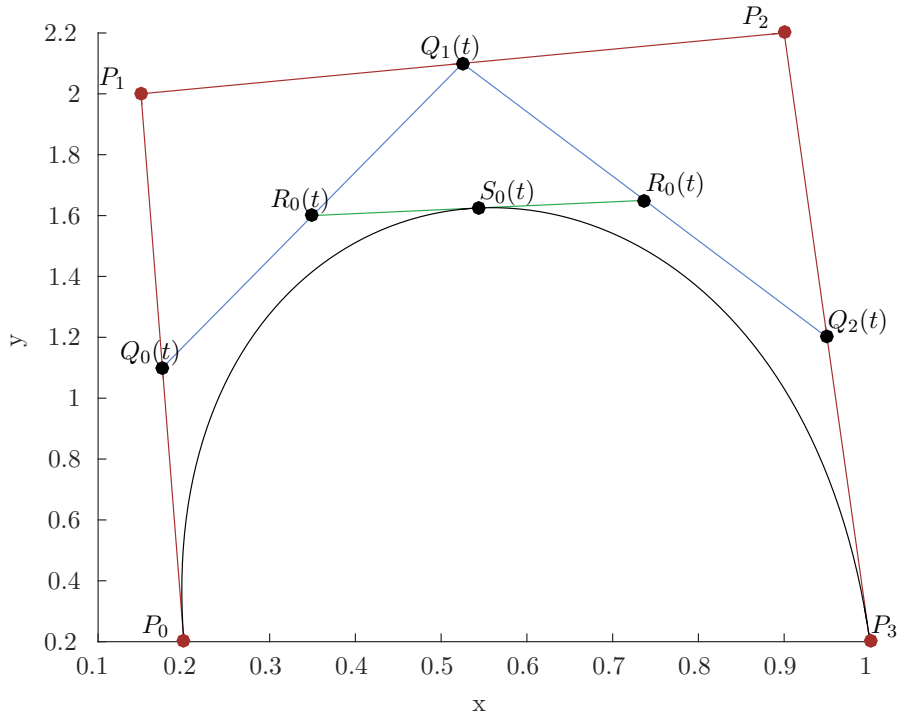


Figure 2-3: A quadratic Bézier Curve

secret by Citroën because of the company's policy. Later the the algorithm was formalized and made popular by Pierre Bézier himself. It was until 1974 when Paul de Casteljau published a paper about his algorithm and was acknowledge for his work. De Casteljau's algorithm is given by

$$S(t) = \sum_{i=0}^{d} p_i b_{i,d}(t) \tag{2-8}$$

where $S(t)$ is the Bézier curve, p the control point, $d+1$ the number of control points, d the degree and $b(t)$ the Bernstein polynomials.

2-3 B-Splines

Bézier curves have certain shortcomings when a more complex-shape curve has to be approximated. They do not have local control and there is a strict dependency between the degree of the curve and the number of control points. So what if a complex-shaped curve has to be approximated. An option is to use a higher degree Bézier curve, but the disadvantage is that higher degree Bézier curves are computational expensive to compute and hard to control. When a small change is made to whole Bézier curve has to be recalculated. To overcome this disadvantage it would be great to join the ends of several Bézier curves together into one curve, i.e. piecewise polynomial functions, and make this ends tangent with a certain order of continuity. This is what B-Splines are. Rather than having a single higher degree Bézier

curve, multiple lower degree Bézier curves are joined onto each other. The first appearance of B-Splines, or Basis-Splines was in the paper of Schoenberg (Schoenberg, 1946). However Schoenberg said that they were known to LaPlace. It was until 1966 that this paper was published that the B-Splines were given their name. In this paper (Curry & Schoenberg, 1966) the importance of B-Splines is acknowledge and was intended to show that B-Splines are truly basic splines. Later this is promoted by de Boor as well in (C. D. Boor, 1976). Similar to the Bézier curves the B-Spline is given as a linear combination of a control point p and a basis function $N(t)$.

$$S(t) = \sum_{i=0}^{n} p_i N_{i,d}(t) \tag{2-9}$$

The major difference compared to Bézier curves is that the domain of a B-Spline is divided by knots t and that the entire interval of basis functions are not non-zero. The knots are determined by the knot-vector. A vector with $m+1$ non-decreasing numbers. A knot-vector with equally spaced knots is called uniform and trivially a knot vector with non-equally spaced knots is called non-uniform. The knots is what give B-splines their local control. The knots span certain segments, the so called knot-span. Each segment can be seen as a Bézier curve with its own basis functions $N_{i,d}(t)$. The number of knots is equal to $m+1$, the number of control points to $n+1$ and the degree is d. By definition there is a relation between the knots, degree and control points: $m = n + d + 1$. What this relation tells is that for a second degree spline (i.e. $d = 2$) and five control points there will be eight knots. The number of knot-spans or segments is given by subtracting the degree from the number of control points or $n + 1 - d$. For the latter situation this results in $5 - 2 = 3$ segments.

Knots may re-appear k number of times. This is called multiplicity. The multiplicity of the knots however decreases the continuity. The continuity of the B-spline is given by C^{d-k}.

In 1972 Cox and de Boor derived at the same time recursive algorithm that calculates the basis functions, which is explained in (Cox, 1975) and (C. de Boor, 1972) respectively. The Cox - de Boor algorithm is given by (2-11).

$$S(t) = \sum_{i=0}^{n} B_{i,d}(t) c_i \tag{2-10}$$

$$\begin{aligned} N_{i,0}(t) &= \begin{cases} 1 & \text{if } t_i \leq t \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,d}(t) &= \frac{t - t_i}{t_{i+d} - t_i} N_{i,\,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} N_{i+d,\,d-1}(t) \end{aligned} \tag{2-11}$$

The t's in (2-11) with subscript are not the parameters t but the knots. The one in (2-11) indicates in which segment t is in. Different forms of the B-Splines exist and using knot multiplicity a closed or open B-Splines can be created. The nice properties of the B-Splines initiated the start in search for a multivariate general formulation.

2-4 Multivariate Splines

In continuation to find a true generalisation for the multivariate splines many research is performed in this field. Finding a multivariate generalisation turned out to be rather difficult. Multiple formulation have been generated, among them are the tensor splines, polyhedral splines, thin-plate splines and simplex splines. The different formulations all have some advantages and shortcomings regarding the use of local or global basis functions, computational efficiency, and very important for aerodynamic modelling if they can be used to fit a scattered dataset. The simplex splines method is very promising. It uses triangular patches called the Bézier patch. Only since recent years the simplex splines increased in popularity and more research is performed. It was until 2005 before they were used in aerodynamic modelling and system identification. In the next section the MSBS method is elaborated on how they can be used for system identification on a scattered dataset and how to give them a physical interpretation.

2-5 (Physical) Multivariate Simplex B-Splines

The MSBS is a general function approximator. It can fit any scattered non-linear dataset, is compatible with linear regression methods and has local basis functions. Furthermore it requires a geometric support structure called a triangulation and a disadvantage is that the B-coefficients, to be estimated, don't have a straightforward physical meaning. In this section the theory of the Multivariate Simplex B-splines is explained in three parts. The first part is about the simplex, used to create the geometric support structure, the second part is about the barycentric coordinate system and the last part about the Bernstein basis polynomials in barycentric space.

2-5-1 Simplex

A simplex is a generalisation of a point, line, triangle, tetrahedron, etc. An n dimensional n-simplex has $n+1$ vertices. The vertices are connected to each other with edges. Each n-simplex forms a convex hull of $n+1$ independent vertices. A 0-simplex has one vertex and is therefore a point. A 1-simplex is a line-segment, a 2-simplex forms a triangle and a 3-simplex a tetrahedron. A n-simplex is affinely independent $\in \mathbb{R}^n$. An affine space is a generalization of the properties of the Euclidean space. In the affine space no origin exists. For example a 2-simplex, a triangle with 3 vertices, is two dimensional affine space or an affine plane. Because of this property simplices can be used to approximate any n-dimensional shape. Therefore they are often used for creating 3D ($n = 3$) graphical computer animations and modelling. The number of n-faces can be determined with Pascal's triangle and are given for 1-simplex up to a 6-simplex in table 2-1. In this table an important property of the simplex faces can be obtained, namely they are simplices as well only with dimensions lower than n. This makes the simplex recursively. An example of a 2-simplex is given in Figure 2-5. Figure 2-4 displays a 3-simplex, also known as a tetrahedron. Multiple simplices can be combined. When aligning the facets of these adjacent simplices a triangulation is created.

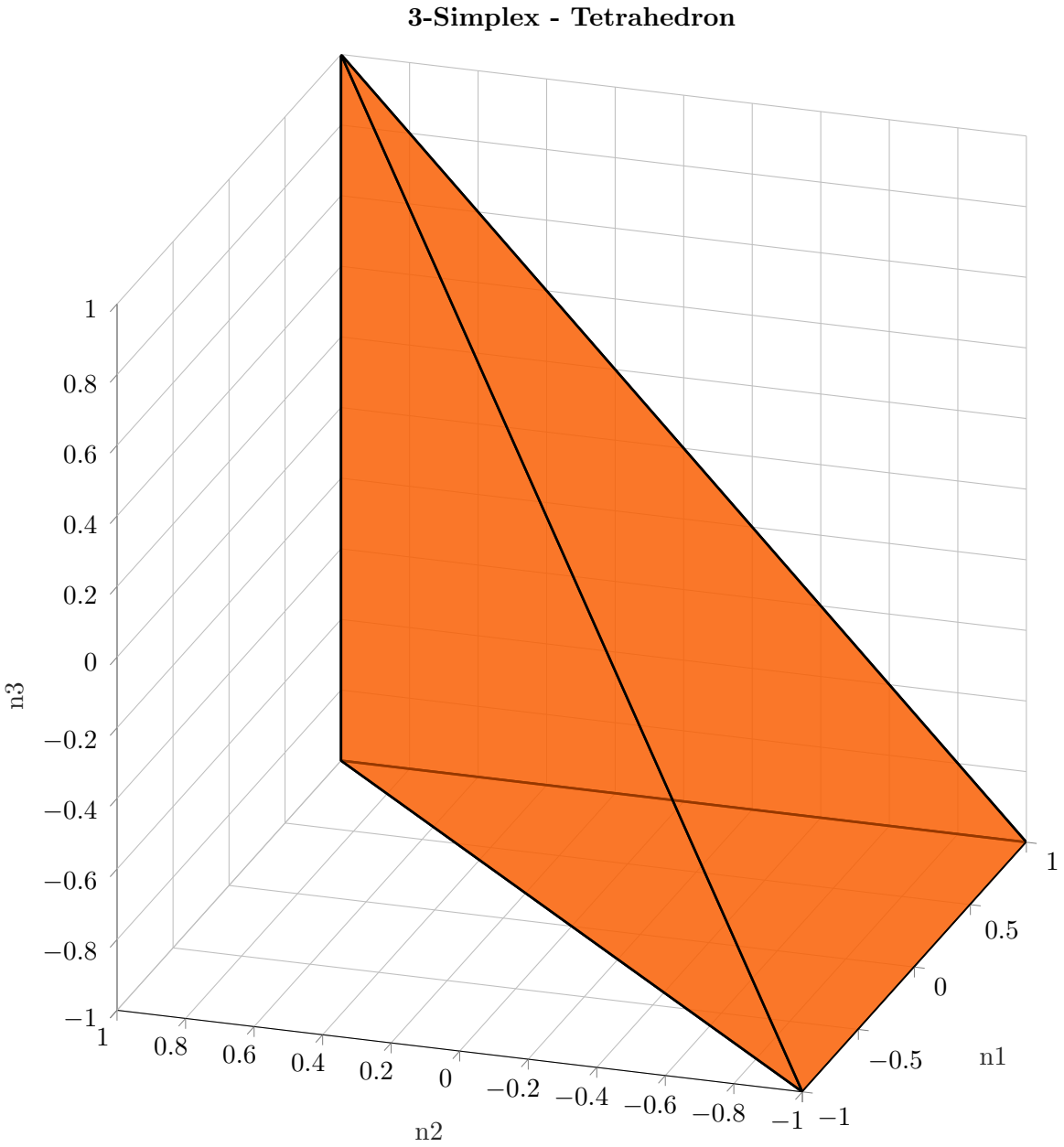


Figure 2-4: An example of a 3-simplex, or tetrahedron, with four vertices, six edges and four 2-faces.

Table 2-1: Faces of n-simplex

n	0-faces (vertices)	1-faces (edges)	2-faces	3-faces	4-faces	5-faces	6-faces
0 point	1						
1 line	2	1					
2 triangle	3	3	1				
3 tetrahedron	4	6	4	1			
4-simplex	5	10	10	5	1		
5-simplex	6	15	20	15	6	1	
6-simplex	7	21	35	35	35	21	1

2-5-2 Barycentric Coordinate System

Each simplex has its own local coordinated system called the Barycentric Coordinates. The Barycentric coordinates are introduced in 1827 by Möbius. A barycentre is an other word for Centre of Mass (c.m.). A coordinate within a simplex is determined relative to the vertices of the simplex. One can say that a coordinates or location of the barycentre is represented by the "mass" each vertex has. The barycentric coordinates used for the MSBS are normalised so

$$1 = \sum_{i=0}^{n} b_i \tag{2-12}$$

Where b_i is a vertex in barycentric space. So when a single barycentric coordinate equals 1 all the "mass" is located at this vertex. Hence the point within the simplex is located on this vertex. Before the barycentric coordinate system can be used, the coordinates of points within a simplex given in the Cartesian system need to be transformed first. This linear transformation is given by equation (2-13)

$$\begin{aligned} b_0 &= 1 - \sum_{i=1}^{n} b_i \\ \begin{bmatrix} b_1 & b_2 & \dots & b_n \end{bmatrix}^T &= \mathbf{A}_{t_j}^{-1} \cdot \left(\mathbf{x}^T - \boldsymbol{\nu}_0^T \right) \\ \mathbf{A}_{t_j} &= \begin{bmatrix} (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_0)^T & (\boldsymbol{\nu}_2 - \boldsymbol{\nu}_0)^T & \dots & (\boldsymbol{\nu}_n - \boldsymbol{\nu}_0)^T \end{bmatrix} \end{aligned} \tag{2-13}$$

In the previous equations the locations of the vertices in Cartesian coordinates for a n-simplex t_j are given by $(\nu_0, \nu_1, \dots, \nu_n)$. A point $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is transformed using the non-singular matrix $\mathbf{A}_{t_j}$ to normalised barycentric coordinates $(b_0, b_1, \dots, b_n)$. When this point x is outside the simplex t_j the Barycentric coordinates are negative. In equation (2-14) a short-hand notation is introduced for the Cartesian to Barycentric coordinate transformation. In Figure 2-5 a 2-simplex indicates the barycentric coordinates of the vertices and a point located exactly in the middle.

$$b = b_{t_j}(x) \;\forall\; x \in t_j \tag{2-14}$$

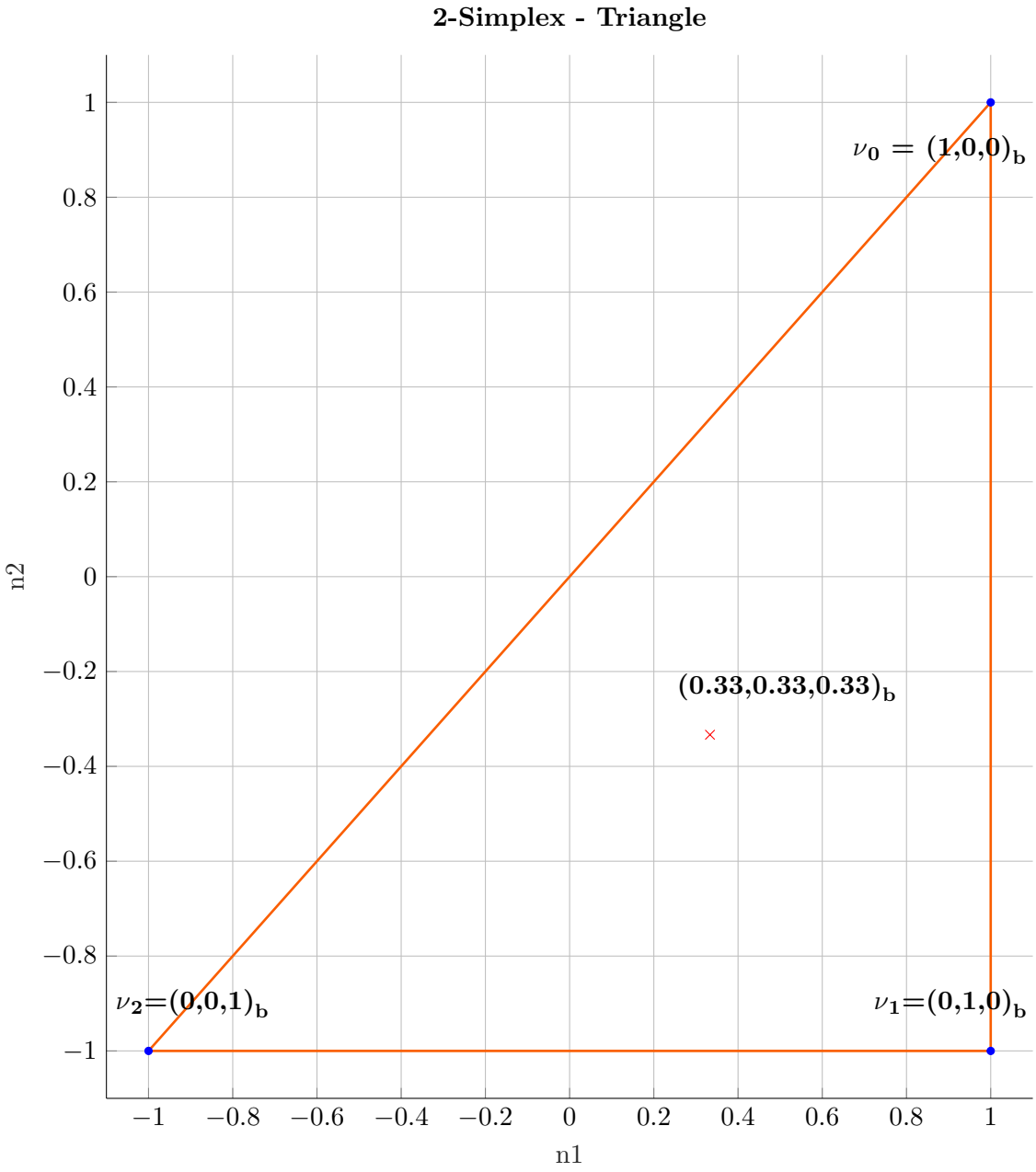


Figure 2-5: Two dimensional 2-simplex, indicating the normalised $n = 3$ barycentric coordinates of the vertices and a point. The axis are in Cartesian coordinates.

2-5-3 Multivariate Bernstein Basis Polynomials

The MSBS polynomials are formed using the summation of $n+1$ barycentric coordinates as variables raised to a certain degree d.

$$p\left(b_0, b_1, \cdots, b_n\right) = \left(b_0 + b_1 + \cdots + b_n\right)^d \tag{2-15}$$

Using the multinomial theorem in (2-16) the polynomial in (2-15) can be expanded in individual terms, i.e. a linear summation of basis functions.

$$\begin{aligned}\left(b_0 + b_1 + \cdots + b_n\right)^d &= \sum_{\kappa_0+\kappa_1+\cdots+\kappa_n=d} \frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \ldots b_n^{\kappa_n} \\ &= \sum_{\kappa_0+\kappa_1+\cdots+\kappa_n=d} \mathbf{B}_{\kappa_0,\kappa_1,\cdots,\kappa_n}(b_0, b_1, \ldots, b_n) \\ \text{where} \quad d \geq 0 \quad &\text{and} \quad (\kappa_0 + \kappa_1 + \cdots + \kappa_n) \in \mathbb{N}\end{aligned} \tag{2-16}$$

In the previous equation $\kappa = (\kappa_0, \kappa_1, \ldots, \kappa_n)$ and is called the multi-index. The multi-index is given the following properties

$$\begin{aligned}|\kappa| &= \kappa_0 + \kappa_1 + \cdots + \kappa_n \\ \kappa! &= \kappa_0!\kappa_1! \ldots \kappa_n!\end{aligned} \tag{2-17}$$

Using these properties in (2-17) and the short-hand notation in (2-14), equation (2-16) is simplified

$$\begin{aligned}\left(b_0 + b_1 \cdots + b_n\right)^d &= \sum_{\kappa_0+\kappa_1+\cdots+\kappa_n=d} \frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} \\ &= \sum_{|\kappa|=d} \frac{d!}{\kappa!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} \\ &= \sum_{|\kappa|=d} \frac{d!}{\kappa!} b^{\kappa} \\ &= \sum_{|\kappa|=d} \mathbf{B}_{\kappa}^{d}(b) \\ &= 1\end{aligned} \tag{2-18}$$

In (2-18) the summation to 1 is the result of the normalised barycentric coordinates, making the basis functions stable. What this implies is that in barycentric space all the simplices are equal no matter if they differ in size in Cartesian space. The basis functions are active only locally and 0 elsewhere, this makes the basis local. Together they form a Stable Local Basis.

2-5-4 de Boor's Theorem

The de Boor's theorem states that it is possible to express any polynomial $p(x)$, given in Cartesian coordinates, by using the B-form (2-18) in combination with B-coefficients $\mathbf{c}$, see (2-19).

$$p\left(x\right) = \sum_{|\kappa|=d} c_{\kappa}^{t_j} \mathbf{B}_{\kappa}^{d}\left(b_{t_j}\left(x\right)\right) \quad \forall\ x \in t_j \tag{2-19}$$

and by definition a simplex spline is only locally active

$$p\left(x\right) = 0 \quad \forall\ x_i \notin t_j \tag{2-20}$$

An interesting and very important feature of the B-coefficients is their unique spatial location within a simplex. The barycentric coordinate system makes it possible to ask the question: where is my coefficient located? This property will be very important to generate continuity within the spline model but also for validation. The basics for the de Boor's theorem are given in the publication (C. de Boor, 1987). For a more comprehensive derivation, (De Visser, 2011) is recommended.

2-5-5 Transformation to Physical Space

When using MSBS for Aerodynamic Model Identification the estimated parameters cannot be physically interpreted. The B-coefficients don't have a straightforward physical interpretation. The model structure of every simplex is in de form of a polynomial in the Barycentric Coordinates system. The physical parameters, or stability derivatives, form a very important part of the flight dynamics. They are studied extensively and are used to analyse the stability and handling characteristics of an aircraft. Therefore a transformation from the Barycentric parameters to physical parameters will be very advantageous.

Let's start with this transformation. For example a two dimensional, second degree model is created of the pitching moment coefficient $C_m(\alpha, \delta_e)$ using the MSBS method. The polynomial using the de Boor's formulation of the B-form, (2-19), has the form of equation (2-21) and includes physical variables transformed to barycentric coordinates, i.e. physical input values such as α and δ_e are transformed first to b_0, b_1, b_2. Latter formulation is given for a single simplex t_j and is written in matrix form first and for clarity expanded to an equivalent single equation using the multinomial expansion of (2-16).

$$\begin{aligned}
C_m(\alpha,\delta_e) &= \sum_{|\kappa=2|} \mathbf{B}_{\kappa}^{2}(b_{t_j}(\alpha,\delta_e)) \cdot c_{\kappa}^{t_j} \quad \forall(\alpha,\delta_e) \in t_j \\
&= \begin{bmatrix} \mathbf{B}_{2,0,0}^{2}(b_{t_j}(\alpha,\delta_e)) & \mathbf{B}_{2-1,1,0}^{2}(b_{t_j}(\alpha,\delta_e)) & \cdots & \mathbf{B}_{0,0,2}^{2}(b_{t_j}(\alpha,\delta_e)) \end{bmatrix} \cdot \begin{bmatrix} c_{2,0,0}^{t_j} \\ c_{2-1,1,0}^{t_j} \\ \vdots \\ c_{0,0,2}^{t_j} \end{bmatrix} \\
&= c_{200}^{t_j} b_0^2 + c_{110}^{t_j} 2 b_0 b_1 + c_{101}^{t_j} 2 b_0 b_2 + c_{020}^{t_j} b_1^2 + c_{011}^{t_j} 2 b_1 b_2 + c_{002}^{t_j} b_2^2
\end{aligned} \tag{2-21}$$

The idea is to use the Physical (Cartesian) to Barycentric coordinates transformation and find the relation between the Barycentric variables and the Physical variables. The transformation is given by (2-22) where A_{t_j} and k_{t_j} depend on the geometry of the Simplex.

$$b(\alpha, \delta_e) = \mathbf{A}_{t_j} \cdot \begin{bmatrix} \alpha \\ \delta_e \end{bmatrix} + \mathbf{k}_{t_j} \tag{2-22}$$

Rewritten in matrix form the relation between the Physical variables and the Barycentric variables for simplex t_j is given by

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \delta_e \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \tag{2-23}$$

which equals

$$\begin{aligned} b_0 &= a_{11}\alpha + a_{12}\delta_e + k_1 \\ b_1 &= a_{21}\alpha + a_{22}\delta_e + k_2 \\ b_2 &= a_{31}\alpha + a_{32}\delta_e + k_3 \end{aligned} \tag{2-24}$$

Substitute these terms into equation (2-21) yields 2-25

$$\begin{aligned} C_m(\alpha, \delta_e) = {} & c_{200}^{t_j}(a_{11}\alpha + a_{12}\delta_e + k_1)^2 + c_{110}^{t_j} 2(a_{11}\alpha + a_{12}\delta_e + k_1)(a_{21}\alpha + a_{22}\delta_e + k_2) \\ & + c_{101}^{t_j} 2(a_{11}\alpha + a_{12}\delta_e + k_1)(a_{31}\alpha + a_{32}\delta_e + k_3) + c_{020}^{t_j}(a_{21}\alpha + a_{22}\delta_e + k_2)^2 \\ & + c_{011}^{t_j} 2(a_{21}\alpha + a_{22}\delta_e + k_2)(a_{31}\alpha + a_{32}\delta_e + k_3) + c_{002}^{t_j}(a_{31}\alpha + a_{32}\delta_e + k_3)^2 \end{aligned} \tag{2-25}$$

by grouping each variable the physical coefficients reveal themselves in 2-26

$$
\begin{aligned}
C_m(\alpha,\delta_e) &= \underbrace{c_{200}k_1^2 + 2c_{110}k_1k_2 + 2c_{101}k_1k_3 + c_{020}k_2^2 + 2c_{011}k_2k_3 + c_{002}k_3^2}_{C_{m_0}} \\
&+ \underbrace{2\left(c_{200}a_{11}k_1 + c_{110}(a_{11}k_2a_+21k_1) + c_{101}(a_{11}k_3 + a_{31}k_1) + c_{020}a_{21}k_2 + \right.}_{\underbrace{\left. c_{011}(a_{21}k_3 + a_{31}k_2) + c_{002}a_{31}k_3\right)}_{C_{m_\alpha}}\alpha} \\
&+ \underbrace{2\left(c_{200}a_{12}k_1 + c_{110}(a_{12}k_2 + a_{22}k_1) + c_{101}(a_{32}k_1 + a_{12}k_3) + c_{020}a_{22}k_2 + \right.}_{\underbrace{\left. c_{011}(a_{32}k_2 + a_{22}k_3) + c_{002}a_{32}k_3\right)}_{C_{m_\delta}}\delta_e} \\
&+ \underbrace{\left(c_{200}a_{11}^2 + 2c_{110}a_{11}a_{21} + 2c_{101}a_{11}a_{31} + c_{020}a_{21}^2 + 2c_{011}a_{21}a_{31} + c_{002}a_{31}^2\right)}_{C_{m_{\alpha 2}}}\alpha^2 \\
&+ \underbrace{2\left(c_{200}a_{11}a_{12} + c_{110}(a_{11}a_{22} + a_{12}a_{21}) + c_{101}(a_{11}a_{32} + a_{12}a_{31}) + \right.}_{\underbrace{\left. c_{020}a_{21}a_{22} + c_{011}(a_{21}a_{32} + a_{22}a_{31}) + c_{002}a_{31}a_{32}\right)}_{C_{m_{\alpha\delta}}}\alpha\delta_e} \\
&+ \underbrace{\left(c_{200}a_{12}^2 + 2c_{110}a_{12}a_{22} + 2c_{101}a_{12}a_{32} + c_{020}a_{22}^2 + 2c_{011}a_{22}a_{32} + c_{002}a_{32}^2\right)}_{C_{m_{\delta 2}}}\delta_e^2
\end{aligned}
\tag{2-26}
$$

Rewriting the previous equation in matrix notation gives the transformation matrix $\mathbf{\Lambda}_{t_j}$

$$
\begin{bmatrix} C_{m_0} \\ C_{m_\alpha} \\ C_{m_\delta} \\ C_{m_{\alpha 2}} \\ C_{m_{\alpha\delta}} \\ C_{m_{\alpha\delta}} \\ C_{m_{\delta 2}} \end{bmatrix}_{t_j} = \mathbf{\Lambda}_{t_j} \cdot \begin{bmatrix} c_{200} \\ c_{110} \\ c_{101} \\ c_{020} \\ c_{011} \\ c_{002} \end{bmatrix}_{t_j} \tag{2-27}
$$

where the transformation matrix equals

$$
\mathbf{\Lambda}^{t_j} = \begin{bmatrix}
k_1^2 & 2k_1k_2 & 2k_1k_3 & k_2^2 & 2k_2k_3 & k_3^2 \\
2a_{11}k_1 & 2(a_{11}k_2 + a_{21}k_1) & 2(a_{11}k_3 + a_{31}k_1) & 2a_{21}k_2 & 2(a_{21}k_3 + a_{31}k_2) & 2a_{31}k_3 \\
2a_{12}k_1 & 2(a_{12}k_2 + a_{22}k_1) & 2(a_{32}k_1 + a_{12}k_3) & 2a_{22}k_2 & 2(a_{32}k_2 + a_{22}k_3) & c_{002}a_{32}k_3 \\
a_{11}^2 & 2a_{11}a_{21} & a_{11}a_{31} & a_{21}^2 & 2a_{21}a_{31} & a_{31}^2 \\
2a_{11}a_{12} & 2(a_{11}a_{22} + a_{12}a_{21}) & 2(a_{11}a_{32} + a_{12}a_{31}) & 2a_{21}a_{22} & 2(a_{21}a_{32} + a_{22}a_{31}) & 2a_{31}a_{32} \\
a_{12}^2 & 2a_{12}a_{22} & 2a_{12}a_{32} & a_{22}^2 & 2a_{22}a_{32} & a_{32}^2
\end{bmatrix} \tag{2-28}
$$

The physical transformation $\mathbf{\Lambda}$ is determined for all simplices and can be combined in a single large sparse transformation matrix, see (2-29)

$$\begin{aligned}\mathbf{c}_p &= \mathbf{\Lambda} \cdot \mathbf{c}_b \\ &= \begin{bmatrix} \mathbf{\Lambda}_{t_1} & 0 & 0 & 0 \\ 0 & \mathbf{\Lambda}_{t_2} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{\Lambda}_{t_j} \end{bmatrix} \cdot \mathbf{c}_b \end{aligned} \tag{2-29}$$

With the linear transformation $\mathbf{\Lambda}$ the MSBS model can now be physically interpreted. This transformation is reversible, so inverting $\mathbf{\Lambda}$ makes it possible to transform from physical space to barycentric space. The order of appearance of the physical terms is given by the generalized equation (2-30)

$$\begin{aligned} p(x_1, x_2, \ldots, x_n)_{t_j} &= \sum_{|\kappa|=d} c_{p_\kappa} \cdot 1^{\kappa_0} \cdot x_1^{\kappa_1} x_2^{\kappa_2} \ldots x_n^{\kappa_n} \\ \text{where } \kappa &= (\kappa_0, \kappa_1, \kappa_2, \ldots, \kappa_n) \in \mathbb{N} \text{ is the multi-index} \\ \text{with property } |\kappa| &= (\kappa_0 + \kappa_1 + \kappa_2 + \cdots + \kappa_n) \end{aligned} \tag{2-30}$$

In this equation are the physical variables given by $(x_1, x_2, \ldots, x_n)$ with degree d and physical parameters c_p. In equation (2-31) an example for a two dimensional $n = 2$ and third degree $d = 3$ physical polynomial is given. Such a polynomial exist for all simplices t_j and $\forall\ x\ \in t_j$

$$\begin{aligned} p(x_1, x_2) = & c_{p_{00}} \\ & c_{p_{10}} \cdot x_1 + c_{p_{01}} \cdot x_2 \\ & c_{p_{20}} \cdot x_1^2 + c_{p_{11}} \cdot x_1 \cdot x_2 + c_{p_{02}} \cdot x_2^2 \\ & c_{p_{30}} \cdot x_1^3 + c_{p_{21}} \cdot x_1^2 \cdot x_2 + c_{p_{12}} \cdot x_1 \cdot x_2^2 + c_{p_{03}} \cdot x_2^3 \end{aligned} \tag{2-31}$$

Chapter 3

Aerodynamic Modelling

Aerodynamic modelling is a system identification process in creating an aerodynamic model for the flight dynamics of an aircraft. In essence system identification relates to the procedure of giving an input u while measuring the output y and subsequently determining the model or plant s. Harry Greenberg was one of the first to publish about aerodynamic modelling in (Greenberg, n.d.). He used various methods to determine the longitudinal stability parameters of an idealised simplified aircraft. One of the methods is the least squares curve fitting. In (Shinbrot, 1951) Marvin Shinbrot describes about the application of the least squares method to the calculation of stability coefficients. Research and applications on aerodynamic modelling increased significantly in the 60s. The use of computers made it possible to measure and process large amounts of (flight test) data.

An aircraft can have highly non-linear behaviour, especially high performance aircraft such as the F-16 (Morelli, 1998) and the Eurofighter (Bava, Hoare, Garcia-Mesuro, & Oelker, 1999). Also in certain envelopes, for example stall, unsteady aerodynamic models are required. Overtime the necessity for non-linear aerodynamic models became clear and more research is performed in this field. According to (Jategaonkar, 2006) the process of deriving a model can be classified in two categories: phenomenological models and behavioural models. Phenomenological models require a high a-priori knowledge and can get complex. The models are derived from the Equations of Motion (EOM) and give physical insight of the dynamics of the system. This can simplify the validation of the model. The behavioural models are created by using the input and output without looking at the internal process. Less a-priori knowledge is required. This makes the model easier to identify, but much harder to verify. It also lacks a physical interpretation. This classification reflects on earlier research in modelling of non-linear aerodynamics. The latter classifications can be extended from a so called white-box model via a grey-box model to a black-box model. Where the white-box is phenomenological and the black-box the behavioural. Examples of black-box models involve neural-networks and fuzzy logics. The MSBS are considered a grey-box model because of the not straightforward physical interpretation. A physical transformation of the parameters can change the MSBS method into a white-box model. Another white-box model example is an aerodynamic model using (non-)linear polynomial basis functions which are linear in

model terms. Multiple polynomial models can be created and are connected with blending techniques. This results in large aerodynamic database tables. Together they cover a wide range of the flight envelope (Da Ronch, Ghoreyshi, & Badcock, 2011). A downside of the tables is the difficulty to physical interpret the dynamics of the aircraft (Müller, 2006).

System identification using fuzzy logics is described in (Takagi & Sugeno, 1985). Two possibilities exist in using fuzzy logics, the first is a fuzzy set composed of membership functions that mimics the process of decision making of a human. A disadvantages is that the approximated function is zero order continuous. The other approach is modelling based on multidimensional fuzzy reasoning. Multiple internal functions, mostly linear in parameters, are weighted using membership functions. The internal function creates the fuzziness of the model. The grading of the membership functions is on an interval from 0 to 1, where 0 is no influence and 1 full influence. Next cells are formed by using a membership function from each input variable. The output of the model is a weighted average of the output of all cells. Aerodynamic modelling using this approach is published in (Brandon, 1998) and (Brandon & Morelli, 2016). The membership functions blend the fuzzy logic internal polynomial functions into one model making the method capable for modelling non-linear dynamics. A problem is that this blending is non-trivial making the operation hard to automate and rely on expert knowledge. Also the physical interpretation is lost because of the black-box model type.

Neural networks is another example of a black-box model/ behavioural model. Again, the consequence is that the internal dynamics are unknown and no physical interpretation of the model is possible. It uses a general non-linear function approximator with global basis functions. Multiple layers with neurons are weighted to model a set of scattered data. The input layer is a combination of the state inputs and the control inputs. The output of the input layer, is the input of the hidden layer. Multiple hidden layers are possible with different number of neurons. The final layer is the output layer. Each neuron is a summation of the weighted inputs. This summation passes through a threshold function, often a sigmoid function. The sigmoid function has the property that it acts like a step-function while remaining differentiable. The derivative of a sigmoid function can be expressed in terms of the function itself. System identification with neural networks is an non-linear optimisation problem. An often used and successful training algorithm is the Levenberg-Marquardt Backpropagation algorithm (Yu, M., & Wilamowski, 2011). Except that it is a black-box model the neural networks approach is successfully used in (non-linear) aerodynamic modelling (Kirkpatrick, May Jr, & Valasek, 2013), control allocation tasks (Lee & Kim, 2001), aerodynamic design (Rai & Madavan, 2000), pattern recognition (Shahbandi & Lucidarme, n.d.) and mission/ trajectory planning.

All of the aforementioned system identification techniques have certain advantages and disadvantages. System identification capable of modelling high non-linear dynamics are often black-box models lacking a physical interpretation of the model. The global basis functions give them inner model continuity, but make them very computational intensive because of the non-sparse solution systems. A better and more general solution for modelling high non-linear aircraft is the Multivariate Simplex Splines (De Visser, 2011) and the corresponding physical transformation. Splines are piecewise polynomial functions. The MSBS is a general function approximator using local basis functions to fit a scattered dataset. It is compatible with linear regression methods. A minor disadvantage is that it

requires a geometric support structure called a triangulation. The local basis functions make them computational efficient and introduces the property to adjust the model locally. This system identification technique is successfully used in the aerodynamic modelling process for both linear (C. de Visser, Mulder, & Chu, 2010) and non-linear aerodynamic modelling (C. de Visser, Mulder, & Chu, 2009) and also in solving Partial Differential Equations (PDE) by transforming them into differential algebraic equations and control allocation tasks (Tol, Visser, Sun, Kampen, & Chu, 2016). Other possible application will be efficient flight envelope prediction and adaptive non-linear control. Also the physical transformation to Cartesian space, a formulation for global coordinates, is implemented in an interval analysis (C. C. de Visser, Kampen, Chu, & Mulder, n.d.). The triangulation creation is involved in the optimisation process, i.e. the triangulation doesn't have to be defined a-priori. The interspline formulation solves the triangulation optimisation by optimising the vertex locations of simplices. Recently a physical-splines formulation is developed. This transformation gives the MSBS and its coefficients a physical interpretation making it a promising method for non-linear aerodynamic modelling.

In the remainder of this chapter a mathematical description of the aircraft dynamics is derived and the aircraft to be modelled, the flying laboratory aircraft PH-LAB including its instruments, is introduced. Thereafter it is explained how flight-test measurements are processed and used for reconstructing the flight path. In the subsequent chapter the aerodynamic modelling process is elaborated using the MSBS process in combination with the new formulation of the physical transformation.

3-1 Equations of Motion

The aircraft model used in this thesis is governed by the EOM. They describe the dynamics of an aircraft. The EOM will be used to identify an aerodynamic model using a mathematical description of aerodynamic forces and moments. Newton's Second Law is used to this. In this chapter a brief explanation is given how the EOM are derived by giving the translational motion, the rotational motion with respect to the Earth-Centered, Earth-Fixed (ECEF) frame and the vehicle body frame. An overview of the reference frames is given in appendix F. Finally the attitude equations are derived. As mentioned before Newton's Second Law is used to derive the EOM. Newton's Second Law only holds in an inertial reference system and is given by

$$\mathbf{F} = \int d\mathbf{F} = \frac{d}{dt}\left(\mathbf{V}_p dm\right) \tag{3-1}$$

and for a constant mass m

$$\mathbf{F} = m\frac{d\mathbf{V}_G}{dt} = m\mathbf{A}_G \tag{3-2}$$

During the derivations it is assumed that the aircraft is a rigid body and the aircraft has a plane of symmetry.

3-1-1 Equations of Translational Motion

The general formulation of the equations of translational motion in the inertial frame with variable mass m subjected to external forces is given by

$$\mathbf{F}^I_{\text{ext}} = m\frac{d^2\mathbf{r}^I_{cm}}{dt^2} + 2\mathbf{\Omega}^I_{bI} \times \int_m \frac{\delta\tilde{\mathbf{r}}}{\delta t}dm + \int_m \frac{\delta^2\tilde{\mathbf{r}}}{\delta t^2}dm \tag{3-3}$$

The terms respectively in (3-3) represent the acceleration of the centre of mass with respect to the inertial-frame, the Coriolis force due to changing mass and the relative force due to changing mass. The $\frac{d}{dt}(\cdot)$ indicates a derivative of a vector quantity taken in inertial space and $\frac{\delta}{\delta t}(\cdot)$ indicates a derivative in the local frame. Using the principle of solidification, (e. a. Mulder J. A., 2000), equation (3-3) can be rewritten for a rigid body with mass m at time t using translational and rotational equations, given the true external forces and moments together with the Coriolis and relative forces and moments (the apparent forces and moments). Consider a vehicle, again with variable mass, moving with respect to the ECEF frame indicated by index C. It has a velocity $\mathbf{V}_C$ and a rotation w.r.t. the inertial plantocentric frame of $\mathbf{\Omega}^b_{bI}$. The distance to the centre of the central body is indicated with $\mathbf{r}^C_{cm}$ and the vehicle is subjected to external forces $\mathbf{F}^C_{\text{ext}}$ given by

$$\mathbf{F}^C_{\text{ext}} = m\frac{d\mathbf{V}^C_C}{dt} + 2m\mathbf{\Omega}^C_{CI} \times \mathbf{V}^C_C + m\mathbf{\Omega}^C_{CI} \times \left(\mathbf{\Omega}^C_{CI} \times \mathbf{r}^C_{cm}\right) \tag{3-4}$$

Equation (3-4) can be solved for the dynamic equations of translational motion in the the North-East-Down reference frame indicated by index E

$$\dot{V}_N = \frac{F^E_x}{m} - 2\Omega_t V_E \sin\delta - \Omega_t^2 R \sin\delta\cos\delta - \frac{V_E^2\tan\delta - V_N V_D}{R} \tag{3-5}$$

$$\dot{V}_E = \frac{F^E_y}{m} + 2\Omega_t \left(V_D\cos\delta + V_N\sin\delta\right) + \frac{V_E}{R}\left(V_N\tan\delta + V_D\right) \tag{3-6}$$

$$\dot{V}_D = \frac{F^E_z}{m} - 2\Omega_t V_E\cos\delta - \Omega_t^2 R\cos^2\delta - \frac{V_E^2 - V_N^2}{R} \tag{3-7}$$

In these equations the velocities north, east and down are given by $V_G = [V_N\ V_E\ V_D]$ respectively. The definition of spherical position is used, where R is the distance to the centre of the the central body and δ the latitude. The angular velocity of the central body is given by Ω_t. For Earth this is approximately 7.29×10^{-5} rad/s.

Flight test data is measured with different sensors and the sensors are attached to the aircraft. Hence the measurements are made in the body-frame. Therefore it is desirable to have the aforementioned translational equations solved in the body frame.

$$\mathbf{F}^b_{\text{ext}} = m\left.\frac{d\mathbf{V}_G}{dt}\right|_E^b = m\left.\frac{d\mathbf{V}_G}{dt}\right|_b^b + m\mathbf{\Omega}^b_{bE} \times \mathbf{V}_b \tag{3-8}$$

Where the velocity components in the body frame are given by $\mathbf{V}_b = [u\ v\ w]^T$ and the rotational components by $\mathbf{\Omega}^b_{bE} = [p\ q\ r]^T$. They represent the the X, Y and Z axis, of the aircraft, respectively. External forces subjected to the aircraft are aerodynamic forces $\mathbf{F}_{\text{aero}}$ and a gravitational force $\mathbf{F}_{\text{gravity}}$. The gravitational force is given in the ECEF reference frame and has to be transformed to the body-frame first.

$$\begin{aligned}\mathbf{F}^b_{\text{gravity}} &= \mathbb{T}_{bE} \cdot \mathbf{F}^E_{\text{gravity}} \\ &= \mathbb{T}_{bE} \cdot \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \\ &= \begin{bmatrix} -\sin\theta \\ \sin\phi\ \cos\theta \\ \cos\phi\ \cos\theta \end{bmatrix}\end{aligned} \tag{3-9}$$

The aerodynamic forces are given by $\mathbf{F}_{\text{aero}} = \begin{bmatrix} X^b\ Y^b\ Z^b \end{bmatrix}^T$. Substituting $\mathbf{F}_{\text{aero}}$ and (3-9) in equation (3-8) gives

$$\begin{bmatrix} -\sin\theta \\ \sin\phi\ \cos\theta \\ \cos\phi\ \cos\theta \end{bmatrix} + \begin{bmatrix} X^b \\ Y^b \\ Z^b \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \tag{3-10}$$

Solving for the forces in the body frame yields

$$X = m(\dot{u} + qw - rv) + mg\sin\theta \tag{3-11}$$

$$Y = m(\dot{v} + ru - pw) - mg\sin\phi\cos\theta \tag{3-12}$$

$$Z = m(\dot{w} + pv - qu) - mg\cos\phi\cos\theta \tag{3-13}$$

3-1-2 Equations of Rotational Motion

Next the moment equations, based on the moments acting on the aircraft with respect to the c.m., are determined. The general formulation of the total external moments acting on the c.m. of an aircraft in the body frame are given by

$$\begin{aligned}\mathbf{M}^b_{cm} = &\int_m \tilde{\mathbf{r}} \times \left(\frac{d\mathbf{\Omega}^b_{bI}}{dt} \times \tilde{\mathbf{r}} \right) dm + \int_m \tilde{\mathbf{r}} \times \left[\mathbf{\Omega}^b_{bI} \times \left(\mathbf{\Omega}^b_{bI} \times \tilde{\mathbf{r}} \right) \right] dm + \\ &+ 2\int_m \tilde{\mathbf{r}} \times \left(\mathbf{\Omega}^b_{bI} \times \frac{\delta\tilde{\mathbf{r}}}{\delta t} \right) dm + \int_m \tilde{\mathbf{r}} \times \frac{\delta^2\tilde{\mathbf{r}}}{\delta t^2} dm\end{aligned} \tag{3-14}$$

After solving the previous equations the so-called Euler equations are found. Without going into detail of deriving them they are given by

$$\dot{\mathbf{\Omega}}_{bI}^{b} = \mathbb{I}^{-1}\left(\mathbf{M}_{cm}^{b} - \mathbf{\Omega}_{bI}^{b} \times \mathbb{I}\mathbf{\Omega}_{bI}^{b}\right) \tag{3-15}$$

With $\mathbf{M}_{cm}^{b} = [L\ M\ N]^T$ and the rotation vector expressed in the body frame w.r.t inertial frames $\mathbf{\Omega}_{bI}^{b} = [p\ q\ r]^T$. The last term in equation (3-15) is the time derivative of the angular momentum of a vehicle and is defined in (3-16).

$$\mathbf{B}_{cm} = \mathbb{I}_{cm} \cdot \mathbf{\Omega}_{bE}^{b} \tag{3-16}$$

$\mathbb{I}$ is the inertia tensor of the vehicle given by

$$\mathbb{I}_{cm} = \begin{bmatrix} \int(r_y^2 + r_z^2)dm & -\int(r_x r_y)dm & -\int(r_x r_z)dm \\ -\int(r_x r_y)dm & \int(r_x^2 + r_z^2)dm & -\int(r_y r_z)dm \\ -\int(r_x r_z)dm & -\int(r_y r_z)dm & \int(r_x^2 + r_y^2)dm \end{bmatrix} = \begin{bmatrix} I_{xx} & -J_{xy} & -J_{xz} \\ -J_{yx} & I_{yy} & -J_{yz} \\ -J_{zx} & -J_{zy} & I_{zz} \end{bmatrix} \tag{3-17}$$

Applying the assumption an aircraft has a plane of symmetry in the $X_b\,Z_b$ plane, hence $J_{xy} = J_{yx} = 0$ and $J_{yz} = J_{zy} = 0$. Equation (3-17) reduces to

$$\mathbb{I}_{cm} = \begin{bmatrix} I_{xx} & 0 & -J_{xz} \\ 0 & I_{yy} & 0 \\ -J_{zx} & 0 & I_{zz} \end{bmatrix} \tag{3-18}$$

Substituting equations (3-18) in (3-15) gives

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & -J_{xz} \\ 0 & I_{yy} & 0 \\ -J_{xz} & 0 & I_{zz} \end{bmatrix}^{-1} \cdot \left\{ \begin{bmatrix} L \\ M \\ N \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & -J_{xz} \\ 0 & I_{yy} & 0 \\ -J_{xz} & 0 & I_{zz} \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right\} \tag{3-19}$$

The latter matrix equation is solved for the moments acting on the c.m. of the vehicle given in the body frame

$$L = I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - J_{xz}(pq + \dot{r}) \tag{3-20}$$

$$M = I_{yy}\dot{q} + (I_{xx} - I_{zz})pr + J_{xz}(p^2 - r^2) \tag{3-21}$$

$$N = I_{zz}\dot{r} + (I_{yy} - I_{xx})pq + J_{xz}(qr - \dot{p}) \tag{3-22}$$

3-1-3 Dimensionless Forces and Moments

The forces and moment equations can be made dimensionless. This gives a better understanding and interpretation of the stability derivatives, because they become dimensionless as well. The dimensionless form for the force equations is defined by

$$C_X = \frac{X}{\frac{1}{2}\rho V^2 S} = \frac{ma_X - T_x}{\frac{1}{2}\rho V^2 S} \tag{3-23}$$

$$C_Y = \frac{Y}{\frac{1}{2}\rho V^2 S} = \frac{ma_Y}{\frac{1}{2}\rho V^2 S} \tag{3-24}$$

$$C_Z = \frac{Z}{\frac{1}{2}\rho V^2 S} = \frac{ma_Z}{\frac{1}{2}\rho V^2 S} \tag{3-25}$$

and the dimensionless moment equations by

$$C_l = \frac{L}{\frac{1}{2}\rho V^2 S\bar{b}} = \frac{\dot{p}I_{xx} + qr\left(I_{zz} - I_{yy}\right) - \left(pq + \dot{r}\right)J_{xz}}{\frac{1}{2}\rho V^2 S b} \tag{3-26}$$

$$C_m = \frac{M}{\frac{1}{2}\rho V^2 S\bar{c}} = \frac{\dot{q}I_{yy} + rp\left(I_{xx} - I_{zz}\right) + \left(p^2 - r^2\right)I_{xz}}{\frac{1}{2}\rho V^2 S\bar{c}} \tag{3-27}$$

$$C_n = \frac{N}{\frac{1}{2}\rho V^2 S b} = \frac{\dot{r}I_{zz} + pq\left(I_{yy} - I_{xx}\right) + \left(qr - \dot{p}\right)J_{xz}}{\frac{1}{2}\rho V^2 S b} \tag{3-28}$$

3-2 Flight Test Data

The aircraft to be modelled is the Cessna Citation II. The aircraft model in this project is governed by the EOM and are used to identify the aerodynamic model describing the dynamics of the aircraft. Experimental flights are performed using different types of manoeuvres, such as the 3211 and doublet manoeuvres. They are described in (J. A. Mulder, 1986). During these experimental flights the flight-test data is recorded. In appendix G the flight envelope of the Cessna Citation II is given including the available flight test data at the moment of writing.

3-2-1 PH-LAB

The aircraft used to perform test flights and collect measurements is a Cessna Citation II, model 550 aircraft, see Figure 3-1. The flight test data will be used to create a full flight envelope aerodynamic model of this aircraft using P-Splines. The Cessna Citation II is a small business jet developed in the 70s and upgraded in the 90s. Almost 1000 Citation II are built. The Cessna 550 used for the test-flights is the PH-LAB and is co-owned by Delft University of Technology (DUT) and the Netherlands Aerospace Centre (NRL). The aircraft was built in 1993 and is converted so it can be used as a flying laboratory. It has a modern glass

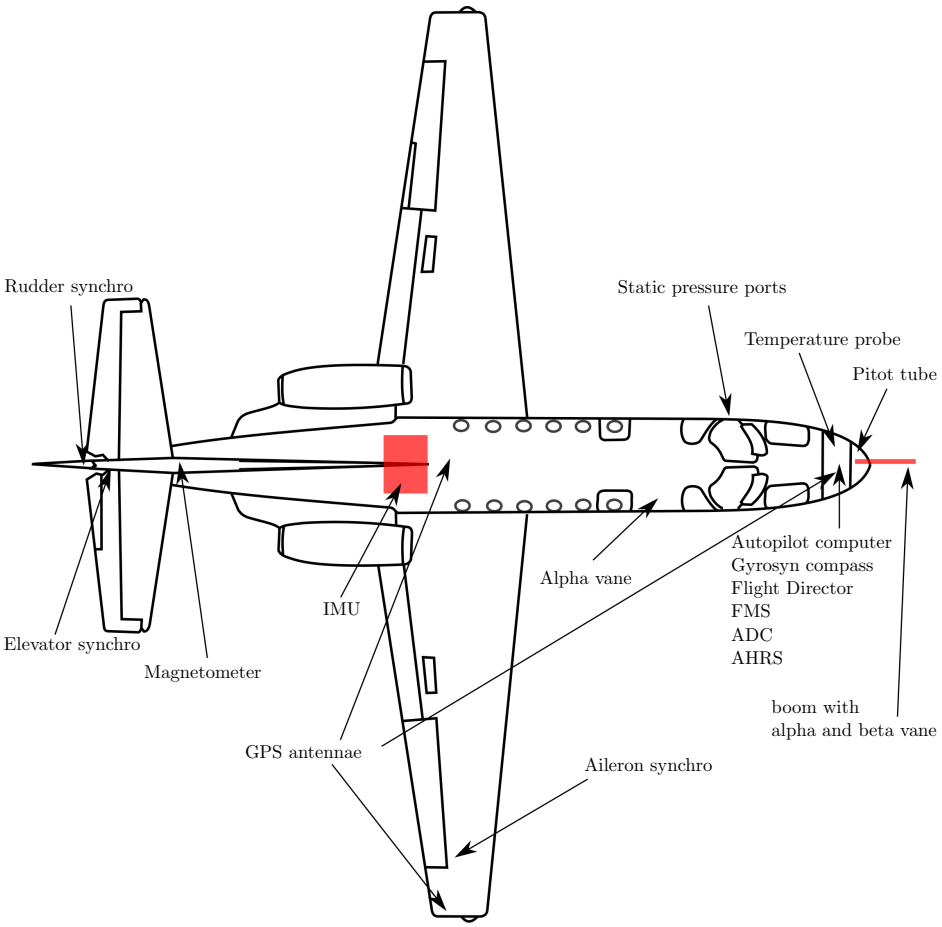


Figure 3-1: PH-LAB Cessna Citation II aircraft

cockpit and fly-by-wire flight controls. Flight test data is measured using special high-fidelity flight test instruments such as vertical gyroscopes for measuring axis-rotation, rate gyroscopes measuring axis rotational rates, Inertial Measurement Unit (IMU) measuring accelerations, static-probe for the altitude, pitot-static probes for the true airspeed, Attitude and Heading Reference System (AHRS), GPS for location and ground-speed, alpha vane for the angle of attack and other aircraft systems via the Flight Management System (FMS). Furthermore a boom is available for better measurements of the angle of attack and the side-slip angle. The instruments including their specifications are listed in Table 3-2. The properties and performance of the PH-LAB are tabulated in Table 3-1.

3-2-2 Flight-Path Reconstruction

The measurements are stored in a Hierarchical Data Format (HDF5) and are converted to a Matlab data file, a .mat file, using the software HEFTIG. An earlier developed toolbox by the task-force is used to reconstruct the flight-path using these measurements. This toolbox pre-processes the data first. Table 3-2 shows that instruments have different sampling rates. Hence re-sampling is required for unification. Estimations of the inertia tensors Ixx, Iyy, Izz are taken into account and also the aircraft empty mass, fuel mass, passenger (pax) mass and

Table 3-1: Aircraft details of PH-LAB

PH-LAB	
Manufacturer	Cessna
Type	Citation II Model 550
Dimensions	
Length	14.39 m
Wingspan b	15.90 m
Wing Area S	30.00 m^2
Wing Chord Length $\bar{c}$	2.06 m
Height	4.57 m
Mass Properties	
Empty Weight	3,906 kg
Maximum Payload	1,400 kg
Maximum Fuel Load	2,250 kg
Maximum Takeoff Weight	6,715 kg
$I_{xx,\text{empty}}$	12,392 kg·m^2
$I_{yy,\text{empty}}$	31,501 kg·m^2
$I_{zz,\text{empty}}$	41,908 kg·m^2
$I_{xz,\text{empty}}$	2,252 kg·m^2
Performance	
Engine	2× Pratt & Whitney JT15D-4 turbofan
Maximum Thrust	2×11.1 kN
Maximum Cruise Speed	710 km/h
Maximum Operating Altitude	13,000 m

Table 3-2: Instrumentation of the PH-LAB

Parameter	**Unit**	**F_s [Hz]**	**1σ std**	**Source**
X_b-axis rotation	[rad]	1	8.70×10^{-3}	Sperry Vertical Gyro
Y_b-axis rotation	[rad]	50	8.70×10^{-3}	Sperry Vertical Gyro
Z_b-axis rotation	[rad]	50	1.73×10^{-3}	Gryosyn Compass
X_b-axis rotational rate	[rad/s]	100	2.00×10^{-3}	LITEF μFORS Rate Gyro
Y_b-axis rotational rate	[rad/s]	100	2.00×10^{-3}	LITEF μFORS Rate Gyro
Z_b-axis rotational rate	[rad/s]	100	5.00×10^{-3}	LITEF μFORS Rate Gyro
X_b-axis linear acceleration	$[m/s^2]$	100	2.00×10^{-2}	Q-Flex 3100 Accelerometer
Y_b-axis linear acceleration	$[m/s^2]$	100	2.00×10^{-2}	Q-Flex 3100 Accelerometer
Z_b-axis linear acceleration	$[m/s^2]$	100	3.00×10^{-2}	Q-Flex 3100 Accelerometer
Altitude	[m]	1	3.00×10^{-1}	Static Probe
True Airspeed	[m/s]	50	1.00×10^{-1}	Pitot-Static Probe
Angle of Attack	[rad]	1000	3.50×10^{-3}	Alpha Vane
Angle of Sideslip	[rad]	1000	3.50×10^{-3}	Beta Vane

cargo mass. For the fuel mass it is important to know the rate of fuel burn, including the fuel-tank dimensions and location. Moments caused by pax and cargo have to be corrected. For the flight-path reconstruction an Unscented Kalman Filter (UKF) is used. The UKF has good results for systems with moderate to high non-linearities. The states estimates are improved in terms of accuracy and robustness by using this filter. The author refers to (Hoek, Visser, & Pool, 2017) for more background information and results about the flight-path reconstruction and the UKF. At the moment of writing flight-test data of multiple experimental flights is available, including a test-flight with 38 stalls.

3-3 Simulation Framework

The software used for this thesis is Matlab including additional toolboxes, for example the aerospace toolbox, and toolboxes created by the C&S department. For simulation Simulink is used. The TU Delft provides an academic Matlab licence. The C&S department uses the simulation framework DASMAT (Linden, 1998). DASMAT is divided in multiple blocks: airdata, wind model, gravity model, EOM, aerodynamic forces and moments, engine forces and moments, landing gear model and observation model. Despite some deficiencies it forms a good basis for a simulation framework. The task-force earlier performed a feasibility study of upgrading DASMAT where an overhaul importance label was assigned to the individual blocks. At the moment of writing a start was made and some blocks are upgraded using a preferable embedded Matlab function representation. In this thesis the new simulation framework is used and the P-Splines model will update the aerodynamic forces and moments block.
For the MSBS a Matlab toolbox was created by the C&S Department. As part of the modelling process the toolbox creates a triangulation based on the scattered dataset and the physical-splines transformation.

Chapter 4

Aerodynamic Modelling with Physical-Splines

This chapter continues with the aerodynamic modelling process while using the MSBS for system identification in combination with the reconstructed scattered flight-test data. The earlier explained physical transformation is used and added to this process. The properties of the physical transformation are investigated and their contribution to the modelling process is elaborated.

Creating a model structure for regression is an important step in the system identification process. This mathematical structure should describe the aerodynamic forces and moments explained in 3-1 accurately. However it is not always evident what the form of this model structure is and what terms to include or exclude. Adding more terms can increase the approximation power of the model, but also lead to an over-fit when turbulence or noise in the dataset are modelled. This reduces the accuracy. Also the dimension of the model, what states/ variables to include in the model, determines the model complexity. Especially high dimensional simplex triangulation used in spline models let the model complexity explode. A citation by Tarentol (Tarantola, 2005): *"There is one problem with large-dimensional spaces that is easy to underestimate: they tend to be terribly empty."* For a high dimensional models ($n > 10$) millions of data points should be available.

Prototyping and system identification using the MSBS method is divided in three phases: geometric model structure selection, polynomial model structure selection and model estimation and validation. This iterative process is repeated with different model settings until an optimal or adequate model is found. In the following sections this process of prototyping is explained and the physical-splines formulation is used to merge new and existing aerodynamic modelling techniques into the simplex spline modelling process. An overview of this process is summarised in a chart appended to this preliminary report in appendix H.

4-1 Geometric Model Structure Selection

In this part of the process the model dimension is determined by selecting an appropriate candidate set that is capable of identifying the model. With the candidate set and dimension known the geometric support structure (triangulation) is created.

4-1-1 Candidate Sets

A general definition in aerodynamic modelling is that the aerodynamic forces and moments are a function of the state variables and the control variables δ. The variables used for modelling the six different force and moment equations are given by (4-1).

$$\begin{aligned} C_i &= \left(M,\ h,\ \alpha,\ \hat{\dot{\alpha}},\ \beta,\ \hat{\dot{\beta}},\ \hat{p},\ \hat{q},\ \hat{r},\ \delta\right) \text{ for } i = X,\ Y,\ Z,\ l,\ m,\ n \\ \delta &= (\delta_a,\ \delta_e,\ \delta_r) \end{aligned} \tag{4-1}$$

In this equation M $(-)$ is the Mach number, h (m) the altitude, α (rad) the angle of attack, β (rad) the side-slip angle and p, q, r (rad/s) the rotational rates in the body-frame. δ_a (rad), δ_e (rad), δ_r (rad) are the aileron, elevator and rudder deflections respectively. The dot indicates a time derivative. For consistency the state variables are made dimensionless (indicated with a hat symbol) by using the definition

$$M \equiv \frac{V}{a_0},\ \hat{p} \equiv \frac{pb}{V},\ \hat{q} \equiv \frac{q\bar{c}}{V},\ \hat{r} \equiv \frac{rb}{V},\ \hat{\dot{\alpha}} \equiv \frac{\dot{\alpha}\bar{c}}{V},\ \hat{\dot{\beta}} \equiv \frac{\dot{\beta}b}{V} \tag{4-2}$$

Determining the candidate set is part of the geometric model structure selection. Creating an aerodynamic model requires flight test data with measurement in the complete flight envelope. The number of measurements needed, depend on the non-linear behaviour of the aircraft and the size of the flight envelope. By taking all state and control variables in consideration all combinations within the flight envelope have to measured. This is a high dimensional space and requires many flight hours. In order to decrease the large multidimensional space a commonly used and accepted assumption, based on an aircraft symmetry, is to decouple the aerodynamic model into a longitudinal (symmetrical) and lateral (asymmetrical) model. No correlation between the longitudinal and lateral models exist any more. The symmetric longitudinal forces and moment are (X,Z and m) and the asymmetric lateral force and moments are (Y and l, n). This is tabulated in 4-1 in combination with the corresponding state and control variables. This assumption can only be made for smaller amplitude disturbances and deviations. For larger amplitude both angle of attack and slide-slip angle should be considered in the candidate set for the longitudinal and lateral models.

A candidate set can be determined with multiple methods. The first assumption is that the candidates, the state and control variables, do have a significant meaning to the dynamics of the Cessna Citation II. Hence it should make sense to use them for the system identification process. A candidate set for the Cessna Citation II was created in (De Visser, 2011). The candidates and therefore the dimension of the models were determined using a method called the occurrence of hysteresis. In this research thesis the obtained candidate sets will be used.

Table 4-1: Longitudinal and Lateral Model

Model	Limited to moderate non-linearities	Significant non-linearities
Longitudinal (X, Z, m)	$M,\ h,\ \delta_e,\ T_e$	$\hat{\alpha},\ \hat{\dot{\alpha}}, \hat{q}$
Lateral (Y, l, n)	$M,\ h,\ \delta_a,\ \delta_r$	$\hat{\beta},\ \hat{\dot{\beta}},\ \hat{p},\ \hat{r}$

4-1-2 Triangulation

A triangulation is based on multiple n-dimensional simplices. The simplices are ordered in such a way they form a n dimensional rectangle while the n dimensional faces of neighbouring simplices are aligned. The multivariate simplex spline polynomials will be defined on these simplices. A triangulation is created around the convex hull of the data-points used for identification, i.e. all identification data-points are in the convex hull of the triangulation $\mathcal{T}$. In order to determine if a single data point x is within a certain simplex t_j the barycentric coordinates, described in (2-13), are used with the simple rule

$$\begin{aligned} &b_i > 0 \rightarrow \mathbf{x} \in t_j,\ \forall i \\ &b_i < 0 \rightarrow \mathbf{x} \notin t_j, 0 \leq i \leq n \end{aligned} \tag{4-3}$$

Creating a triangulation is a non-trivial process and many possibilities exist. The triangulation can be created based on the aerodynamic behaviour of the aircraft. More information about functions on 3D simplices and multivariate simplices is found in (Lai & Schumaker, 2007). An inter-spline method for optimising the triangulation is researched in (C. C. de Visser et al., n.d.). The shape of a simplex within a triangulation does also have influence on the final model and can cause numerical problems. One simplex metric is called sliver. A sliver matrix has a high circumsphere radius compared to the shortest edge. Also the smallest angle between two edges is very low. The simplex will have a high skewness value and expands a much larger domain for some dimensional directions compared to other dimensional directions and other simplices. The process of creating an optimised triangulation is beyond the scope of this thesis. Therefore from this point onward the triangulation is based on a single or multiple attached hyperrectangles. This assumption has a postive effect that slivered simplices do not occur within the triangulation.

With the triangulation $\mathcal{T}$ defined a convenient notation to use for a spline space according, (De Visser, 2011), is

$$\mathcal{S}_d^r(\mathcal{T}) := \{s \in C^r(\mathcal{T}) : s|_t \in \mathcal{P}_d,\ \forall t \in \mathcal{T}\} \tag{4-4}$$

where s is a n-simplex spline function with degree d and continuity of order r on triangulation $\mathcal{T}$. The space of all polynomials with degree d is defined by $\mathcal{P}$.

Table 4-2: n-hyperrectangle Properties

n	n-simplices	aligned (n-1)-faces
1	1	0
2	2	1
3	6	6
4	24	36
5	120	240
6	720	1800
7	5040	15120

4-1-3 Hyperrectangle

As mentioned in the previous section, a triangulation will be created within a hyperrectangle. A hyperrectangle is an n-dimensional analogue of a rectangle ($n = 2$). Multiple equal sized n-simplices will fill the hyperrectangle while the faces of neighbouring simplices keep alignment to each other. The last property is very important to guarantee continuity between the splines-model and not breaking the splines geometric structure. The process used for creating n-hyperrectangles with n-simplices is the Kuhn Triangulation (Kuhn, 1960). An alternative algorithm is the Delaunay triangulation. In many programming languages this algorithm uses the Qhull c-library. However it was found that a bug creates an incorrect non-simplicial facets for adjacent simplices, i.e. the faces of neighbouring simplices are not aligned properly when the dimension is $n \geq 3$. This bug is reported end 2006 and is at the moment of writing (2017) not solved. How to fix this bug is still unknown.

Multiple hyperrectangles can be joined together in any of the n-dimensions. The faces of simplices aligned with the hyperrectanlge boundaries are equally aligned to other hyperrectangle boundary simplex faces, creating continuity between simplices of different hyperrectangles. In this way a more complex triangulation can be created in the dimension of one of the variables. In Figure 4-1 and Figure 4-2 examples of hyperrectangle triangulations are visualised.

The hyperrectangle-assumption does introduce a problem. The number of simplices within the hyperrectangle expands with $n!$. Higher dimension models will have a large number of simplices and require many data points. High dimensions become large voids! In table 4-2 the number of simplices and number of aligned faces, of different n-dimensional triangulations are tabulated.

The hyperrectangle dimensions and position in Cartesian Space are determined by the maximum and minimum values of the identification dataset of each variable included in the model. The identification data points are in the convex hull of the simplices and therefore the hyperrectangle. This way the triangulation spans the whole domain of measured flight envelope flight data. The latter statement is only valid when the density of flight test data is sufficient enough in combination with an evenly spatial data distribution throughout the domain of the triangulation. However in real-life this is not always the case. Measurements in extreme flight conditions are mostly sparse and are located in remote, boundary areas of the flight envelope. The low density can cause some simplices to be empty or near empty. This will create poor performing simplices or unwanted inner-model discontinuities.

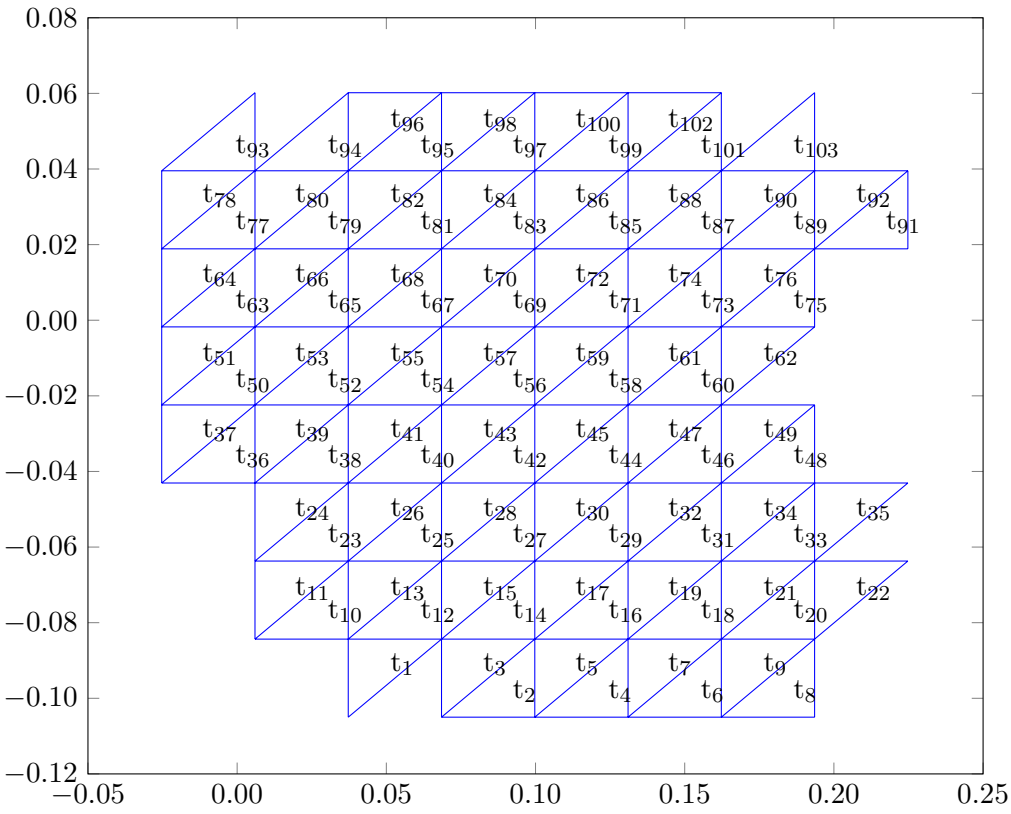


Figure 4-1: An example of multiple two dimensional hyperrectangles forming together a triangulation

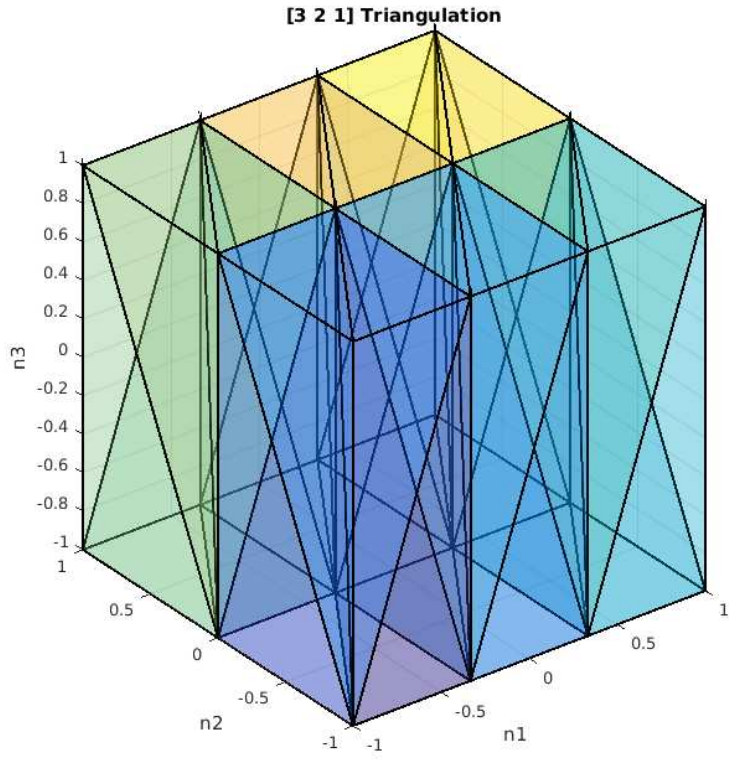


Figure 4-2: A three dimensional triangulation with a $3 \times 2 \times 1$ combined hyperrectangles

4-1-4 Data & Optimising the Triangulation

As described in 4-1-2 finding an optimal triangulation is not a trivial process. The orientation and size of the simplices is limited to the number of hyperrectangles. Ideally data points are scattered nice and evenly over a simplex. In a non-optimal triangulation data-points within a simplex can be clustered or are scattered non-uniformly. On the other hand simplices can also be empty because of voids created by outliers introducing inner-model discontinuity. This means that empty simplices at boundaries or within the triangulation can exist. Also some simplices, after validation, can have a poor performance. A solution is to delete outliers decreasing the domain size or find clusters within the dataset and create separate models for each cluster. To visualise flight-test data for high dimensional triangulations a matrix-plot can be created. This $n \times n$ matrix-plot visualises a pair-wise relation between the state and control variables included in the model. A scatter plot of the data-points gives an indication of the distribution of the data measured in this dimension. The diagonal plots show the variables univariate histogram of the distribution of the measurements in the measured domain. This way flight envelope regions can be indicated very easily as well the regions with missing or badly distributed measurements. Also searching for clusters in the flight-test data or the individual state and control variables can be helpful to get more insight of the flight-test data and triangulation quality. A cluster algorithm such as the k-means, (Jain, 2010), can be used for this. Validating the cluster is very important, because clusters can be identified while they are not convincing. A silhouette as described in (Rousseeuw, 1987) can be used for this. The silhouette plot shows how well the measurements belong to a certain cluster based on their tightness and separation, i.e. which state or control variable clusters the dataset. Understanding the flight-test data working with is important for a better understanding of the final results. Using and interpreting the data is not a trivial task, while having a large influence on the aerodynamic model created. The remainder of this section will introduce methods that can be used in situations triangulation problems occur because of the flight-test data

Hyperrectangle Rotation

In this section a method is introduced that can increase the performance of the triangulation by making use that a n-hyperrectangle does not have n-axis of symmetry, hence when performing for example $\tau = 90$ degrees planar rotations on the coordinates of vertices of the triangulation, a new rotated triangulation is obtained. One can see this rotation as a transformation of the vertices of the simplices and therefore a rotation of simplices within the triangulation. The transformation matrix for a planar rotation spanned by dimensions n_1 and n_2 is given by

$$\bar{\boldsymbol{\nu}}_{n_{1,2}} = \begin{bmatrix} cos(\tau) & -sin(\tau) \\ sin(\tau) & cos(\tau) \end{bmatrix} \cdot \boldsymbol{\nu}_{n_{1,2}} \tag{4-5}$$

where $\boldsymbol{\nu}$ are the vertices with coordinates in the plane spanned by $\{n_1, n_2\}$. Simplices in the triangulation will be rotated relative to dimensions n_1 and n_2 with an angle of τ. The triangulation is normalised first between -1 and 1 to make sure that the hyperrectangle has a symmetric/ rectangular shape and the middle of the hyperrectangle is aligned with the

Table 4-3: Number of planes per dimension

Dimension n	2	3	4	5	6
Number of planes	1	1	2	2	3

origin, hence no translation is required. After the rotation the normalisation is undone. The result is that only a transformation is performed on the vertices making the simplices rotate within a hyperrectangle. A planar rotation has the property that the plane is mapped to itself. The number of planes per dimension n is given by equation (4-6) and in table 4-3 this is summarised for dimension 2 to 6.

$$\left\lfloor \frac{n}{2} \right\rfloor \tag{4-6}$$

Together with the normalisation, the orthogonality between the planes make sure that a partial coordinates rotation of vertices in the dimensions spanning the plane does not influence any coordinates of vertices in other dimensions. Rotations of 90 degrees can be applied to the triangulation in search for an optimal orientation. The rotation can be repeated in other dimensions until a satisfactory orientation of the triangulation is found. Rotations for other angles τ, except 90 , 180, 270 degrees, changes the orientation of the convex hull of the triangulations and can cause identification data-points to be outside of the triangulation convex hull. The optimal orientation can be subjected to different objectives. One objective for an optimal orientation is where the most simplices are filled. Another objective is to find an orientation where the data is evenly distributed over the different simplices. Afterwards with an algorithm the number of adjacent $n-1$ faces of the simplices can be counted in order to verify a successful rotation. An interesting side note is to think about the situation when performing 1-dimensional transformations to vertices on hyperrectangle boundaries. Effectively this means that the convex hull of the triangulation can be adjusted to remove voids. This can be visualised with the matrix-plot introduced earlier.

Extrapolation

As described in the previous section, a problem to expect is that the distribution of data within the hypercube triangulation is not ideal. Missing measurements are mainly caused because of physical limitations of the aircraft and aviation restrictions. Physical limitations include parts of the flight envelope that introduces an increased safety risk, for example stall. Aviation restrictions can be air traffic control related, for example speed and altitude restrictions. For the overall increase of model quality it can sometimes be better to delete these simplices and extrapolate the model in these regions afterwards or interpolate when more measurements are available. Extrapolation techniques can make use of neighbouring simplices or by using differential constraints (C. C. de Visser, Chu, & Mulder, 2011). Also states, often the velocity and altitude, with less influence on the dimensionless force and moment equations can be made global model terms. They are removed from the splines formulation reducing the dimension of the triangulation.

4-2 Parameter Estimation

For a better understanding of the polynomial model structure selection phase of the modelling process, the parameter estimation section is elaborated first. The estimation of the parameters is an important step in a system identification process. For aerodynamic modelling the coefficients are determined using the flight test measurements and sub-sequentially they can be used for observing the stability derivatives when they are transformed to physical-coefficients. For now the assumption is made that a certain P-Splines polynomial model structure is selected using the selection procedure elaborated that will be elaborated in the next section.

Recalling the multinomial theorem used for MSBS, for convenience repeated in (4-7). The terms are formed by the multinomial expansion of the polynomial in this equation. The polynomial is in Barycentric space. Each term will have a multinomial coefficient and a corresponding parameter to be estimated. The multinomial coefficient and parameter form after estimation the coefficient of a term. Note that when $n = 1$ equation (4-7) equals the binomial theorem and the binomial coefficients are given by the famous Pascal's triangle. Hence the multinomial theorem can be interpreted as the generalised form of Pascal's triangle.

$$p(x) = \sum_{j=1}^{t_j} \sum_{|\kappa|} \left(c_{|k|} \frac{d!}{\kappa_0! \kappa_1! \dots \kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \dots b_n^{\kappa_n} \right)_{t_j} \quad \forall \; x \; \in \; t_j$$

with (4-7)

$$d \geq 0$$
$$(\kappa_0, \kappa_1, \dots, \kappa_n) \in \mathbb{N}$$
$$|k| = \kappa_0 + \kappa_1 + \dots + \kappa_n = d$$

and with the Boor's theorem any physical polynomial $p(x)$ can be expressed in the B-form. The short-hand notation of (4-7) is given by (4-8)

$$\mathbf{p}(x) = \mathbf{B}^d(\mathbf{b}_{t_j}(x)) \cdot \mathbf{c}^{t_j} \tag{4-8}$$

In (4-9) an example of a two dimensional $n = 2$ and third degree $d = 3$ expansion of a B-form polynomial is given for a single simplex t_j. The spatial location of the B-coefficients is shown in Figure 4-3.

$$\begin{aligned} \mathbf{B}^3(\mathbf{b}_{t_j}(x_1, x_2)) \cdot \mathbf{c}^{t_j} =& c_{300} b_0^3 + 3c_{210} b_0^2 b_1 + 3c_{201} b_0^2 b_2 + 3c_{120} b_0 b_1^2 + 6c_{111} b_0 b_1 b_2 \\ &+ 3c_{102} b_0 b_2^2 + c_{030} b_1^3 + 3c_{021} b_1^2 b_2 + 3c_{012} b_1 b_2^2 + c_{003} b_2^3 \end{aligned} \tag{4-9}$$

The polynomial structure for each simplex t_j is initially determined by the dimension n and degree d in barycentric space and later on by the polynomial model structure phase in the equivalent representation of this polynomial in physical space. The formed B-net structure is very important and gives the unique spatial location property to the B-coefficients. With

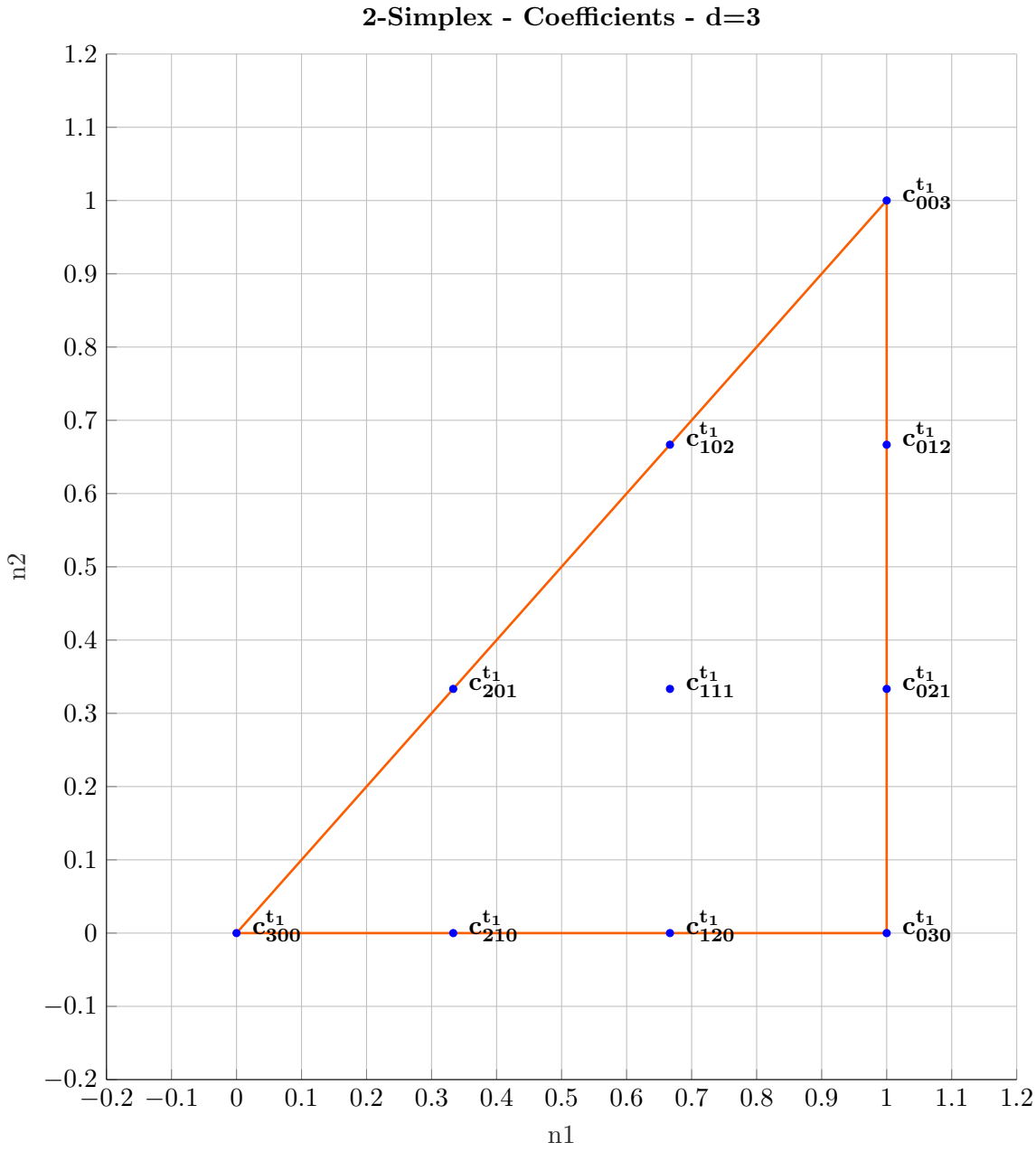


Figure 4-3: A single 2-simplex with a degree $d = 3$ B-form polynomial. The B-coefficients, ten in total, have a unique spatial location in the barycentric space.

the flight-test data the so-called B-form matrix of a simplex can be created. Each column represents a model term (one of the basis functions). The number of rows is determined by the number of measurements M. It is very important all model terms are presented and ordered correctly. If otherwise the B-net will break and modelling fails. This makes the splines system identification technique different to other techniques using linear regression methods. The B-matrix is given by (4-10)

$$\mathbf{B}^d(\mathbf{b}_{t_j}(x)) = \begin{bmatrix} \mathbf{B}^d_{d,0,0}(\mathbf{b}_{t_j}(x_{1_{t_j}})) & \mathbf{B}^d_{d-1,1,0}(\mathbf{b}_{t_j}(x_{1_{t_j}})) & \dots & \mathbf{B}^d_{0,1,d-1}(\mathbf{b}_{t_j}(x_{1_{t_j}})) & \mathbf{B}^d_{0,0,d}(\mathbf{b}_{t_j}(x_{1_{t_j}})) \\ \mathbf{B}^d_{d,0,0}(\mathbf{b}_{t_j}(x_{2_{t_j}})) & \mathbf{B}^d_{d-1,1,0}(\mathbf{b}_{t_j}(x_{2_{t_j}})) & \dots & \mathbf{B}^d_{0,1,d-1}(\mathbf{b}_{t_j}(x_{2_{t_j}})) & \mathbf{B}^d_{0,0,d}(\mathbf{b}_{t_j}(x_{2_{t_j}})) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{B}^d_{d,0,0}(\mathbf{b}_{t_j}(x_{M_{t_j}})) & \mathbf{B}^d_{d-1,1,0}(\mathbf{b}_{t_j}(x_{M_{t_j}})) & \dots & \mathbf{B}^d_{0,1,d-1}(\mathbf{b}_{t_j}(x_{M_{t_j}})) & \mathbf{B}^d_{0,0,d}(\mathbf{b}_{t_j}(x_{M_{t_j}})) \end{bmatrix} \tag{4-10}$$

and the b-coefficient vector to be estimated by (4-11)

$$\mathbf{c}_b^{t_j} = \begin{bmatrix} \mathbf{c}^{t_j}_{d,0,0} & \mathbf{c}^{t_j}_{d-1,1,0} & \dots & \mathbf{c}^{t_j}_{0,1,d-1} & \mathbf{c}^{t_j}_{0,0,d} \end{bmatrix}^T \tag{4-11}$$

The B-form matrix created for all simplices t_j can be written in a single sparse regression matrix. The regression matrix is formed by (4-12) and the coefficients to be estimated by (4-13). The subscript b indicates that the regression matrix is in barycentric space and the B-coefficients are estimated. The regression matrix $\mathbf{X}_b$ has a sparse solution system. Performing parameter estimation in barycentric space is favourable over parameter estimation in physical space. Transforming the regression matrix to physical space using the transformation described in Chapter 2-5-5 increases the condition number significantly introducing numerical errors. The better condition number of $\mathbf{X}_b$ is because the Barycentric coordinate system is a normalised coordinate system.

$$\mathbf{X}_b = \begin{bmatrix} \mathbf{B}^d(\mathbf{b}_{t_1}(x_{t_j})) & 0 & 0 & 0 \\ 0 & \mathbf{B}^d(\mathbf{b}_{t_2}(x_{t_2})) & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{B}^d(\mathbf{b}_{t_j}(x_{t_j})) \end{bmatrix} \tag{4-12}$$

$$\boldsymbol{\theta}_b = \begin{bmatrix} \mathbf{c}_b^{t_1} & \mathbf{c}_b^{t_2} & \dots & \mathbf{c}_b^{t_j} \end{bmatrix}^T \tag{4-13}$$

The matrix and vector in (4-12) and (4-13) above is written in a shorter form in equation (4-14) and is called the regression model. With system identification the parameters $\boldsymbol{\theta}_b$ are estimated using the regression matrix $\mathbf{X}_b$ (with independent columns) and a model output vector $\mathbf{y}$. The error, or residuals, is given by $\boldsymbol{\epsilon}$ and is defined by the difference between the output $\mathbf{y}$ and the modelled output $\hat{\mathbf{y}}$. During regression the residuals are minimised. The hat, $\hat{}$, indicates estimated values.

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{X}_b \hat{\boldsymbol{\theta}}_b \\ \mathbf{y} &= \hat{\mathbf{y}} + \boldsymbol{\epsilon} \\ \mathbf{y} &= \mathbf{X}_b \hat{\boldsymbol{\theta}}_b + \boldsymbol{\epsilon} \end{aligned} \tag{4-14}$$

4-2-1 Ordinary Least Squares

Remember the MSBS method is compatible with linear regression methods. A famous linear regression method to solve for the parameters $\boldsymbol{\theta}$ is the Ordinary Least Squares (OLS) method. In this method a cost function J will sum the residuals squared and minimises it to find the solution. A quick derivation is given.

$$\begin{aligned} J &= ||\mathbf{y} - \hat{\mathbf{y}}||_2 \\ &= ||\mathbf{y} - \mathbf{X}_b\hat{\boldsymbol{\theta}}_b||_2 \\ &= \frac{1}{2}\left(\mathbf{y} - \mathbf{X}_b\hat{\boldsymbol{\theta}}_b\right)^T \cdot \left(\mathbf{y} - \mathbf{X}_b\hat{\boldsymbol{\theta}}_b\right) \end{aligned} \tag{4-15}$$

The optimisation function J is linear in the parameters and can be minimised by solving for the derivative equal to zero as follows.

$$\begin{aligned} \min_{\theta} &= ||\mathbf{y} - \mathbf{X}_b\hat{\boldsymbol{\theta}}_b||_2 \\ \min_{\theta} &= \frac{1}{2}\hat{\boldsymbol{\theta}}_b^T \cdot \left(\mathbf{X}_b^T \cdot \mathbf{X}_b\right) \cdot \hat{\boldsymbol{\theta}}_b - \hat{\boldsymbol{\theta}}_b^T \cdot \mathbf{X}_b^T \cdot \mathbf{y} + \frac{1}{2}\mathbf{y}^T \cdot \mathbf{y} \\ \frac{\partial J}{\partial \hat{\theta}_b} &= \left(\mathbf{X}_b^T \cdot \mathbf{X}_b\right) \cdot \hat{\boldsymbol{\theta}}_b - \mathbf{X}_b^T \cdot \mathbf{y} = 0 \\ \hat{\boldsymbol{\theta}}_b &= \left(\mathbf{X}_b^T \cdot \mathbf{X}_b\right)^{-1} \cdot \mathbf{X}_b^T \cdot \mathbf{y} \end{aligned} \tag{4-16}$$

Note that the estimator implies certain properties

- The observations of the flight test data in the regression matrix $\mathbf{X}_b$ is uncorrelated with the residuals, using (4-16) it is derived as follow

$$\begin{aligned} \left(\mathbf{X}_b^T \cdot \mathbf{X}_b\right) \cdot \hat{\boldsymbol{\theta}}_b &= \mathbf{X}_b^T \cdot \mathbf{y} \\ \left(\mathbf{X}_b^T \cdot \mathbf{X}_b\right) \cdot \hat{\boldsymbol{\theta}}_b &= \left(\mathbf{X}_b^T \cdot \mathbf{X}_b\right)\hat{\boldsymbol{\theta}}_b + \mathbf{X}_b^T \cdot \boldsymbol{\epsilon} \\ \mathbf{X}_b^T \cdot \boldsymbol{\epsilon} &= 0 \end{aligned} \tag{4-17}$$

- Similar the output $\mathbf{y}$ is uncorrelated with the residuals

When performing regression in physical space the following properties also hold because of the constant C_{a_0} parameter.

- The column with the "constant" term has a column of ones. Because of the minimization of the sum of the residuals squared, the sum of the residuals $\sum_1^m \epsilon_m = 0$.
- From the above it follows that the sample mean of the residuals equals zero, i.e. $\bar{\epsilon} = \frac{\sum \epsilon}{m} = 0$
- With the sample mean $\bar{\epsilon} = 0$ the sample mean $\bar{y}$ equals the sample mean $\bar{\hat{y}}$.

These properties indicate the assumptions based on the Gauss-Markov theorem, forming the OLS estimator. The assumptions state that

- A linear relationship exists between the regression matrix $\mathbf{X}_b$ and the output $\hat{\mathbf{y}}$, i.e. the model is linear in the parameters.
- The identification condition states that when the columns of the regression matrix are linear independent, there is no multicollinearity. $\mathbf{X}_b$ is full rank
- The expected value for the residuals for any value x_i equals zero. $E\{\epsilon_i|x_i\} = 0$
- The correlation of residuals in the covariance matrix equal zero, i.e. $E\{\epsilon_i\epsilon_j|x_i\} = \text{cov}\{\epsilon_i, \epsilon_j|x_i\} = 0 \; \forall \; i \neq j$ These are the off diagonal terms
- The variance $\forall \; i$ equals: $\text{var}\{\epsilon_i^T|x_i\} = \sigma^2$. This is the assumption of homoscedasticity and states there is no autocorrelation and the variance of the residuals is constant. Note that the variance of the residuals are the terms of the diagonal of the residual covariance matrix, $\text{cov}\{\epsilon|x\} = \sigma^2\mathbf{I}$

4-2-2 Generalised Least Squares

When using the MSBS method for aerodynamic modelling the assumption is made that the aerodynamic behaviour of an aircraft can be described by polynomials. In real live the behaviour of an aircraft is much more complex and the polynomial model is an approximation of this aircraft. The model will have a certain error. When highly non-linear aircraft are modelled or non-linear regions of the flight envelope such as stall a more complex (time variant) polynomial structure is necessary. A problem can arise that because of the increasing complexity the residuals are not homoscedastic any more, indicating there is a correlation between the residuals. For these situations it is more appropriate to use the Generalised Least Squares (GLS) method. The assumptions made for GLS method are similar to the assumptions made for OLS, except the assumption for the covariance matrix. The basics of GLS were published in 1936 by Alexander Aitken (Aitken, 1936). There exists a covariance matrix given by (4-18) and which is positive definite and non-singular.

$$\text{cov}\{\boldsymbol{\epsilon}|x\} = \boldsymbol{\Sigma} \tag{4-18}$$

The cost function of the GLS method is now given by

$$J_{\text{GLS}} = \frac{1}{2}\left(\mathbf{y} - \mathbf{X}_b\hat{\boldsymbol{\theta}}_b\right)^T \cdot \boldsymbol{\Sigma}^{-1} \cdot \left(\mathbf{y} - \mathbf{X}_b\hat{\boldsymbol{\theta}}_b\right) \tag{4-19}$$

In an analogous fashion as (4-16) equation (4-19) is solved for the parameters

$$\hat{\boldsymbol{\theta}}_b = \left(\mathbf{X}_b^T \cdot \boldsymbol{\Sigma}^{-1} \cdot \mathbf{X}_b\right)^{-1} \cdot \mathbf{X}_b^T \cdot \boldsymbol{\Sigma}^{-1} \cdot \mathbf{y} \tag{4-20}$$

The problem arise that the covariance is not known a-priori. A suggested option in literature is to use a two-step hybrid approach. First an OLS regression is used to determine an estimate

of the residual covariance matrix, see (Klein & Morelli, 2006). This will be elaborated in Chapter 4-4. This approach however has to be used with caution, see (Maddala, 1971). A problem is that the computational time for the covariance matrix can be large. Also this matrix has to be inverted in the GLS method. Values in the estimated covariance matrix can be small making the matrix badly conditioned and close to singular after inversion.

4-2-3 Constraint Optimisation

Earlier it was mentioned that columns of the regression matrix cannot be deleted, because it will break the important B-net structure. A model term in barycentric space cannot be deleted. In the next chapter will be explained how to enforce a certain physical model structure to the splines. For this a different method for regression is required. The desired model structure is subjected to the optimisation problem in the form of constraints. Constraints can be hard, soft (weighted/ penalised) and relaxed. Classifications of hard constraints are equality constraints (4-22) and (4-23), inequality constraints (4-24) and (4-25) and bounded constraints (4-26).

$$\underset{c_b}{\textbf{argmin}} \quad J(c_b) \qquad \text{Optimisation Function} \tag{4-21}$$

$$\textbf{subjected to} \quad H(c_b) = c_{b_e} \qquad \text{Equality Constraint} \tag{4-22}$$

$$G(c_p) = 0 \qquad \text{Equality Constraint to delete terms} \tag{4-23}$$

$$G_l(c_p) \leq c_{p_u} \qquad \text{Inequality Constraint upper bound} \tag{4-24}$$

$$G_u(c_p) \geq c_{p_l} \qquad \text{Inequality Constraint lower bound} \tag{4-25}$$

$$c_{p_l} \leq G_b(c_p) \leq c_{p_u} \qquad \text{Bounded Constraint} \tag{4-26}$$

Numerous methods are used and developed to solve for non-linear, quadratic and linear optimisation functions subjected to aforementioned constraints. Well known methods for solving the Karush-Kuhn-Tucker (KKT) conditions are: Trust-region method, Active-set method and Interior Point method.
The KKT conditions are given by (4-27) and published for the first time in (Kuhn & Tucker, 1951). They allow the optimisation function to be subjected to linear equality constraints by using Lagrange multipliers. In fact the KKT conditions are a generalised method of Lagrange multipliers.

$$\begin{bmatrix} \mathbf{X}_b^T \cdot \mathbf{X}_b & \mathbf{H}\mathbf{G}_b^T \\ \mathbf{H}\mathbf{G}_b & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\theta}_b \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_b^T \cdot \mathbf{y} \\ \mathbf{0} \end{bmatrix} \tag{4-27}$$

In equation (4-27) the regression matrix $\mathbf{X}_b$ is similar to (4-12). The equality constraints in barycentric space are given by $\mathbf{H}$ and the physical equality constraints transformed to the barycentric space by $\mathbf{G}$. Combined they form $\mathbf{H}\mathbf{G}_b$. If the KKT matrix is positive definite the coefficients $\boldsymbol{\theta}_b$ can be solved directly using the inverse of the KKT matrix

$$\begin{bmatrix} \boldsymbol{\theta}_b \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_b^T \cdot \mathbf{X}_b & \mathbf{H}\mathbf{G}_b^T \\ \mathbf{H}\mathbf{G}_b & \mathbf{0} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{X}_b^T \cdot \mathbf{y} \\ \mathbf{0} \end{bmatrix} \tag{4-28}$$

However the regression matrix $\mathbf{X}_b$ and the constraints matrix in barycentric space $\mathbf{HG}_b$ are often large and sparse. Preferably inversion of these matrices is avoided. Large and sparse matrices are often ill-conditioned, meaning that they are very sensitive and increase the output error because of numerical errors. These matrices are often close to singular when inverted, see (Gould & Hribar, n.d.). When the constraints in the KKT matrix are rank-deficient, i.e. rank of $\mathbf{HG}_b <$ row $\mathbf{HG}_b$, the KKT matrix is singular. In order to inverse the matrix the Moore-Penrose pseudo inverse can be used. However this also results in a high condition number. The KKT matrix is again ill-conditioned, has a high condition number and the inverse is close to singular. Recommended is to use a numerical approach. For example the pre-conditioned conjugate or an iterative solver, see (Awanou & Lai, 2005) and (Awanou, Lai, & Wenston, 2005). The iterative solver is given by equations (4-29) and (4-30) and is based on the augmented Lagrangian method. It was found that this iterative solver has a fast convergence rate.

$$\hat{\boldsymbol{\theta}}_b^1 = \left(2\mathbf{X}_b^T \cdot \mathbf{X}_b + \frac{1}{\epsilon}\mathbf{HG}_b^T \cdot \mathbf{HG}_b\right)^{-1} \cdot \left(2\mathbf{X}_b^T \cdot \mathbf{y} - \mathbf{HG}_b^T \cdot \hat{\boldsymbol{\lambda}}^{(0)} + \frac{1}{\epsilon} \cdot \mathbf{HG}_b^T\right) \tag{4-29}$$

$$\hat{\boldsymbol{\theta}}_b^{l+1} = \left(2\mathbf{X}_b^T \cdot \mathbf{X}_b + \frac{1}{\epsilon}\mathbf{HG}_b^T \cdot \mathbf{HG}_b\right)^{-1} \cdot \left(2\mathbf{X}_b^T \cdot \hat{\boldsymbol{\theta}}_b^{(l)} + \frac{1}{\epsilon} \cdot \mathbf{HG}_b^T\right) \tag{4-30}$$

Equation (4-29) is used for initialization. For the subsequent iterations l, equation (4-30) is used. A good value for $\epsilon = 10^{-6}$ and $\hat{\lambda}^{(0)} = 1$.

Subjecting the optimisation problem to inequality and bounded constraints requires a different approach. One method that can solve for inequality constraints is the active-set method. Different implementation of this method exist. An efficient way of solving and implementing the active-set method is described by (Kuindersma, Permenter, & Tedrake, 2014) and (Harkegard, 2002). This algorithm verifies first if the inequality constraints are violated. The active-set is an iterative method starting with transferring inequality constraints to equality constraints to become part of the "active-set". The method is initialised by creating an active-set containing the equality constraints only. This optimisation is solved using the (iterative) KKT method, explained in the previous paragraph. The estimated coefficients are used to determine if any of the non-active inequality constraint are violated. If so, they are added to the active-set as equality constraint equal to its upper or lower bound and the next iteration starts. An advantage of the active-set method is that only violated inequality constraints become part of the set active solving the optimisation problem. When only a few of the inequality or bounded constraints are violated the optimisation can be solved fast and is less computational expensive. A disadvantage is that for every step the matrix holding the constraints changes. Also when many constraints are violated the method can become slow. For quadratic programming and linear constraints this method can be solved in polynomial time. The iterative solver is successfully applied and implemented in the active-set method handling both equality and inequality constraints. A summary of the active-set method including a Matlab script example is appended in appendix J.

The interior-point method is an iterative method and is sometimes called the barrier method, see (Robere, 2012). It can be used to solve convex optimisation problems, such as quadratic optimisation. A barrier function in combination with slack variables will prevent violation of the inequality constraints. Lagrangian multipliers λ are used for equality constraints. The

barrier function variable becomes each iteration l smaller and is approaching zero when $l \to \infty$ forming the barrier. It is necessary to rewrite/ augment the constraints to standard form such that they are equal to 0 or ≥ 0. The interior point method can only be solved iteratively. For example Newton's method can be used for this, but in literature it is advised, for a quadratic optimisation, to use the affine scaling algorithm. The optimisation function, in (4-15) has a quadratic form and is linear in the parameters. The affine scaling algorithm is proven to have a polynomial time convergence for convex optimisation (Monteiro, Adler, & Resende, 1990) and has a fast convergence rate. Initial the affine scaling method is developed in the Soviet Union by I.I. Dikin and published in 1967. In the eighties more research and attention was given to this method, see (Vanderbei & Lagarias, 1990). Overall the interior-point method can be used to solve large optimisation problems and has a fast convergence rate. A disadvantage is that all inequality constraints are used when computing the search direction, even when some inequality constraints are not violated. The solution can only be obtained iteratively. More details of the interior-method can be found in appendix K.

In the end both solvers give good results. The reason to choose one above the other mainly depends on computation time. For on-line modelling the solver with lowest computational time will be favourable.

4-3 Polynomial Model Structure Selection

When a candidate set is chosen the initial basis polynomial model structure of the MSBS in Barycentric space is determined by the degree. Using the multinomial theorem, see (4-7), the basis model is expanded into individual terms. Increasing the degree and as a matter of fact the number of terms, the approximation power of the model increases and the model error decreases. However by doing so the possibility of over-fitting increases, i.e. instrumentation noise and measured turbulence will be modelled. Also some simplices can require a more difficult model structure because of local non-linearities. The physical transformation will be used to introduce a polynomial model structure selection to the MSBS aerodynamic modelling process. Remember deleting terms from the regression matrix will break the B-net structure. With the linear transformation from Chapter 2-5-5 the simplex basis polynomials in Barycentric space can be transformed to Cartesian space. At this point the initial physical polynomial for each simplex is available. The simplices together still form the triangulation based on a hyperrectangle. In the remainder of this chapter a method is introduced to determine the importance/ significance for all physical model terms in each simplex and how this information can be used in creating the polynomial model structure and the corresponding constraints.

4-3-1 Model Term Significance

The parameters of the basis polynomials for each simplex will be estimated by the data points of the dataset within that simplex. The complexity, e.g. the non-linearity, of the data points between simplices can differ making the significance of individual terms for each simplex different. With the basis polynomials as starting point a method to determine the significance of individual model terms will be explained in this section. The regression matrix, see Chapter 4-2, is used in order to do this. The columns of this matrix, sometimes called the

explanatory variables are the model terms. However the significance of the polynomial terms in physical space are of interest, but the regression matrix is formed in Barycentric space, see (4-12). Using the same linear transformation defined in (2-28) the regression matrix is transformed to Cartesian space

$$\mathbf{X}_p = \mathbf{X}_b \cdot \mathbf{\Lambda}^{-1} \tag{4-31}$$

Before the significance can be determined an extra transformation is required. The physical regression matrix is transformed to the orthogonal domain. This makes the residuals of the physical model terms independent. Hence every time when a model term is added, the previous calculated model term significance and its corresponding influence on the residuals does not change. When the columns of $\mathbf{X}_p$ are independent there exists an orthogonal basis $\mathbf{W}_p$. This orthogonal matrix has the following properties

- let $\mathbf{W}$ be a $q \times r$ matrix, the columns of $\mathbf{W} \in \mathbb{R}^r$, i.e. $\mathbf{W}$ has full column rank and are linear independent
- therefore the columns of $\mathbf{W}$ are orthogonal, i.e. $\mathbf{w}_k \cdot \mathbf{w}_l = 0 \ \ \forall\ k = (1, 2, \ldots, r)$ and $l = (1, 2, \ldots, r)$ when $k \ \neq l$
- when the columns of $\mathbf{W}$ are normalised then $\mathbf{W}^T \cdot \mathbf{W} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix

The columns of $\mathbf{X}_p$ are orthogonalised using a forward orthogonalisation process called the Gram-Schmidt process. For each column k of the orthogonalized regression matrix $\mathbf{W}$ the significance, in the form of the Total Error Reduction Ratio (ERR), of a term can be determined. Also an orthogonal coefficient vector $\mathbf{g}$ exist such that the output $\mathbf{y}$ is determined while the residuals $\boldsymbol{\epsilon}$ are minimum, see (4-32). The parameter estimation process, e.g. to obtain $\mathbf{g}$ is explained in more detail in Chapter 4-2

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{g} + \boldsymbol{\epsilon} \tag{4-32}$$

The general principle of calculating the ERR is given by (4-33) and (4-34). This process can be repeated for all t_j simplices. the symbols $\circ$ and $\oslash$ are the Hadamard product and Hadamard division respectively. These products are element wise operation on matrices or vectors with the same dimension.

$$\mathbf{g}_{t_j} = \mathbf{y}_{t_j}^T \cdot \mathbf{W}_{t_j} \oslash \operatorname{diag}\left(\mathbf{W}_{t_j}^T \cdot \mathbf{W}_{t_j}\right)^T \tag{4-33}$$

$$\mathbf{err}_{t_j} = \frac{\mathbf{g}_{t_j}^{\circ 2} \circ \operatorname{diag}\left(\mathbf{W}_{t_j}^T \cdot \mathbf{W}_{t_j}\right)^T}{\mathbf{y}_{t_j}^T \cdot \mathbf{y}_{t_j}} \tag{4-34}$$

The ERR is the significance of a term to the model. The k number of columns of $\mathbf{W}$ are analysed individually on their Individual term Error Reduction Ratio (err). Rewriting (4-32) for a column $\mathbf{w}_k$ yields

$$\mathbf{y} = \mathbf{g}_k \cdot \mathbf{w}_k + \boldsymbol{\epsilon}_k \tag{4-35}$$

So what is the ERR exactly? The Mean Squared Error (MSE) of the estimator measures the average of the squared residuals $\boldsymbol{\epsilon}_k$, i.e. the difference between the estimator $\mathbf{g}_k\mathbf{w}_k$ and the expected output value $\mathbf{y}$. For a dataset using M measurements this is given by (4-36), including a formulation using a vector notation.

$$\text{MSE} = \frac{1}{M}\sum_{l=1}^{M}\epsilon_k(l)^2 = \frac{1}{M}\cdot\boldsymbol{\epsilon}_k^T\cdot\boldsymbol{\epsilon}_k \tag{4-36}$$

Next the influence of a term on the reduction of MSE yields important information about the model structure. Especially when the terms are orthogonal to each other. The Reduction of the MSE Reduction of Mean Squared Error (R-MSE) is calculated as follows, (4-37)

$$\text{R-MSE} = \frac{1}{M}\sum_{l=1}^{M}g_k^2 w_k(l)^2 = \frac{1}{M}\cdot g_k^2\cdot\mathbf{w}_k^T\cdot\mathbf{w}_k \tag{4-37}$$

Similar the Total R-MSE Total Reduction of Mean Squared Error (TR-MSE) for M measurements is calculated using (4-38)

$$\text{TR-MSE} = \frac{1}{M}\sum_{l=1}^{M}y(l)^2 = \frac{1}{M}\cdot\mathbf{y}^T\cdot\mathbf{y} \tag{4-38}$$

When the R-MSE and the TR-MSE are known the reduction of the MSE when a term k is included can be expressed in a percentage of the TR-MSE. This is called the ERR and for a term k given by (4-39).

$$\text{err}_k = \frac{\frac{1}{M}\sum_{l=1}^{M}g_k^2 w_k(l)^2}{\frac{1}{M}\sum_{l=1}^{M}y(l)^2} = \frac{g_k^2\cdot\mathbf{w}_k^T\cdot\mathbf{w}_k}{\mathbf{y}^T\cdot\mathbf{y}} \tag{4-39}$$

However it was found that the location of a term within the regression matrix changes its ERR. Terms that appear earlier often have a higher err compared to other terms. So the order of terms has influence on the err. An undesirable order of terms may produce incorrect significance of terms. The model structure may be determined incorrectly when one does not take this into account. In the following paragraph an algorithm is introduced that determines the order of appearance of terms within the estimator based on the err value. The algorithm uses an orthogonal regression procedure based on stepwise regression, introduced by (Billings & Voon, 1986) and (Billings, Korenberg, & Chen, 1988). Using this procedure the order of the terms is determined by their significance.

In the initial stage all terms $k = (1, 2, \ldots, r)$ in the regression matrix $\mathbf{X}_p$ are considered possible candidates for the orthogonal matrix $\mathbf{W}$. For each term the err is calculated.

$$\text{err}_1 = \max_{k=1}^{r} \left(\frac{g_k^2 \cdot \mathbf{w}_k^T \cdot \mathbf{w}_k}{\mathbf{y}^T \cdot \mathbf{y}} \right)$$

where (4-40)

$$\mathbf{w}_k = \mathbf{x}_{p_k} \quad \text{and} \quad g_k = \frac{\mathbf{w}_k^T \cdot \mathbf{y}}{\mathbf{w}_k^T \cdot \mathbf{w}_k}$$

The first term to select is $k_1 = k_{\max}$ with the highest err, i.e. $\text{err}_1 = \max\{\text{err}_k \ , \ 1 \leq k \leq r\}$ and in combination with $\mathbf{w}_{k1}$, g_{k1} the following relation holds

$$\hat{\mathbf{y}}_{k1} = \mathbf{w}_{k1} \cdot g_{k1} + \boldsymbol{\epsilon}_{k1} \tag{4-41}$$

For the second iteration all terms are considered as possible candidates again except k_1 from the first iteration, i.e. $k = (1, 2, \ldots, r) \; \forall \; k \neq k_1$. First the candidate column of $\mathbf{X}_p$ is orthogonalised with respect to $\mathbf{w}_{k1}$ using the Gram-Schmidt process given by

$$\mathbf{w}_k = x_{p_k} - \alpha_{k1\,k} \cdot \mathbf{w}_{k1} \tag{4-42}$$

where the orthogonalisation is given by

$$\alpha_{k1\,2} = \frac{\mathbf{w}_{k1}^T \cdot \mathbf{x}_{p_k}}{\mathbf{w}_{k1}^T \cdot \mathbf{w}_{k1}} \tag{4-43}$$

Similar as before the err is calculated

$$\text{err}_2 = \max_{k=1}^{r} \left(\frac{g_k^2 \cdot \mathbf{w}_k^T \cdot \mathbf{w}_k}{\mathbf{y}^T \cdot \mathbf{y}} \right) \; \forall \; k \neq k_1$$

where (4-44)

$$g_k = \frac{\mathbf{w}_k^T \cdot \mathbf{y}}{\mathbf{w}_k^T \cdot \mathbf{w}_k}$$

Again select term $k_2 = k_{\max}$ with the highest err, i.e. $\text{err}_2 = \max\{\text{err}_k \ , \ 1 \leq k \leq r \ , \ \forall \; k \neq k_1\}$ with $\mathbf{w}_{k2} = \mathbf{x}_{k2} - \alpha_{k1\,k2} \cdot \mathbf{w}_{k1}$ and g_{k2} and err_2
Equation (4-41) is updated by adding the sequential term k_2

$$\hat{\mathbf{y}} = \mathbf{w}_{k1} \cdot g_{k1} + \mathbf{w}_{k2} \cdot g_{k2} + \boldsymbol{\epsilon}_{k1} + \boldsymbol{\epsilon}_{k2} \tag{4-45}$$

Because of the orthogonalisation process not only are the terms orthogonal to each other, but also individual residual terms $\boldsymbol{\epsilon}$. Hence $\boldsymbol{\epsilon}_{k1}$ does not change when $\boldsymbol{\epsilon}_{k2}$ is added. This is similar for the orthogonal parameters. When g_{k2} is added g_{k1} does not change. The process in (4-42), (4-43) and (4-44) continues until the err of last column k_r is determined and

$$\hat{\mathbf{y}} = \mathbf{w}_{k1} \cdot g_{k1} + \mathbf{w}_{k2} \cdot g_{k2} + \cdots + \mathbf{w}_{kr} \cdot g_{kr} + \boldsymbol{\epsilon}_{k1} + \boldsymbol{\epsilon}_{k2} + \cdots + \boldsymbol{\epsilon}_{kr} \tag{4-46}$$

At this point the order of appearance is saved and the final individual term significance is known. This process is repeated for all simplices and is summarised in the ERR vector for each simplex t_j given by (4-47). The sum of the $\mathbf{err}_{t_j}$ is between 0 and 1 and tells something about the quality of the polynomial structure for a simplex t_j. Higher sums are better.

$$\mathbf{err}_{t_j} = \begin{bmatrix} \text{err}_1 & \text{err}_2 & \dots & \text{err}_r \end{bmatrix} \tag{4-47}$$

The Error Reduction Ratio $\mathbf{err}_{t_j}$ vector in (4-47) will be used to create physical model constraints in order to reduce the model size, constraint certain terms per simplex and prevent model over-fitting. The real advantage of this orthogonal procedure combined with stepwise regression compared to stepwise regression only is that the effect that terms have on each other is removed. Without orthogonalization a term can become insignificant when other terms are introduced.

4-3-2 Physical and Barycentric Splines Constraints

Next possibilities are explored how the term significance and the constraints can be used in both the MSBS and the Physical-Splines method. Different types of constraints, repeated in (4-48), are introduced. They will subject the optimisation process for the parameters in such a way that continuity between the simplices is created and the desired physical polynomial structure is imposed.

$\underset{c_b}{\textbf{argmin}}$	$J(c_b)$	Optimisation Function	(4-48)
subjected to	$H_c(c_b) = c_b^{t_j}$	Continuity of simplices	(4-49)
	$G_{dc}(c_p) = 0$	Decoupling of terms	(4-50)
	$G_{err}(c_p) = 0$	Error Reduction Ratio	(4-51)
	$G_{pl}(c_p) \leq c_p^{ub}$	Physical Upper Bound	(4-52)
	$G_{pu}(c_p) \geq c_p^{lb}$	Physical Lower Bound	(4-53)
	$c_p^{lb} \leq G_{pb}(c_p) \leq c_p^{ub}$	Physical Bounded	(4-54)

4-3-3 Smoothness & Continuity in Barycentric space

The B-coefficients have a unique spatial location within each simplex. This spatial location makes it possible to generate continuity constraints between the simplices and make the splines model continuous. The continuity of order r between the simplex polynomials can be important for several reasons. First zero order continuity makes sure no inner "gaps" in the model between simplices exist. When the aerodynamic model is used in a full-motion simulator this prevents excessive and possibly damaging movements. Higher order splines models can be required in applying control allocation methods. Increasing the continuity will let the simplices influence each other and makes the splines model more smooth. The barycentric parameter variances decreases, see (C. C. de Visser, Chu, & Mulder, 2009). The

equality constraints for r^{th} order continuity are created using equation (4-55) and are in Barycentric space.

$$\begin{aligned} c^{t_j}_{\kappa_0,m,\kappa_1} &= \sum_{|\gamma|=m} c^{t_i}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(\nu_*), \quad 0 \leq m \leq r \\ \kappa_0 + m + \kappa_1 &= d \\ (\kappa_0 + m + \kappa_1) + (\gamma_0 + \gamma_1 + \gamma_2) &= d \end{aligned} \tag{4-55}$$

In this equation the order of the continuity is smaller than the degree, i.e. $r < d$. With a multi-index γ independent of κ and ν_* the out-of-edge vertex of t_j.

4-3-4 Decoupling between physical parameters

$$\begin{gathered} G_{dc}(c_{p_\kappa})_{t_j} = \sum_{|\kappa|=d} c_{p_\kappa} = 0 \\ \text{where} \quad \kappa = (\kappa_0, \kappa_1, \kappa_2, \ldots, \kappa_n) \in \mathbb{N}^\ltimes \text{ is the multi-index} \\ \text{with property } \{|\kappa| = (\kappa_0 + \kappa_1 + \kappa_2 + \cdots + \kappa_n) \wedge \kappa \in (\kappa_0, \kappa_1, \kappa_2, \ldots, \kappa_n) \neq \delta\} \end{gathered} \tag{4-56}$$

Higher degree MSBS models have coupled model parameters. A coupled model variable, for example $\alpha \cdot \delta_e$, indicates a relation between these two variables. In other words when α changes δ_e changes and vice versa. The corresponding model parameter $C_{m_{\alpha\delta}}$ are difficult physically to interpret. In general for a $f(x, y)$ non-splines model this means that x or y do not change outside their dimension. The one dimensional views of these variables stay the same. In the case of the MSBS method this is holds for each simplex as well. However between simplices this will differ.

4-3-5 Error Reduction Ratio Constraint on physical parameters

Recalling the Error Reduction Vector $\mathbf{err}_{t_j}$ in (4-47). In this vector the significance of each term is listed taking the order of terms into account. This information will be used to create physical parameter constraints based on the ERR. These physical constraints $G_{err}(c_p)$, see (4-51), are created on the physical-coefficients (c_p) and are constraint to zero. Multiple stopping rules exist to determine the polynomial model structure and prevent over-fitting. Some of these rules are elaborated in (Klein & Morelli, 2006) and (Lombaerts, Chu, Mulder, & Joosten, 2007). In this section three methods are introduced and how to use them in combination with Physical-Splines in creating the constraints using $\mathbf{err}_{t_j}$ and a stopping rule

1. The first method is to set a threshold value on the $\mathbf{err}_{t_j}$. Terms with an $\text{err}_{t_j} \leq \text{err}_{tr}$ are assumed to be insignificant to the model. According to (Billings & Voon, 1986) a typical value for a threshold is between 0.001 and 0.05. With these threshold sufficient terms are included to reduce the prediction errors to a sequence of white noise. Terms are constrained with the following rule

$$G_{err}(c_{p_k})_{t_j} = 0 \ \forall \ \{k \mid k \in (1, 2, \ldots, r) \ , \ (\text{err}_{k_{t_j}} \leq \text{err}_{tr})\} \tag{4-57}$$

2. The second method is to introduce an over-fit penalty as is introduced by (Klein & Morelli, 2006). Recalling the MSE in (4-36). Every time a term is added the MSE decreases. This can result in over-fitting the model and adding too many insignificant terms. A stopping rule that can be used is the Predicted Square Error (PSE) given by

$$\begin{aligned} \text{PSE} &= \frac{1}{M} \cdot \boldsymbol{\epsilon}^T \cdot \boldsymbol{\epsilon} + \sigma^2_{max} \cdot \frac{p_k}{M} \\ &= \text{MSE} + \sigma^2_{max} \cdot \frac{p_k}{M} \end{aligned} \tag{4-58}$$

Terms are added to the estimator in the order determined by the err. Every time a term enters the estimator p increases with one. In equation (4-58) M is the number of measurements and σ^2_{max} is the a-priori upper-bound estimate of the squared error determined by the output variance

$$\sigma^2_{max} = \frac{1}{M} \sum_{i=1}^{M} (y(i) - \bar{y})^2 = \frac{1}{M} \cdot (\mathbf{y} - \bar{y})^T \cdot (\mathbf{y} - \bar{y}) \tag{4-59}$$

where the mean $\bar{y}$ of $\mathbf{y}$ is given by

$$\bar{y} = \sum_{i=1}^{M} = y(i) \tag{4-60}$$

The over-fit penalty gives the PSE a minimum when k number of terms are added to the model, i.e. the over-fit penalty increases faster than the MSE decreases. When this point is reached all the remaining terms, not included in the model, are constraint to zero with the rule

$$G_{err}(c_{p_k}) = 0 \; \forall \; \{ \; k \mid k \in (1, 2, \ldots, r) \, , \, k \notin \; p_k \} \tag{4-61}$$

3. This method constraints all terms when the orthogonalisation process reaches a fit-error based on a certain threshold set on either the MSE, determined with (4-36) or when the sum of the $\mathbf{err}_{t_j}$ reaches its threshold. All the terms k not included at this point are removed from the physical-splines polynomial.

$$G_{err}(C_{p_k}) = 0 \; \forall \; \{ \; k \mid k \in (1, 2, \ldots, r) \; , \; k \notin \; \text{err}_{tr} \geq 1 - \sum_{k} err_k \} \tag{4-62}$$

The system identification process in aerodynamic modelling is iteratively and different models with model settings are created. Part of the model settings are the previous introduced rules and different threshold. The decision of which stopping rule to choose depends on the validation of the different models in terms of model output but also computational time and model size. Other stopping rules not mentioned include using the Coefficient of Determination R^2 and the Predicted Sum of Squares (PRESS). In aerodynamic modelling researched is performed on different stopping rules and according to (Lombaerts et al., 2007) the predicted squared error gives the best results for off-line modelling.

4-3-6 Physical Bounded Constraints

Working in the Cartesian coordinates system (Physical space) gives the possibility to physically interpret the parameters of the model. As explained in Chapter 2-5-5 a linear transformation is used to transform the model parameters of the simplices from Barycentric coordinate system to the Cartesian coordinate system. Besides the easier understanding of the aircraft dynamics, potential identification errors can also be noticed. An estimated aerodynamic model operates within certain physical bounds. When the realistic physical range of these physical bounds are known in advance they can be imposed to the parameter estimation process in form of inequality constraints. The inequality constraints are based on the the control derivatives, see (e. a. Mulder J. A., 2000). Control derivatives give an indication on how forces and moments change due to certain variables such as angle of attack α or side-slip angle β. An example of a control derivative, or sometimes called a stability derivative, is C_{m_α}. This non-dimensional parameter indicates the change of the dimensionless pitch moment C_m, the moment around the y-axis, with respect to the angle of attack α. For a statical stable aircraft this well known stability derivative has to be smaller than zero. When the aircraft to be identified and modelled is known to be stable an inequality constraint can be created. This inequality constraint in the latter example is the partial derivative of the pitching moment C_m with respect to α, i.e. $\partial C_m/\partial\alpha < 0$. Latter inequality constraint will serve as upper bound.

Stability Derivative Functions

The well known stability derivatives, or control derivatives, are based on PDE. In aeronautical engineering these derivatives are mostly studied as constants, because a linearised first order Taylor expansion of the equation of motions is used as model structure. In order to model non-linear dynamical behaviour, while physically interpreting the derivatives, an earlier mentioned method is often used. For this method multiple aerodynamic models are estimated and all stability derivatives are stored in large lookup tables. The models are blended together afterwards. The other method, while keeping physical interpretation, is the physical-splines. Using physical-splines one of the considerations is to increase the degree of the model structure. By doing so the stability derivatives do not have their familiar "engineering" interpretation any more but are now formed by a function depending on one variable or multiple when physical parameters are not decoupled. Beside that a lookup table with stability derivatives in the form of constants also can be considered as a function, four methods are mentioned briefly to keep the simplified form, i.e. "engineering" interpretation.

1. After parameter estimation, the splines model can be linearised for each simplex, just for the sake to have an estimate of a stability derivative in the form of a constant.

2. Limit the spline model degree to a maximum of one and increase the number of simplices instead to model non-linearities. All the non-linearities of the model are still captured. The disadvantage of this method is that the order of continuity has a maximum of zero.

3. Create numerous models based on the purpose and application. Create a high degree splines model used for control allocation or simulation and a one degree splines-model for physical interpretation.

4. Applying the physical parameter decoupling constraints, described in Chapter 4-3-4, makes the control derivatives of a higher degree splines model easier to interpreted without loosing the approximation power of higher degree polynomial model structures.

Since a stability derivative is a derivative of the model function the inequality constraints can be based on root finding algorithms. A problem with these algorithms is that they are non-linear, making the linear solver used for parameter estimation useless. As an example to show this "breakage" a very simple univariate derivative function, (4-63), is used where x is a variable and a, b and c are unknown parameters.

$$f'(x) = ax^2 + bx + c \tag{4-63}$$

The well known root finding algorithm is the one used for univariate quadratic equation, the quadratic formula.

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{4-64}$$

where a, b and c are parameters of the equation. Assume possible solutions in real coordinate space $\mathbb{R}$ only, i.e. there is one real solution when $b^2 - 4ac = 0$, two real solutions when the discriminant $b^2 - 4ac > 0$ and no solution when $b^2 - 4ac < 0$. When the control derivative is positive the corresponding inequality constraint is (take note of the non-linearity)

$$b^2 - 4ac < 0 \tag{4-65}$$

The latter inequality constraint is an example. The constraints will be different for each control derivative depending on the physical bounds and the interval formed by the simplices. However it reveals a difficulty. Even a simple straightforward univariate root solving algorithm creates non-linear inequality constraints, requiring a non-linear solver for the multivariate simplex splines. Also the constraint in (4-65) is not bounded by the domain of simplex t_j , i.e. the physical bounded constraint can be violated when

$$b^2 - 4ac \geq 0 \; \forall \; x \notin t_j \tag{4-66}$$

Models with a degree higher than four will have a derivative to the power three. Solving for root locations often requires a numerical approach. Shortcomings with the previous example are summarised in the list below:

- Non-linearity in the constraints
- Root finding algorithms are difficult for higher degree functions
- Problem when extending from univariate to multivariate
- Difficult to limit constraints to the domain of a simplex only

Linearising Stability Constraints using a Grid

This section introduces a method how to overcome the aforementioned problems and implement and apply non-linear equality, bounded and inequality constraints to the parameter optimisation process. The idea is straightforward. The derivative of the polynomial model structure is determined and for this derivative model "virtual" data-points will be created. This is done for a domain, determined by a simplex t_j, and an equalized spaced grid with points in this simplex. The density of grid points can be changed. The location of each point in this grid is given by a n-dimensional coordinate given in Cartesian space. Each grid point is substituted in the model derivative and represent an (in)equality constraint. Equation (4-67) shows a generalisation used for creating a equally spaced grid $\mathcal{X}_{\text{grid}}$ in Cartesian coordinates.

$$\begin{aligned} \boldsymbol{\chi}_{\text{grid}} &= \bigcup_{|\beta|=\tau} \left(\frac{1}{\tau}\beta_0 \cdot \boldsymbol{\nu}_{0_{t_j}} + \sum_{i=1}^{n} \frac{1}{\tau}\beta_i \cdot \boldsymbol{\nu}_{i_{t_j}} \right) \\ &\text{where } \chi = (\chi_1, \chi_2, \ldots, \chi_n) \in \mathbb{R}^n \\ &\text{where } \nu = (\nu_0, \nu_1, \ldots, \nu_n) \in \mathbb{R}^n \\ &\text{where } \beta = (\beta_0, \beta_1, \ldots, \beta_n) \in \mathbb{R}^{n+1} \text{ is the multi-index} \\ &\text{with property } |\beta| = (\beta_0 + \beta_1 + \cdots + \beta_n) \end{aligned} \tag{4-67}$$

In this equation the grid density is determined by $\tau \geq 1$. The location of the $n+1$ vertices of a n-simplex by the matrix $\boldsymbol{\nu} = (\nu_0, \nu_1, \ldots, \nu_n)$ with ν_i an individual coordinate vector of a vertex. An example of the grid creation is plotted in figure 4-4. All these points are substituted into the model derivative forming M_g inequality constraints based on the grid density. The constraint optimisation can be solved using for example the active-set method described in Chapter 4-2. In this method the inequality constraints are not active when they are not violated decreasing the computation time. The general idea behind the grid-method is when $M_g \to \infty$ all points within the domain do not violate the inequality constraints any more. The assumption is that a dense enough grid will be sufficient enough to approach a non-violating constraint optimisation while keep the constraints linear. Hence the regression stays linear. However M_g approaching ∞ is impossible and merely a mathematical notation. The question arises: what is an adequate density of a grid? The minimum distance between roots can give answer to this question, but rely on the not yet determined parameters. Another suitable solution is assume a grid density τ and confirm afterwards if the constraint for the domain are not violated. Either by solving the root locations of the with inequality constraints identified model and compare with the simplex domain or determine the number of roots in the simplex domain. Another option is to determine the minimum distance between the roots and compare with the density, see (Rump, 1979). This will verify if or if no roots exists on the simplex domain. Using the "grid approach" does not need a non-linear solver, hence the function approximation is still compatible with the linear regression method.

Sturm's sequence is a method to count the number of roots on a pre-determined interval. The exact location of these roots is unknown. For the application of physical bounded inequality constraints this is sufficient. When there is no root on the interval there is no sign change. With this sequence it is possible to verify after regression if the control derivative constraints

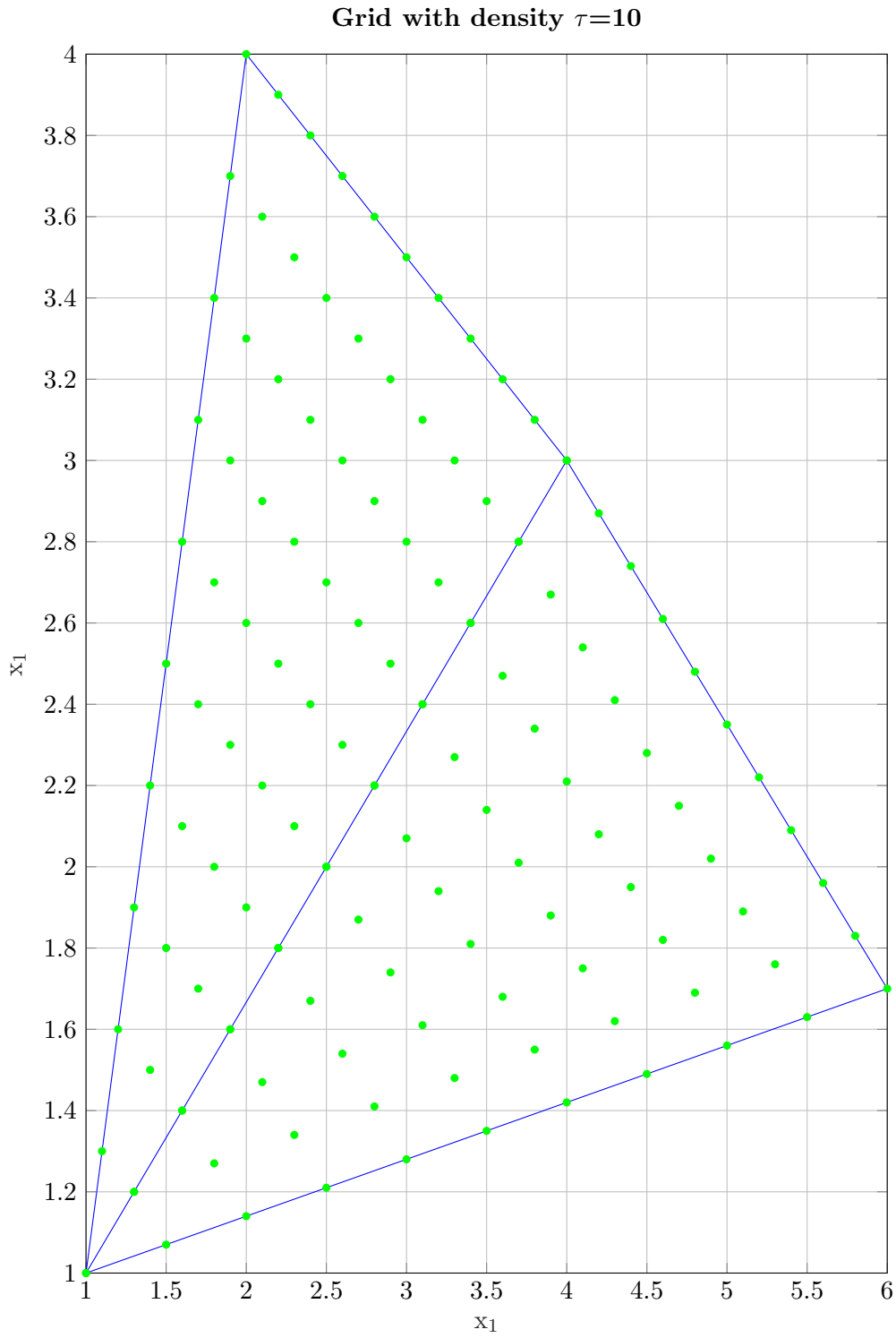


Figure 4-4: An $n = 2$ example with two simplices demonstrating the creation of a uniform grid in physical space with density $\tau = 10$

are not violated. Sturm's sequence is given by

$$\begin{aligned} f_0(x) &= f(x) \\ f_1(x) &= f'(x) \\ f_2(x) = -rem(f_0, f_1) &= f_1(x)q_0(x) - p_0(x) \\ f_3(x) = -rem(f_1, f_2) &= f_2(x)q_1(x) - p_1(x) \\ &\vdots \\ 0 = -rem(f_{m-1}, f_m) & \end{aligned} \tag{4-68}$$

In this sequence f is the polynomial function, i.e. the control derivative, we want to find the roots of in some interval. Start with the original function f, the second step is to determine the derivative of this function f'. The subsequent steps are obtained by dividing the previous two functions using polynomial long division and using the remainder rem of this division. Multiply this remainder with -1. When f has no repeated roots, i.e. square free, the last step is reached, the polynomial f_m which is a non-zero constant. This constant f_m is the greatest common divisor of f. The obtained chain is called the Sturm's chain. With the Sturm's chain it is possible to count the number of roots on the interval. For all elements in the chain substitute the lower bound of the range and take note of the signs. Repeat this for the upper bound. Next look at the number of sign changes for both the upper and the lower bound. Subtracting the number of sign changes yield the number of roots on the interval. A multivariate Sturm's sequence is described in (Pedersen, 1991) and (González-Vega & Trujillo, 1997).

If after regression it was confirmed that the density τ was insufficient, τ will be increased and the parameter estimation process for constraint optimisation is repeated.

Conclusion

Control derivatives are used in a great extent. For example to judge an aircraft's flight dynamics, stability, handling, control etc. This physical interpretation is very useful and with the physical space to Barycentric space transformation this information can be used when models are created with the MSBS method. Because flight test data is used, information about the aircraft to be modelled is known in advance. Hence physical bounds can be applied to the identification process. Models with a degree larger than one will have control derivatives in the form of an univariate or multivariate function. This in combination with their operating simplex domains can make applying the physical bounds a complex process. Using the physical bounds is a part of Semidefinite Programming (SDP), which is a subfield of convex optimization. The inequality constraints form the SDP. Problems encountered by applying the physical bounds in the form of inequality constraints are the non-linear behaviour they produce. To find a solution requirements were set: multivariate support, only applicable on a pre-defined domain, linear in constraints. A solution was found in the form of a grid. The points in the grid are determined by the domain which in turn is determined by the simplex. The grid points can be multivariate. All these points are used as linear inequality constraint belonging to a control derivative. At this point the final model can still have unpredictable behaviour. Since the grid size M_g cannot go to infinity it is unknown what the model does within the unconstrained area in between the points, i.e. a possibility exists

the physical bounds are violated. This is why the model afterwards is verified with a root counting algorithm on a domain: the Sturm's sequence, or other root finding algorithms.

4-3-7 Differential Constraints

The differential constraints can be used for extrapolation and improve the quality of the model on the boundaries. The boundaries of higher degree polynomial models, with poor data coverage, will often show divergent behaviour. B-coefficients differentials, on the boundary of the triangulation, can be constraint in a certain direction reducing this divergent behaviour. The differential constraints, (C. C. de Visser et al., 2011), are expressed in the form

$$\mathbf{D}_b \cdot \mathbf{c}_b = \mathbf{d} \tag{4-69}$$

Where $\mathbf{D}_b$ is a block diagonal matrix with N differential constraints. The vector $\mathbf{c}_b$ is the B-coefficients vector and $d \in \mathbb{R}^{N\times 1}$ is the vector with the values to constraint the differentials to. The matrix $\mathbf{D}_b$ can be formed for all simplices t_j using

$$\mathbf{D}_b = \text{diag}\left(\mathbf{D}_m^{t_j}(\mathbf{a},\mathbf{b}), 1 \leq j \leq J\right) \in \mathbb{R}^{M\times J\cdot\hat{d}} \tag{4-70}$$

where $\mathbf{D}_m^{t_j}(\mathbf{a},\mathbf{b})$ is defined as

$$\mathbf{D}_m^{t_j}(\mathbf{a},\mathbf{b}) := \frac{d!}{(d-m)!}\mathbf{B}^{d-m}(\mathbf{b})\mathbf{P}^{d,d-m}(\mathbf{a}) \tag{4-71}$$

where $\mathbf{P}^{d,d-m}(\mathbf{a})$ is the de Casteljau matrix, d the degree, m the order of the differential constraints, $\mathbf{a}$ the directional coordinate of the constraint direction $\mathbf{u}$ at location $\mathbf{b}$. The directional coordinates and location are given in barycentric space.

The differential constraints can also be created in the physical space using the the physical space to barycentric space coordinate transformation to transform the constraints to the barycentric space. An example of the physical differential constraints are the physical bounded constraints of previous section. Effectively it means that the constraint are created using a directional derivative. The directional derivative for a physical function $p(\mathbf{x})$ in the constraint direction $\mathbf{u}$ is defined by

$$\nabla_{\mathbf{u}}\ p(x_1, x_2, \ldots, x_n) \tag{4-72}$$

For a certain location $\mathbf{v}$ equation (4-72) becomes

$$\nabla_{\mathbf{u}}\ p(\mathbf{v}) = \mathbf{D}_p \cdot \mathbf{c}_p = \mathbf{d} \tag{4-73}$$

The direction coordinate and location are now given in physical space using the Cartesian coordinate system. Equation (4-69) becomes

$$\mathbf{D}_p \cdot \mathbf{c}_p = \mathbf{D}_p \cdot \mathbf{\Lambda} \cdot \mathbf{c}_b = \mathbf{D}_b \cdot \mathbf{c}_b = \mathbf{d} \tag{4-74}$$

The location of the differential constraints $\mathbf{v}$ can be the location of vertices, B-coefficients or arbitrary points. A uniform per-simplex grid $\mathbf{v}_{\text{grid}}$ can be created using the same method

from Chapter 4-3-6, repeated below for convenience

$$\begin{aligned} \mathbf{v}_{\text{grid}} &= \bigcup_{|\beta|=\tau} \left(\frac{1}{\tau}\beta_0 \cdot \boldsymbol{\nu}_{0_{t_j}} + \sum_{i=1}^{n} \frac{1}{\tau}\beta_i \cdot \boldsymbol{\nu}_{i_{t_j}} \right) \\ &\text{where } \nu = (\nu_0, \nu_1, \ldots, \nu_n) \in \mathbb{R}^n \\ &\text{where } \beta = (\beta_0, \beta_1, \ldots, \beta_n) \in \mathbb{R}^{n+1} \text{ is the multi-index} \\ &\text{with property } |\beta| = (\beta_0 + \beta_1 + \cdots + \beta_n) \end{aligned} \tag{4-75}$$

Using the physical formulation and a grid the differential constraints can also be introduced as inequality constraints. These constraints can be used on the stability derivatives of an aircraft, such as $C_{m_\alpha} < 0$. The inequality constraints will have the form of

$$\nabla_{\mathbf{u}}\, p(\mathbf{v}_{\text{grid}}) = \mathbf{D}_p \cdot \mathbf{c}_p \leq \mathbf{d} \tag{4-76}$$

4-4 Splines Model Validation and Quality Assessment

Validation of the Aerodynamic Splines Model is important in order to give an assessment of the model created. Different methods are available. In this chapter convenient methods often used for the MSBS, presented in (De Visser, 2011), are elaborated. The role of the physical-splines in these methods is explained. After estimating an aerodynamic model and validation, the quality of the model is assessed using several aspects based on the outcome of the validation methods.

Recalling that the dataset with measurements was divided in an identification set and a validation set. A convex hull, E_X is formed by this identification set. The convex hull is determined by the extreme points of the identification set in such a way that the smallest convex set is formed which contains all $\mathcal{X}_{ident}$. In this convex hull linear interpolation is possible.

$$E_X = \langle \mathcal{X}_{ident} \rangle \tag{4-77}$$

Only within this convex hull it is possible to perform a proper validation. In order to this the validation dataset will be used. The validation dataset is the share of the dataset not used for the identification and that is within the space of the splines model, i.e. validation points outside this convex hull of the splines model space are not used, see (4-78)

$$\mathcal{X}_{val} = \bigcup_{i=1}^{M_{val}} \mathcal{X}_i \in \mathbb{R}^n \tag{4-78}$$

However since the triangulation is created around the convex hull of the identification dataset, the possibility exists that the convex hull of the triangulation is larger compared to the identification set. Hence some B-coefficients are not within this convex hull E_X. Important to remember is that during validation of the aerodynamic model these B-coefficients and this part of the domain can be expected to have a worse quality assessment. Also when someone extrapolates the model quality cannot be guaranteed.

The model validation is divided in several parts including a raw validation for comparing model output and measured output using the validation set. Furthermore a global model inspection of the physical parameters, analysing the model residuals, stability analysis and statistical analysis. Finally all the individual outcomes are used to give a quality assessment of the aerodynamic model created.

4-4-1 Residual Analysis

For the model validation the validation dataset is used. This dataset is not used to identify the aerodynamic model. The states are given as input to the aerodynamic model. The output of the model is compared to the measured output. Graphs can be created that plot both outputs. Extra attention is given to sections with the experimental flight manoeuvres, because of the higher complexity. Subsequently the model residuals, given by (4-79) are analysed.

$$\boldsymbol{\epsilon}(\mathcal{X}_{val}) = \mathbf{Y}_v - \hat{\mathbf{Y}}_v \tag{4-79}$$

with $\mathbf{Y}_v$ the validation output and $\hat{\mathbf{Y}}_v$ the model output.

First the Coefficient of Determination is calculated. The coefficient of determination has a value between 0 and 1 and is calculated using the MSE.

$$\text{MSE}(\epsilon) = \frac{1}{M}\sum_{i=1}^{M}\epsilon(i)^2 = \frac{1}{M}\boldsymbol{\epsilon}^T \cdot \boldsymbol{\epsilon} \tag{4-80}$$

and the Total Squared Error (TSE)

$$\text{TSE}(\epsilon) = \frac{1}{M}\sum_{i=1}^{M}(Y_v(i) - \bar{Y}_v)^2 = \frac{1}{M}(\hat{\mathbf{Y}}_v + \bar{Y}_v)^T \cdot (\hat{\mathbf{Y}}_v + \bar{Y}_v) \tag{4-81}$$

The relation of the coefficient of determination is now given by (4-82)

$$R^2 = 1 - \frac{\text{MSE}}{\text{TSE}} \tag{4-82}$$

The relation can give an indication on how well the model is fitted in terms of predictability, where $R^2 = 1$ indicates a perfect fit. On the contrary it is not a guarantee that the model quality is good. Hence noise can be fitted or the candidate set used is incorrect.

Next the quality of the fit is determined by calculating the Root Mean Squared Error (RMS) using the MSE

$$\text{RMS}(\epsilon) = \sqrt{\text{MSE}(\epsilon)} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\epsilon(i)^2} = \sqrt{\frac{1}{M}\boldsymbol{\epsilon}^T \cdot \boldsymbol{\epsilon}} \tag{4-83}$$

the RMS however depends on the magnitude of the input variables. In order to have a better interpretation of the RMS they are converted to a relative scale by using

$$\text{RMS}_{\text{rel}}(\epsilon) = \frac{\text{RMS}(\epsilon)}{max(\mathbf{Y}_v) - min(\mathbf{Y}_v)} \tag{4-84}$$

The RMS_{rel} gives an indication of the quality of the regression by using an estimation of the variance. Hence can the model determine the correct output. A low RMS_{rel} means the the output is close to the expected output of the validation dataset.

As mentioned in Chapter 4-2 the assumption is made that the residuals are uncorrelated with each other. This is tested by estimating on a large scale the validation residual autocorrelation as follows.

$$\hat{R}_{t_j}(k) = \frac{1}{M_{t_j}} \sum_{i=1}^{M_{t_j}-k} \epsilon_{t_j}(i)\epsilon_{t_j}(i+k) \ , \quad k = 0, 1, \ldots, M_{t_j} \tag{4-85}$$

For this estimation the autocorrelation for each simplex t_j is determined. Assumed is that simplices do not have influence on each other. This is not always true since the use of smoothness constraints will create continuity between simplices. The residual of each individual simplex is said to be uncorrelated when for 95% of all $k > 0$ the following relation holds

$$\frac{|\hat{R_{t_j}}(k \neq 0)|}{\hat{R}_{t_j}(0)} \leq \frac{\mathcal{N}_{0.95}}{\sqrt{N_{t_j}}} \tag{4-86}$$

When correlation exists between the residuals the Gauss-Markov assumption is violated and the OLS method cannot be used. As mention in Chapter 4-2-1 and Chapter 4-2-2 the GLS has to be used. For the GLS a covariance matrix $\mathbf{\Sigma}$ is required. This covariance matrix is not available a-priori. Therefore a two step approach is introduced. First the parameters are estimated using the OLS method and when the residuals are correlated using (4-86) the covariance matrix is estimated. This estimation of the covariance matrix will be used in the GLS method. Determining this matrix is not a trivial process, using (Klein & Morelli, 2006), an estimation of the residual covariance is given by

$$\mathbf{\Sigma}_{t_j} = \frac{1}{M_{t_j}} \sum_{i=1}^{M_{t_j}-k} \epsilon(i)\epsilon_{t_j}(i+k) \ , \quad k = 1, 2, \ldots, M_{t_j} \tag{4-87}$$

In this residual matrix the assumption is made again that the residuals only have a covariance within the an individual simplex. No covariance exists outside a simplex so

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{t_1} & 0 & \ldots & 0 \\ 0 & \mathbf{\Sigma}_{t_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \mathbf{\Sigma}_{t_j} \end{bmatrix} \tag{4-88}$$

Determining the inverse is a tedious process. The small values of ϵ introduces numerical problems and badly conditioned matrices. The advantage of the assumption made is that the residual covariance matrix becomes a sparse matrix simplifying the calculation of the inverse.

Now the inverse is determined of multiple $(M_{t_j} \times M_{t_j})$ matrices instead of matrix with the size of all data points $(M \times M)$.

$$\mathbf{\Sigma}^{-1} = \begin{bmatrix} \mathbf{\Sigma}_{t_1}^{-1} & 0 & \dots & 0 \\ 0 & \mathbf{\Sigma}_{t_2}^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \mathbf{\Sigma}_{t_j}^{-1} \end{bmatrix} \tag{4-89}$$

Next the residuals of model are analysed for each simplex t_j individual. Using the residuals the empirical confidence bounds are determined using the Chebyshev's inequality.

$$Pr\left(|\epsilon_{t_j} - \mu(\epsilon_{t_j})| \geq a\sigma_{t_j}\right) \leq \frac{1}{a^2} \tag{4-90}$$

Because the simplices are analysed individually a moving average of the of standard deviation σ of the per-simplex residuals ϵ_{t_j} is used. Hence, the mean value of the residuals $\mu(\epsilon_{t_j})$ equals zero because of the Gauss-Markov assumption. The confidence bound changes for each interval and is determined by $a\sigma$-units. In (4-90) the probability of the norm is given. The bound has a symmetric deviation around the mean μ. An often used bound equals 6σ-units. This corresponds to a bound with 97.22% confidence within the standard deviation of the mean.

4-4-2 Stability Analysis

The bounds used for the stability analysis dependent only on the B-coefficients and the degree of the spline model. This proof is presented in (Lai & Schumaker, 2007). This makes the stability of the splines-model relatively easy to determine. The stability analysis is used to validate the behaviour of the splines-model given by the bound (4-91)

$$\frac{||\mathbf{c}_b||}{K} \leq ||p|| \leq ||\mathbf{c}_b|| \tag{4-91}$$

In this equation the lower bound is given by dividing the b-coefficients by a constant K. The lower bound is not analysed and will be skipped. $||p||$ indicates the polynomial norm. The upper bound $||\mathbf{c}_b||$ is by definition the maximum B-coefficient in the vector of B-coefficients is given by

$$||\mathbf{c}_b|| \equiv \max_{|\kappa|=d} |c_\kappa| \tag{4-92}$$

Since the upper bound is analysed only, rewriting (4-91) yields (4-93)

$$||p|| \leq ||\mathbf{c}_b|| \tag{4-93}$$

As mentioned before not all B-coefficients are in the validity region of convex hull E_X. The b-coefficients do not have a global influence on the spline model because of the local basis

functions. Therefore the stability analysis is given by the measure of the ratio, given in (4-94), using only the B-coefficient in E_X.

$$R_{E_X} = \frac{\text{range}\{\mathbf{c}_b \in E_X\}}{\text{range } E_X} \tag{4-94}$$

4-4-3 Statistical Analysis

Parameter collinearity occurs when two or more variables within the candidate set have a high correlation. This means that one variable could predict the other variable. For regression this can mean that the model is very sensitive for minor changes. This decreases the reliability of the final model. Multicollinearity can be determined with the Variance Inflation Factor (VIF) (Stine, 1995). The VIF is calculated using three steps and is repeated for each simplex. The linear transformation, (2-29), is used to transform the barycentric regression matrix, (4-12), to physical space. Rewriting this matrix to an equation form gives for example a per-simplex linear in parameters four dimensional polynomial.

$$p(x_1, x_2, x_3, x_4)_{t_j} = c_{p_0} + c_{p_1}x_1 + c_{p_2}x_2 + c_{p_3}x_3 + c_{p_4}x_4 + \epsilon \tag{4-95}$$

Next if the VIF is determined of x_1, (4-95) is rewritten where x_1 becomes a function of the other explanatory variables

$$x_1 = c_{p_0} + c_{p_2}x_2 + c_{p_3}x_3 + c_{p_4}x_4 + \epsilon_{x_1} \tag{4-96}$$

Using OLS, the previous regression is solved. The VIF can now be determined with

$$\text{VIF}_1 = \frac{1}{1 - R_1^2} \tag{4-97}$$

Where R_1 is coefficient of determination and can be calculated using (4-82). Values for VIF smaller than 4 are assumed no multicollinearity exist (Obrien, 2007). This is repeated for the other variables for each simplex. Please note when higher degree, non-linear, model terms are added for example c_2x^2 there will be a high multicollinearity with the term c_1x.

Next the model parameters covariance matrix is determined. When enough data-points are available the parameter covariance is calculated using the pseudo inverse from (4-28)

$$\text{cov}(\hat{\mathbf{c}}_b) = \sigma^2 \left(\mathbf{X}^T \cdot \mathbf{X}\right)^{-1} \tag{4-98}$$

with σ^2 the global variance of the residuals. An estimation of the per-simplex residual variance is given earlier in this section by (4-87). A more conservative approach is when the global residuals variance is used instead of the per-simplex residual variance. Rewriting (4-87) yields

$$\begin{aligned}\sigma^2 &= \frac{1}{M}\sum_{i=1}^{M}(\epsilon(i)-\bar{\boldsymbol{\epsilon}})^2 \\ &= \frac{1}{M}\sum_{i=1}^{M}(\epsilon(i))^2\end{aligned} \tag{4-99}$$

when the GLS method is used the parameter covariance is given by

$$\text{cov}(\hat{\mathbf{c}}_b) = \left(\mathbf{X}^T \cdot \mathbf{X}\right)^{-1} \tag{4-100}$$

The diagonal of the covariance matrix in (4-98) and (4-100) are the parameter variances, so

$$\text{var}(\hat{\mathbf{c}}_b) = \text{diag}(\text{cov}_b(\hat{\mathbf{c}}_b)) \tag{4-101}$$

Multiple methods are available to determine a confidence bound. A conservative method, the Simultaneous Bonferroni method, is presented in (Kyriakides & Heydt, 2006) and given by

$$|\mathbf{c}_b - \hat{\mathbf{c}}_b| \leq t_{\frac{\alpha}{2p},N-p}\sqrt{\text{var}(\hat{\mathbf{c}})_b} \tag{4-102}$$

with the Student's t-distribution given by $t_{\frac{\alpha}{2p},N-p}$ where $N-p$ are the degrees of freedom and $\frac{\alpha}{2p}$ the significance level.

Remember that the parameters in Barycentric space, the B-coefficients, have a unique spatial location. This spatial location can give an indication where in the convex hull, and in which of the measured part of the flight envelope, the aerodynamic model has a high variance. While creating the model extra attention can be given to these regions. Data can be added or the triangulation can be adjusted. Expected is that B-coefficients on the boundary will have a higher variance. The convex-hull of the triangulation $\langle\mathcal{T}\rangle$ can be larger than the convex-hull of the identification dataset $\langle\mathcal{X}_{id}\rangle$, resulting in B-coefficients locations outside $\langle\mathcal{X}_{id}\rangle$. The variance of the physical-parameters are calculated using the barycentric to physical space transformation $\boldsymbol{\Lambda}$

$$\begin{aligned}\text{cov}(\hat{\mathbf{c}}_p) &= \sigma^2\left((\mathbf{X}\cdot\boldsymbol{\Lambda}^{-1})^T\cdot\mathbf{X}\cdot\boldsymbol{\Lambda}^{-1}\right)^{-1} \\ &= \sigma^2\left((\boldsymbol{\Lambda}^{-1})^T\cdot\mathbf{X}^T\cdot\mathbf{X}\cdot\boldsymbol{\Lambda}^{-1}\right)^{-1} \\ &= \sigma^2\boldsymbol{\Lambda}\cdot\left(\mathbf{X}^T\cdot\mathbf{X}\right)^{-1}\boldsymbol{\Lambda}^T \\ &= \sigma^2\boldsymbol{\Lambda}\cdot\text{cov}(\hat{\mathbf{c}}_b)\cdot\boldsymbol{\Lambda}^T\end{aligned} \tag{4-103}$$

similar as before the physical parameter variances are given by

$$\text{var}(\hat{\mathbf{c}}_p) = \text{diag}(\text{cov}(\hat{\mathbf{c}}_p)) \tag{4-104}$$

Equation (4-102) can be used to determine the confidence interval of the physical parameters. The stability analysis, in Section 4-4-2, can be repeated for the physical coefficients.

4-4-4 Global Model Inspection

After parameter estimation, described in Chapter 4-2, the Barycentric B-coefficients are transformed to physical coefficients. For clarity this is repeated in (4-105). This way the model output can be analysed and interpreted.

$$\hat{\mathbf{c}}_p = \mathbf{\Lambda} \cdot \hat{\mathbf{c}}_b \tag{4-105}$$

The physical coefficients are used for the global model inspection. Do coefficients or stability derivatives represent the expected dynamic behaviour of the aircraft modelled. The symmetric and asymmetric modes of vibrations, such as the short period and phugoid oscillations and aperiodic roll, spiral motion and the Dutch roll, can be analysed using the stability derivatives.

Chapter 5

Conclusion

From 2019 all air-carrier pilots are obliged to go through flight simulator-based stall recovery training. Therefore a task-force was formed by the C&S department. The implication of this new legislation is that flight simulator aerodynamic models need to be updated. Stall and post stall dynamics of an aircraft are very non-linear requiring high-fidelity flight simulation models. Different system identification techniques, for modelling complex non-linear dynamics, are compared. Many of these techniques have shortcomings such as lacking a physical interpretation of the aerodynamic model. Neural networks and fuzzy logics system identification techniques are black-box models and have global basis functions. The non-sparse solution system in combination with non-linear regression methods make them very computational intensive.

A better technique for aerodynamic modelling is the MSBS. Since 2005 the MSBS have proven to be a true general function approximator. They are used successfully for modelling highly non-linear aircraft dynamics. The MSBS method uses local basis functions to fit any scattered dataset. The local basis functions also increase the computational efficiency and give the property to locally adjust/ update the model. MSBS are compatible with linear regression methods, but require a geometric support structure. Creating the triangulation is not always a trivial process. Another shortcoming is that the B-coefficients do not have a straightforward physical interpretation. This makes the modelling process and model validation more difficult. In the introduction of this preliminary a brief overview of splines was given. It all started with physical splines and it ended with the physical-splines formulation. This new formulation makes it for example possible to linearly transform B-coefficient from the Barycentric Space to the Physical Space and physically interpreted them. The new physical transformation used in combination with the MSBS are called the physical-splines.

The aircraft to be modelled with physical-splines is the Cessna Citation II. Experimental flights are performed using the PH-LAB, the Cessna Citation II co-owned by the TU-Delft, creating a flight-test dataset with measurements. The flight-test data is filtered and the flight path is reconstructed. The task force did research on filtering of flight-test data and the reconstruction of the flight-path. A toolbox is created for this. The equation of motions are used to form the six forces and moments equations and are made dimensionless. Other earlier

work performed by the task-force is the labelling of the overhaul importance of the DASMAT simulation framework blocks. Preferably the blocks are updated to the embedded Matlab function representation. The physical-splines model will be part of the aerodynamic block.

The process in creating an aerodynamic model with MSBS is subdivided in three parts: the geometric model structure selection, polynomial model structure selection and the parameter estimation. This process is iteratively creating multiple aerodynamic models. All these model are compared and validation is used to asses the quality.

During the geometric model structure selection different candidate sets of variables are selected. The candidate sets for the Cessna Citation II were determined in a previously performed research using the occurrence of hysteresis. The number of state and control variables in a candidate set determines the model dimension and also the dimension of the triangulation. High dimensional spaces have the tendency to become very large and empty. This introduces problems when not enough flight-test data is available or the distribution of the flight-test data over the measured domain is insufficient. This is mainly caused by physical limitation of the aircraft or aviation restrictions. The missing flight-test data will create voids in high dimensional spaces resulting in empty simplices, poor performing simplices because of bad data distribution and inner model discontinuities. Therefore it was decided to reduce the size of the candidate set by using the assumption that some model states and control variables are independent. This decoupling creates two groups of variables: longitudinal and lateral. Another option is to remove a variable from the MSBS process and introduce it as a global variable. More research and experiments will be performed on physical-splines, on how they can contribute and help identifying sparse datasets and making the method more robust.
Creating a triangulation is not a trivial process and is difficult to visualise for dimensions $n > 3$. For this research project it is decided to use n-hyperrectangles as a basis and include simplices inside the hyperrectangle so they form together the triangulation. The multiple n-simplices keep alignment to each other such that all $(n-1)$-faces are aligned. This alignment is very important for creating continuity in the splines model. An algorithm used for creating the geometric support structure is called the Kuhn Triangulation and will be used in this research project. The complexity of the triangulation can be changed by adding more hyperrectangles per dimension. As mentioned before the quality of the triangulation depends on the flight-test data used. Methods were introduced in this preliminary thesis to visualise the multidimensional data and optimise the triangulation. A matrix-plot visualises the pair-wise relation between the variables of a candidate-set and the distribution of the flight-test data. Outlying measurements in the flight-test data and regions with low number of measurements can be indicated. K-means clustering in combination with a silhouette validation can be used to determine if clustering occurs in the flight-test dataset. Quality improvement options, due to problems caused by the flight-test data, that can be considered are 1) delete outlying measurements, 2) create multiple spline models based on clusters, 3) delete simplices and use extrapolation techniques such as differential constraints afterwards, 4) use physical constraints and 5) change the orientation of simplices in the hyperrectangle. The latter method makes use of a 2-dimensional transformation, or planer rotation, of the coordinates of vertices and therefore effectively rotating the simplices within the triangulation. Multiple rotations in different dimensions are possible. As a results a different triangulation is obtained. Including the rotation in the aerodynamic modelling process an optimal orientation of the triangulation can be found. Also a consideration is to perform 1-dimensional transformations of vertices

on the hyperrectangle boundaries and therefore effectively changing the convex hull or the triangulation to reduce voids in boundary regions.

The polynomial model structure selection is very important in order to not over-fit the aerodynamic model and to select the most important model terms. The disadvantage of the MSBS method is that the polynomial model structure selection is very limited. First because of the lack of physical meaning of the B-coefficients and second because removing model terms in the B-form will break the b-net structure and therefore the splines model. Basically the polynomial model structure for the MSBS is determined by the splines model degree. Changing the degree is the only manner how to influence the model structure. This can cause problems especially when the non-linearities differ per simplex. Simplices with low non-linearities will tend to over-fit quicker. Before implementing the physical-splines method to the polynomial model structure selection, an initial model structure is created first. These initial polynomials terms forming the structure are created by setting the degree. Next the global B-form regression matrix can be transformed to a physical regression matrix using the physical transformation. With the physical regression matrix the model terms can be analysed and (aerodynamic) modelling techniques can be implemented. The importance of model terms, i.e. their significance, can be obtained by a forward orthogonal stepwise regression method. For each step and each simplex the significance of al remaining, not yet selected, terms are considered based on their significance. The significance is calculated as ERR. The term with the highest ERR is the term selected because the ratio in reducing the error in the model is the highest. The stepwise process continues until the error reduction of all terms is known. The orthogonalisation process makes sure that the previous calculated ratios do not change and the residual terms for each step are independent of each other. The main difference, compared to conventional linear interpolation techniques, using a stepwise model term selection approach is that terms cannot be deleted from the regression matrix. With MSBS all terms form a part of the parameter optimisation process. Constraints have to be created for each simplex and subjected to the optimisation process in order to influence or delete terms. These constraints can be equality, inequality and bounded constraints.
Equality constraints created are the smoothness constraints in Barycentric space for splines model continuity between simplices, decoupling of terms constraints and model selecting/ deleting terms constrains based on the ERR. The last two constraints are both created in physical space. The decoupling of terms can make the physical interpretation of the model easier and reduces the overall model size. The ERRs and optimal order of terms can be used to 1) exclude physical terms with a ERR lower than a certain threshold, 2) after a certain fit-error is reached all remaining physical terms are excluded and 3) the PSE introduces a penalty for every physical term added and will result in an optimal number of terms, other terms are hard constraint.
Inequality or bounded constraints can be created based on the a-priori knowledge of the dynamics of the Cessna Citation II to be modelled. For example, some stability derivatives are known to be negative or bounded in certain regions. Stability derivatives in engineering perspective are known as constants. However degree > 1 models will lose the engineering perspective and have functions as stability derivatives. These partial stability derivatives will create non-linear constraints. Furthermore the outcome of stability derivatives within the domain of a simplex are of importance only. In order to maintain the linear in parameters optimisation process and subject the stability derivative constraints to the simplex domain only, a grid method is introduced. This grid method creates multiple data-points depending

on the grid size within a simplex domain. All these data-points are substituted in the partial derivatives and form individual inequality constraints. Afterwards the grid density assumption can be verified by using root finding methods and check if the inequality constraints are not violated. The constraints, except the continuity constraints, can be different for each simplex.

After the constraints are created the last phase, the parameter estimation, starts. First the physical constraints are transformed to barycentric space using the inverse of the physical transformation. Performing regression in barycentric space is preferable because of the normalised coordinate system. This reduces the condition number and numerical problems. Next a cost function is created using the OLS optimisation. For modelling highly non-linear dynamics, when correlation between the residuals cannot be assumed zero, GLS are advised. Optimisation problems with equality constraints only, can be solved with the KKT conditions. For very large matrices an iterative solver exists. For inequality constraints multiple methods are researched. Two of them are the active-set method and the interior-point method. Both methods have good results, but their performance depends on the number of inequality constraints violated. When computational time is important, for example in on-line modelling, one of the methods can be favourable. Finally the aerodynamic model created is validated and an assessment of the quality is given. To support the assessment the model residuals are analysed, the physical parameters and outcome of the validation dataset are analysed and a stability and statistical analysis is performed.

The aerodynamic modelling process using the MSBS in this preliminary thesis was researched and combined with the physical-splines transformation. The new formulation is compared to the current aerodynamic process and new possibilities were explored. This has resulted in that new and conventional aerodynamic modelling techniques, requiring a physical interpretation, are added to the modelling process. With the physical interpretation a per-simplex polynomial model structure can be created that not only reduces the overall model size, but also prevents over-fitting. In future work the new high-fidelity Cessna Citation II model will be created and integrated to current flight simulation framework. The new aerodynamic model will be compared to the base-line model in terms of continuity, model approximation power, computational efficient and applicability in (adaptive) model based control allocation. With help of the newly developed methodology the stall dynamics will be integrated into the aerodynamic model. This research work will contribute in the form of a new methodology and finally a new high-fidelity full flight envelope aerodynamic model of the Cessna Citation II. This high-fidelity model will help air-carrier pilots train on flight simulators and will increase safety in aviation.

Appendix A

Test Case One - Non-Linear Performance

This test case measures the performance of the MSBS technique using the method described in Chapter 2-5, and the tools developed in this thesis. In search for a high non-linear dataset the author came across some pictures of mountains and thought of Digital Elevation Model (DEM) data. This DEM data is available on the internet for free, (Ferranti & Christoph Hormann, n.d.). The source of this data is NASA Shuttle Radar Topographic Mission (SRTM). Specifically the data used in this test case is part of the SRTM30 dataset, meaning it consists of measuring points in a grid form every 30 metres lateral and longitudinal. The elevation, measured perpendicular from the lateral and longitudinal plane, has an accuracy of 1" which is approximately 2.5 cm. From the DEM file data points where extracted where the Matterhorn is located. The Matterhorn is a high (4.478 m), iconic mountain in Switzerland, Figure A-1. A two dimensional $n = 2$ model will be created with input the lateral and longitudinal GPS coordinates and output the elevation.

In this test case the influence of the degree, triangulation size (nTRI), continuity and setting a limit on the error reduction ration (ERR) are examined. Models with degree 1, 2 and 3 are created together with a continuity of C^{-1}, C^0 and C^1, quadrilateral shaped equal sized triangulation of 1x1 up to 15x15 and 5 limits for the ERR. This forms a total of 810 combinations. Other model information is saved as well, such as the calculation time and the estimated parameters. A scatter plot shows that the data is nicely distributed over the the triangulation. Hence there is no need to optimise or rotate the hyperrectangle based on the dataset.

Figure A-2 shows how the triangulation influences the decrease of the the retaliative root mean square error (RMS). The three plots belong to degree 1, 2 and 3 models with no continuity constraints. The degree 2 and 3 models drop below 5% RMS when the triangulation size is around 2x2. The linear model requires a larger triangulation size before the same RMS error is reached. Increasing the triangulation size clearly shows a saturation, i.e. possible indication of overfitting, at around the size of 7x7. Based on this result the test continues with a model of degree 3 and a triangulation size of 7x7.

Figure A-1: The Matterhorn

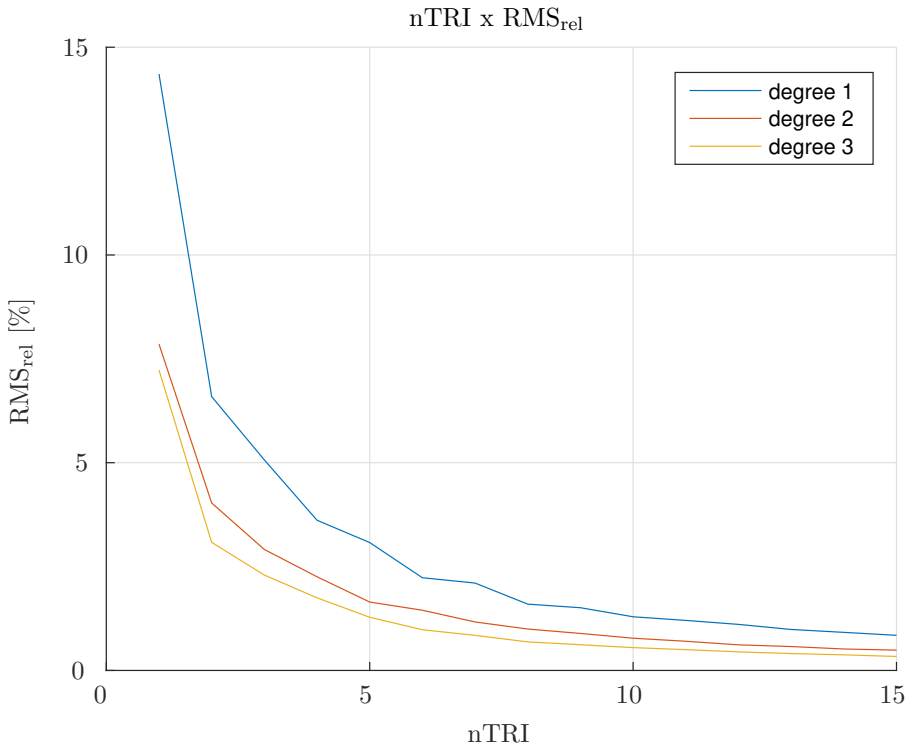


Figure A-2: This graphs shows how the triangulation influences the decrease of the the retaliative RMS error. The three plots belong to degree 1, 2 and 3 models with no continuity constraints.

Next the limits on the ERR are introduced. For each simplex it is determined what the ERR per parameter is using the orthogonalisation method. This method gives the opportunity to use higher degree models with a large variation in non-linearity. A high variation in non-linearity requires for some simplices to have a high degree model where other simplices suffice with a lower degree model. This method prevents over-fitting the "lower" non-linear simplices by introducing physical hard constraints on low-influential parameters, i.e. with a low ERR. An additional benefit is the reduction of the model size in terms of storage. Figure A-3 shows the influence of no continuity constraints (C^{-1}) and 0^{th} and 1^{st} order continuity constraints in combination with hard constraints on physical parameters with a low error reduction ratio (ERR). The purple line indicates the relative physical constraint count ($\mathrm{PC}_{\mathrm{count}}$), i.e. the percentage of parameters hard constraint to zero depending to the $\mathrm{ERR}_{\mathrm{limit}}$ set. The model is of degree 3 with a 7x7 triangulation. Notice the propagating effect in between simplices due to the increasing continuity constraints in combination with the increasing $\mathrm{PC}_{\mathrm{count}}$ physical constraints. Increasing continuity either requires an increase of model degree or decrease for the ERR limit. Based on this results the optimal Matterhorn model with zero order continuity will have a degree of 3, a triangulation size of 7x7 and 20% of its physical parameters hard constraint. This model is identified using a dataset with a total of 38701 elevation measurements and has 752 non-zero physical parameters. The calculation time equals 1.75 seconds. A plot of the Matterhorn using the MSBS model is displayed in Figure A-4.

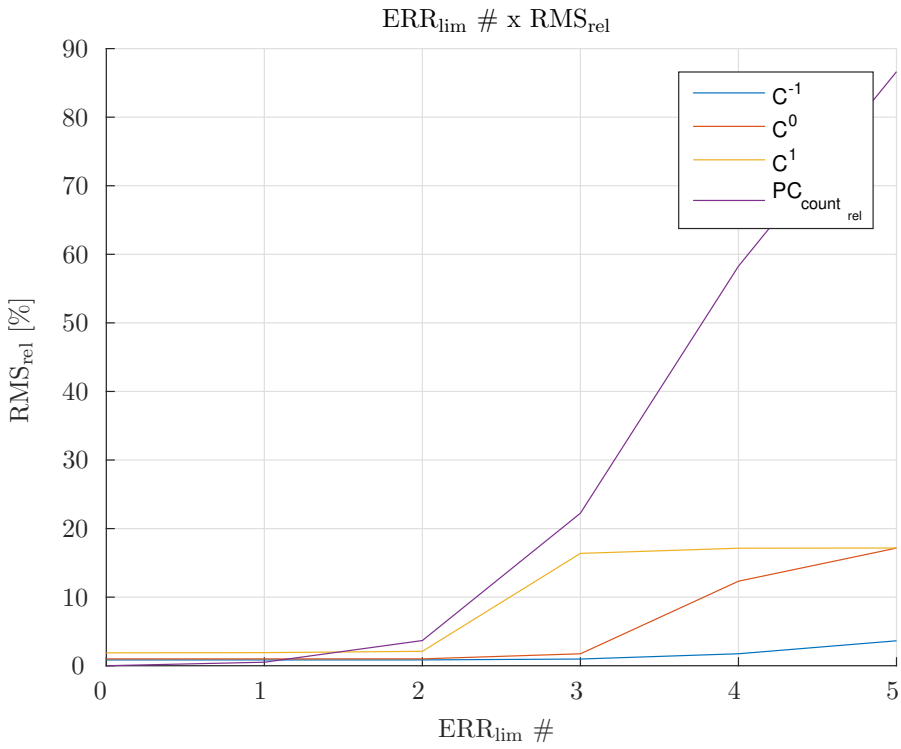


Figure A-3: The graphs shows the influence of no continuity constraints (C^{-1}) and 0^{th} and 1^{st} order continuity constraints in combination with hard constraints on physical parameters with a low error reduction ratio (ERR). The purple line indicates the percentage of parameters hard constraint to zero depending to the ERR limit set. The model is of degree 3 with a 7x7 triangulation.

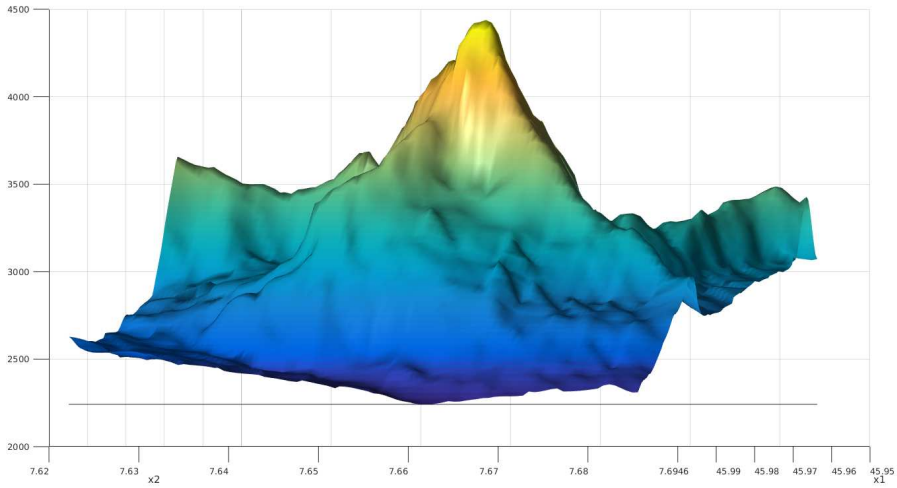


Figure A-4: The Matterhorn using multivariate simplex splines

Appendix B

Test Case Two - Aerodynamic Moment C_m

This test case is performed to determine the dimensionless moment model C_m around the y-axis. States selected for the two initial model structures are (α, q, M, δ_e) and (α, q, M, δ_e, h). Making it a 4-dimensional and 5-dimensional model respectively. These states are chosen in order to make a comparison to the baseline model. Other combination of states and dimensions are possible. Test runs are performed for degree 1 and degree 2, no continuity and $C1$ continuity, 8 triangulations and 5 minimum err values. Furthermore when some simplices are empty the triangulation is rotated to find an optimal orientation. For degree 2 models the decoupling of terms is also tested.

The data set used has 3211 and Hardover inputs on the elevator. It has more than 100,000 measurement points. The measurements are between Mach 0.2-0.6. The data is plotted in a so called 4D matrix plot in Figure B-1. This matrix plot shows the distribution of measurements between the states. The measurements of states α, q and δ_e are distributed normally over the measured range. However measurement for a higher Mach numbers, > 0.35, are limited. Matrix plot (4,3) shows clearly that many manoeuvres were executed between Mach 0.2 and approximately Mach 0.37 and around Mach 0.4. Other Mach numbers will create voids in the hypercube based triangulation because of missing measurements. Therefore it was decided for this model to omit measurements of Mach 0.4 and higher. Eight different triangulation types are tested. Type 1 is a single 4D hypercube. Other triangulation types have extra hypercube(s) on certain axis. Table B-1 gives a summery of the total number of simplices per triangulation type in the second column. In the third column the number of empty simplices and the fourth column the number of empty simplices after an "optimal" rotation are tabulated. After rotating triangulations types 1 to 5 all have data points in all simplices.

Next two plots (Figure B-2 and Figure B-3) are created to see the performance, measured in the relative root mean square error (rms_{rel}), for each triangulation type. The models in plot B-2 do not have continuity. The green lines represent $S_1^{-1}(\mathcal{T}_{nTRI})$, the blue lines $S_2^{-1}(\mathcal{T}_{nTRI})$ and the red lines $S_2^{-1}(T_{nTRI})$ where terms are decoupled. The dotted lines indicated the optimal rotation triangulation. The colour representation in plot B-3 is similar only the

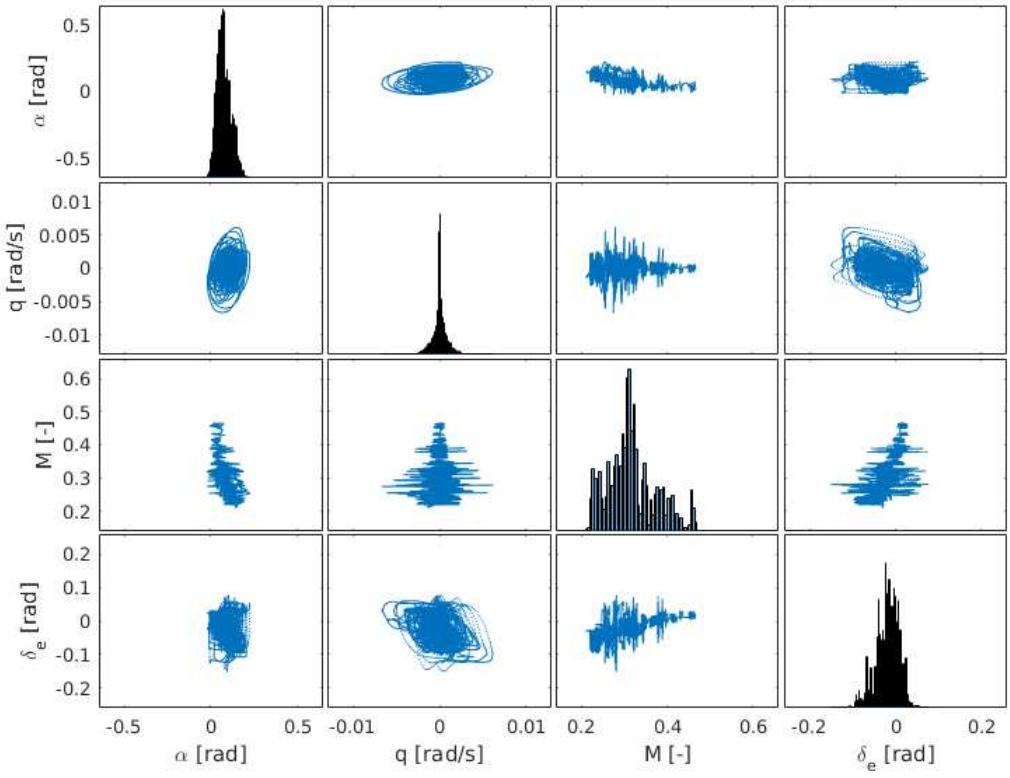


Figure B-1: 4D plot of flight test data

Table B-1: The number of simplices and empty simplices for different triangulation sizes and the influence of rotating the triangulation

nTRI	Number of Simplices	Empty Simplices	Empty after Rotation
1: [1 1 1 1]	24	0	0
2: [2 1 1 1]	48	6	0
3: [1 2 1 1]	48	12	0
4: [1 1 1 2]	48	9	0
5: [2 2 1 1]	96	21	0
6: [2 1 1 2]	96	24	3
7: [1 2 1 2]	96	32	8
8: [2 2 1 2]	192	80	29

models have C0 continuity. When there is no continuity a small performance increase can be seen when the triangulation size is increased. For C0 continuity there is minimal difference between the triangulations for the degree 1 model and the decoupled degree 2 model. Neither of the models is convincing. To prevent over-fitting the degree 1 with [1 1 1 1] triangulation model is used in this demo. An interesting note is that the optimal rotation triangulation of this particular model performs better than the "original" triangulation.

In Figure B-4 the influence of hard constraints based on the error reduction ratio (err) of terms is plotted. When increasing the minimum level of err almost no terms are hard constraint. However with err level #6 this increases. While the RMS_{rel} almost doesn't change the the coefficient of determination drops to almost zero. In the final model it was decided not to hard constraint any terms. The final model is $C_m(\alpha, q, M, \delta_e) \in S_1^0(\mathcal{T}_{1111=24})$

In Table L-1 (appendix) individual simplex information and statistical analysis is tabulated. Two simplices, 3 and 13, have high maximum variances. One explanation can be the relatively low data density of this simplex. It is checked that it is neither caused nor solved by "removing" the continuity. Other simplices which require extra attention are simplex 1 and 2. Both have a low ERR value, meaning that the available terms do not have a high enough approximation power. An explanation can be that these simplices cover a more non-linear part of the flight envelope. A solution is to use a more complex model structure for these simplices. But one has to be careful and needs to check for data errors or noise first to prevent unwanted modelling.

Tables L-2 and L-3, in appendix L, show the physical model parameters of the initial triangulations and the optimal rotation triangulation respectively. While rotating the simplices are placed differently within the hypercube, this creates other neighbouring simplices compared to other rotations. This in combination with continuity gives different results for "poor" performing simplices. The poor performing simplices are again simplex 3 and 13 in Table L-2 (appendix) and 1 and and 2 in Table L-3 (appendix) . These simplices have extreme high parameters for C_{m_q}, but less high compared to the rotated triangulation. Something else that requires attention is that some parameters C_{m_α} are positive. The results of the 5 dimensional model are not plotted. They did not differ very much and the parameter related to the altitude h was estimated to be zero. Indicating it has almost no influence.

To conclude the model created for C_m is very linear. The simplex parameters are very similar. Increasing the dataset used and the flight envelope will probably change this. Another point of interest is to change the initial model structure and add states like the side-slip angle β. Also some simplices need extra attention. In order to improve the quality of these simplices additional constraints are required or they are extrapolated afterwards. It will be advantageous when outlier data points can be included or empty simplices can be modelled to guarantee inner model continuity. A solution is to extrapolate into these simplices using modelled surrounding simplices and outlying data-points. Also a dataset with multiple clusters can be modelled by individual hypercubes triangulations. The void between these clusters can be extrapolated by stretched hypercubes using the modelled neighbouring hypercubes. Overall the physical-spline model parameters performed good compared with the existing baseline model. In Table B-2 the estimated parameters, including mean and variance

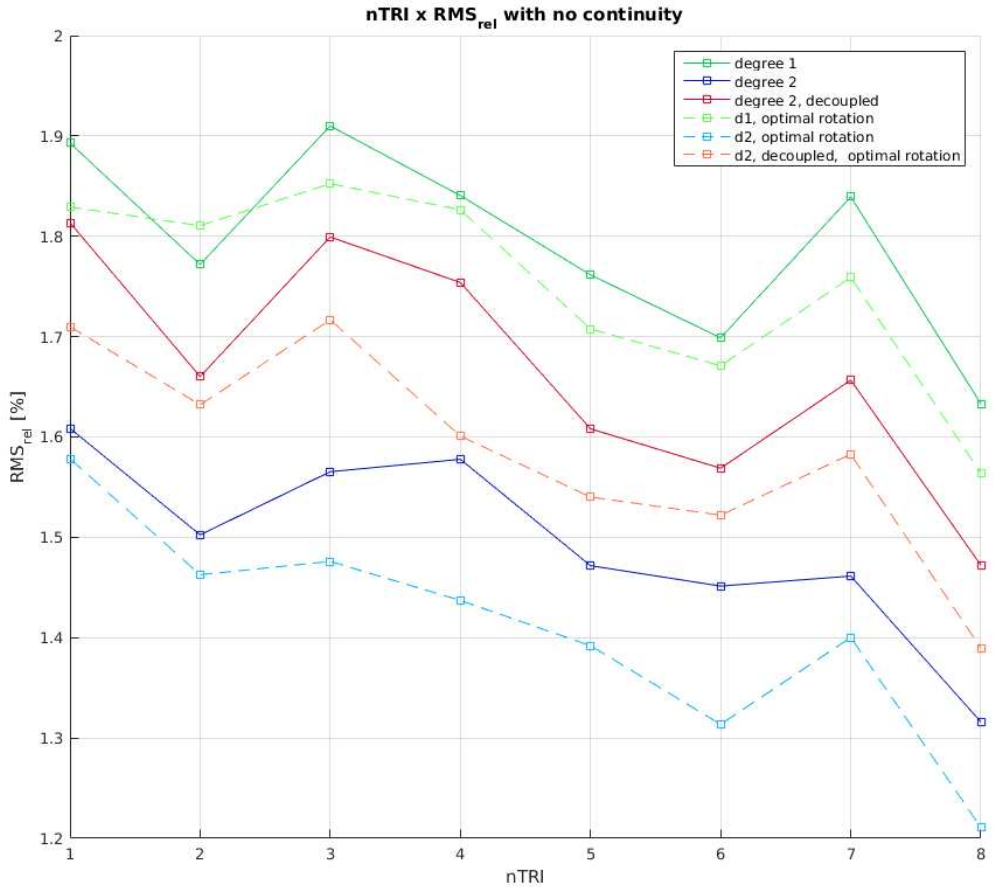


Figure B-2: nTRI x RMS_{rel} with no continuity

is given. Table B-3 compares the results with the current base-line model from (Hoek et al., 2017). Figure B-5 shows the plot of the validation output and the model output using the validation dataset.

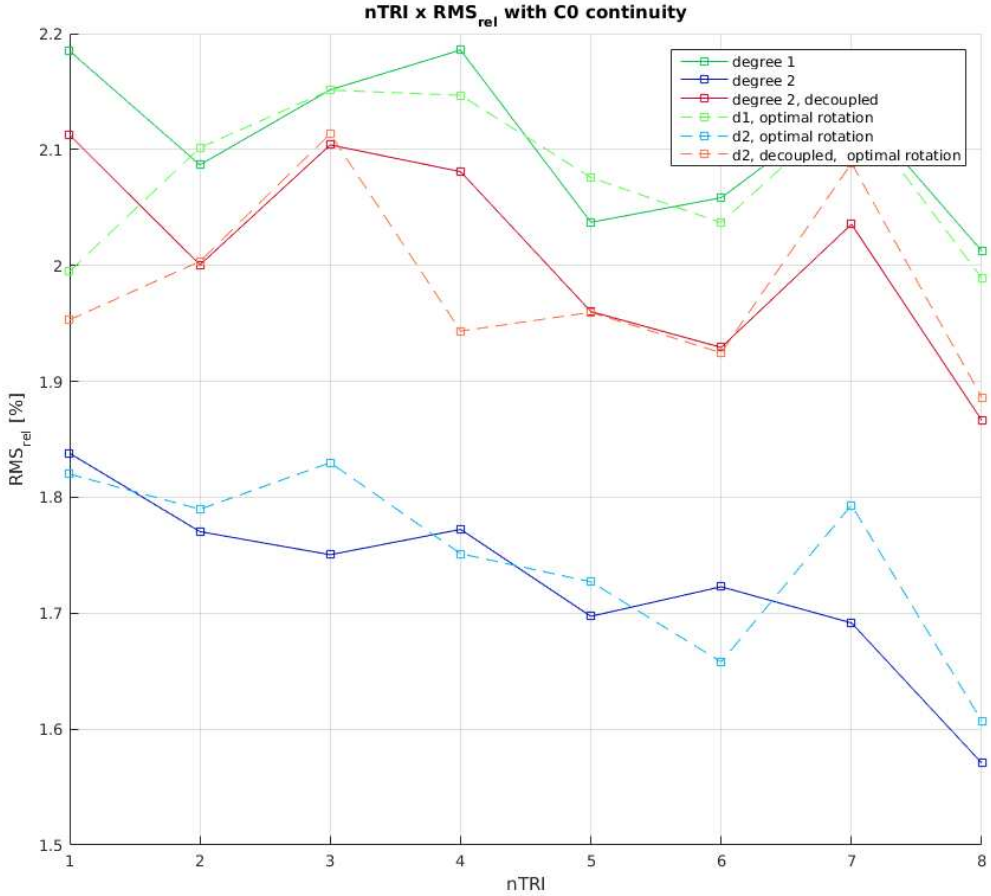


Figure B-3: nTRI x RMS$_{rel}$ with C0 continuity

Table B-2: Estimated parameters mean and variances for $n = 4$, $d = 1$, $r = 0$, [1 1 1 1]

	$\bar{\boldsymbol{\theta}}$	$\boldsymbol{\theta}_{min}$	$\boldsymbol{\theta}_{max}$	$\bar{\boldsymbol{\sigma}}(\theta)^2$	$\boldsymbol{\sigma}(\theta)^2_{min}$	$\boldsymbol{\sigma}(\theta)^2_{max}$
C_{m_0}	-0.06	-0.14	-0.03	3.51×10^{-5}	1.74×10^{-6}	2.10×10^{-4}
C_{m_α}	-0.4	-0.53	-0.34	4.67×10^{-5}	1.74×10^{-5}	8.61×10^{-5}
C_{m_q}	-10.42	-12.96	-8.88	3.35×10^{-2}	2.62×10^{-2}	5.15×10^{-2}
C_{m_M}	0.21	0.13	0.56	2.41×10^{-4}	1.55×10^{-5}	1.40×10^{-3}
C_{m_δ}	-1.11	-1.49	-0.95	3.35×10^{-4}	4.08×10^{-5}	1.70×10^{-3}

Table B-3: Validation fit statistics compared to the base-line model, using 20% of the data as validation set

	RRMSE (%)	RRMSE_B (%)	R^2	R^2_B
C_m	2.1	5.8	0.84	0.76

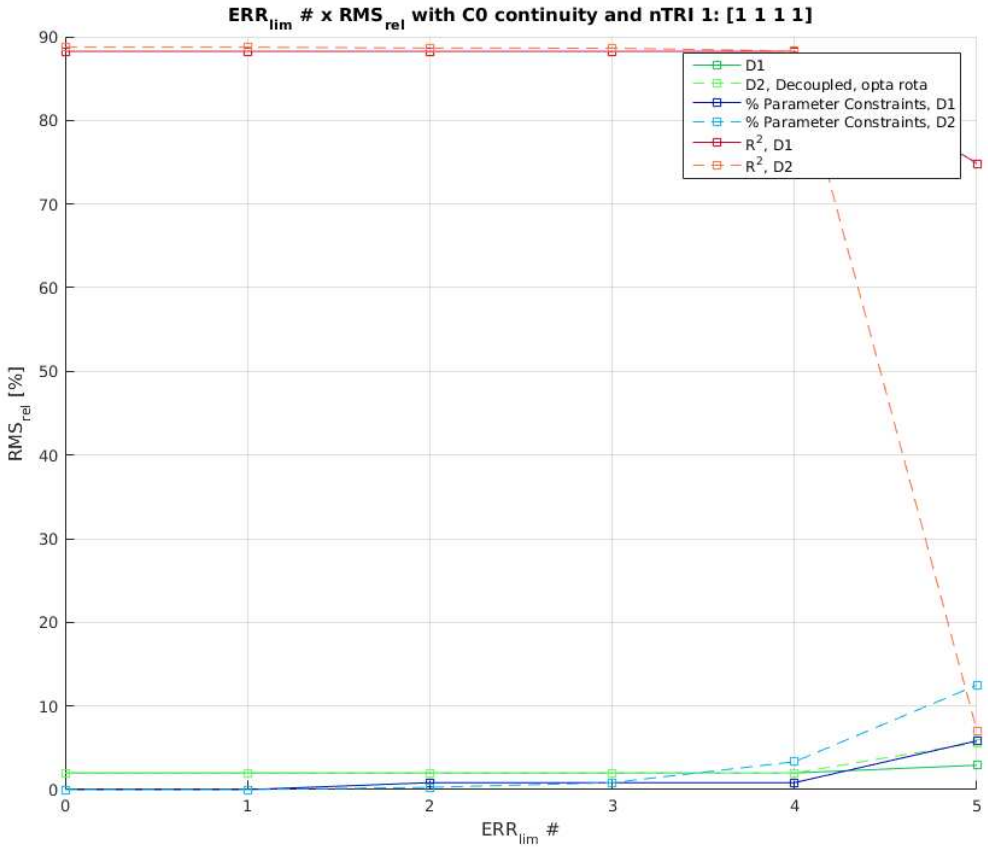


Figure B-4: $ERR_{lim} \times RMS_{rel}$ with C0 continuity and nTRI 1: [1 1 1 1]

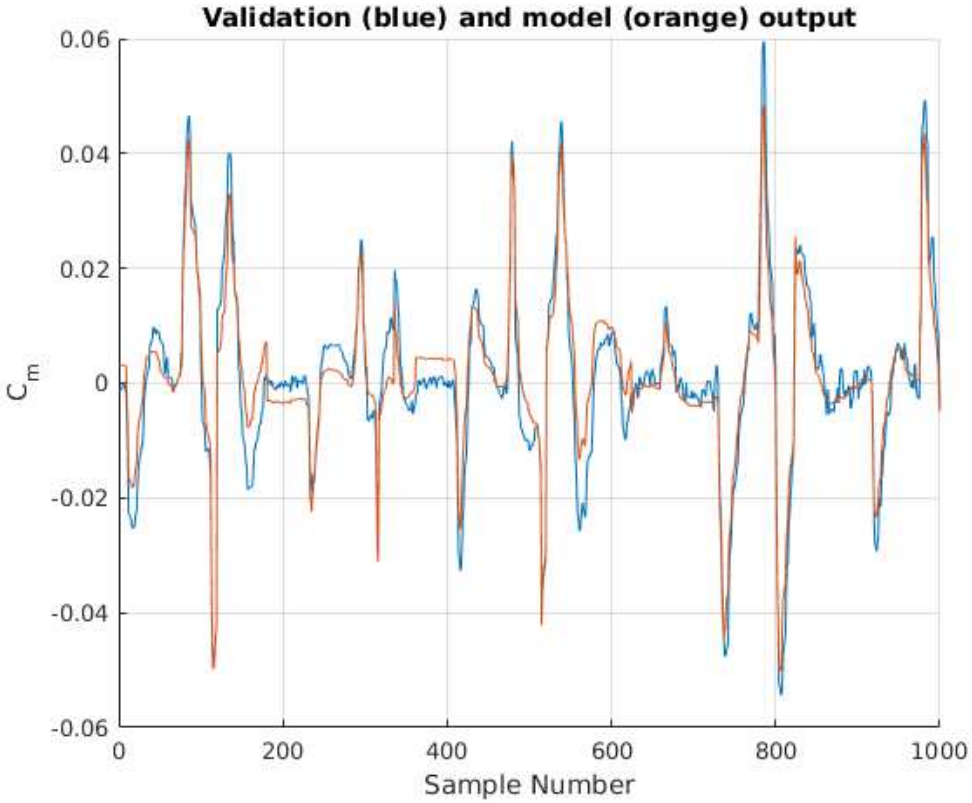


Figure B-5: Model output of C_m when using validation dataset. Blue is the output of the validation and orange the output of the model using the validation dataset

Appendix C

Differential Constraints Boundary Direction

This section contains a description of the proposed solution to determine the correct normal direction (sign) of the differential constraints on the triangulation boundary facets of simplices.

The vectors $\hat{\mathbf{u}}$ and $\hat{\mathbf{p}}$ are unit vectors, hence the hat symbol. The vectors are originated somewhere on the edge of a (n-1)-face of a n-simplex. $\hat{\mathbf{u}}$ indicates the direction of the differential constraints and is by definition perpendicular to the edge. $\hat{\mathbf{p}}$ indicates the direction to the out of edge vertex, see figure 1. The Matlab function `bsplinen_calcEdgeNormals.m` is used to create the constraints. However the vector direction $\hat{\mathbf{u}}$ has to be verified if it is pointing outward, away from the simplex. For a n-simplex the out of edge vertex is always on one side of the extend (n-1)-face. Therefore the angle $\boldsymbol{\theta}$, between $\hat{\mathbf{p}}$ and the inward pointing direction vector $-\hat{\mathbf{u}}$, is always $0 < \boldsymbol{\theta} < 90$. Since unit vectors are used two isosceles triangles can be formed with two different bases. The base length is determined by the magnitude of θ and $180 - \theta$. Using the properties of an isosceles it is possible to prove that one base length is always longer than the other and corresponds to the direction vector pointing outward, away from the simplex.

This is illustrated in the figure where $\overline{\nu_1\,\nu_2}$ is the 1-face of the 2-simplex ν_0, ν_1, ν_2. The out of edge vertex is ν_0. In the figure this 1-face is a line. n_1 and n_2 are forming the bases of the isosceles triangles. Their length is calculated using

$$n_1 = ||\hat{\mathbf{p}} - \hat{\mathbf{u}}|| \tag{C-1}$$

$$n_2 = ||\hat{\mathbf{p}} - (-\hat{\mathbf{u}})|| \tag{C-2}$$

Using the properties of an isosceles in combination with the property $0 < \theta < 90$ it is easy to verify that one base, n_1, is longer than the other base. This base with the longest length corresponds the the correct direction vector $\hat{\mathbf{u}}$.

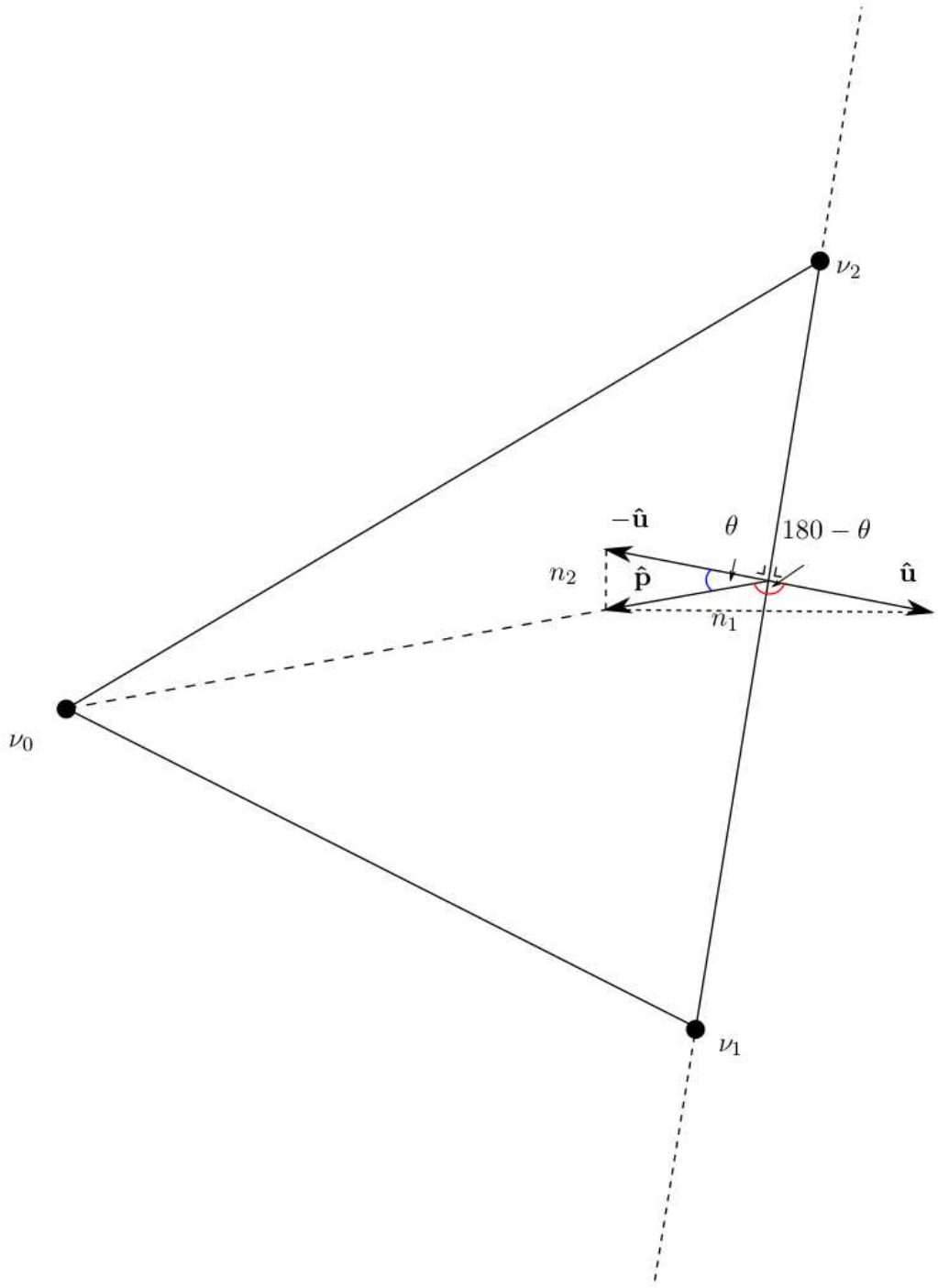

ν_2
$-\hat{\mathbf{u}}$
θ
$180-\theta$
n_2
$\hat{\mathbf{p}}$
$\hat{\mathbf{u}}$
n_1
ν_0
ν_1

Figure C-1

Matlab Code

The aforementioned method is applied using the Matlab code below

```
mid = sum(phie)/n;  % Mid-point of simplex edge normal
phio = setdiff(PHI(TRI(i,:),:),phie,'rows');     % Out of edge vertex
po = (phio-mid)/norm(phio-mid); % norm direction of out of edge vertex
ne1 = norm(normvect - po);
ne2 = norm(normvect.*-1 - po);
if ne2 > ne1
    normvect = normvect .* -1;
end
```

Appendix D

Test Case Three - Differential Boundary Constraints

Measured and instrument noise in aircraft test-data results in divergent behaviour on the triangulation boundaries. Poor data distribution in boundary simplices in combination with the higher barycentric Bernstein terms located on triangulation edges are found to be the main cause for this problem. This is illustrated by figure D-1 This problem was partially solved by introducing differential constraints on the location of B-coefficients normal to the edge facets boundaries. The differential constraints of B-coefficients shared by multiple simplices will have the mean direction of the different normal directions edge facets. Figure D-2 shows the location and direction of the differential constraints on the used triangulation. The differential constraints set the second order derivatives to zero. Figure D-3 shows the effect of using differential constraints. The divergent behaviour still exists. However the magnitude of the divergence is significantly lower compared to the no-constraints experiment. For the experiment involving physical inequality constraints two bounds are set on the stability derivatives C_{m_α} and $C_{m_{\delta_e}}$ with grid density $\tau = d$. The bounds are based on earlier created models of the Cessna Citation II aircraft and test flights and are given by

$$-2 \leq C_{m_\alpha} \leq 0 \tag{D-1}$$

$$-3 \leq C_{m_{\delta_e}} \leq -0.5 \tag{D-2}$$

The result is plotted in figure D-4. The divergent behaviour on the boundaries completely disappear. This example shows clearly the advantage of having a physical interpretation of model. Not only are they used to obtain more knowledge, they also remove errors and control the flight-test data in a more appropriate way. Beside removing the divergent behaviour on the boundaries, the physical inequality constraints also contribute in a more realistic aerodynamic model. The physical inequality constraints only control the derivatives on the grid points location. Between the grid points it is possible that the constraints are violated. Therefore the physical inequality constraints can be considered as a relaxation method. Root finding algorithms for higher degree and dimensional models are highly non-linear. Since

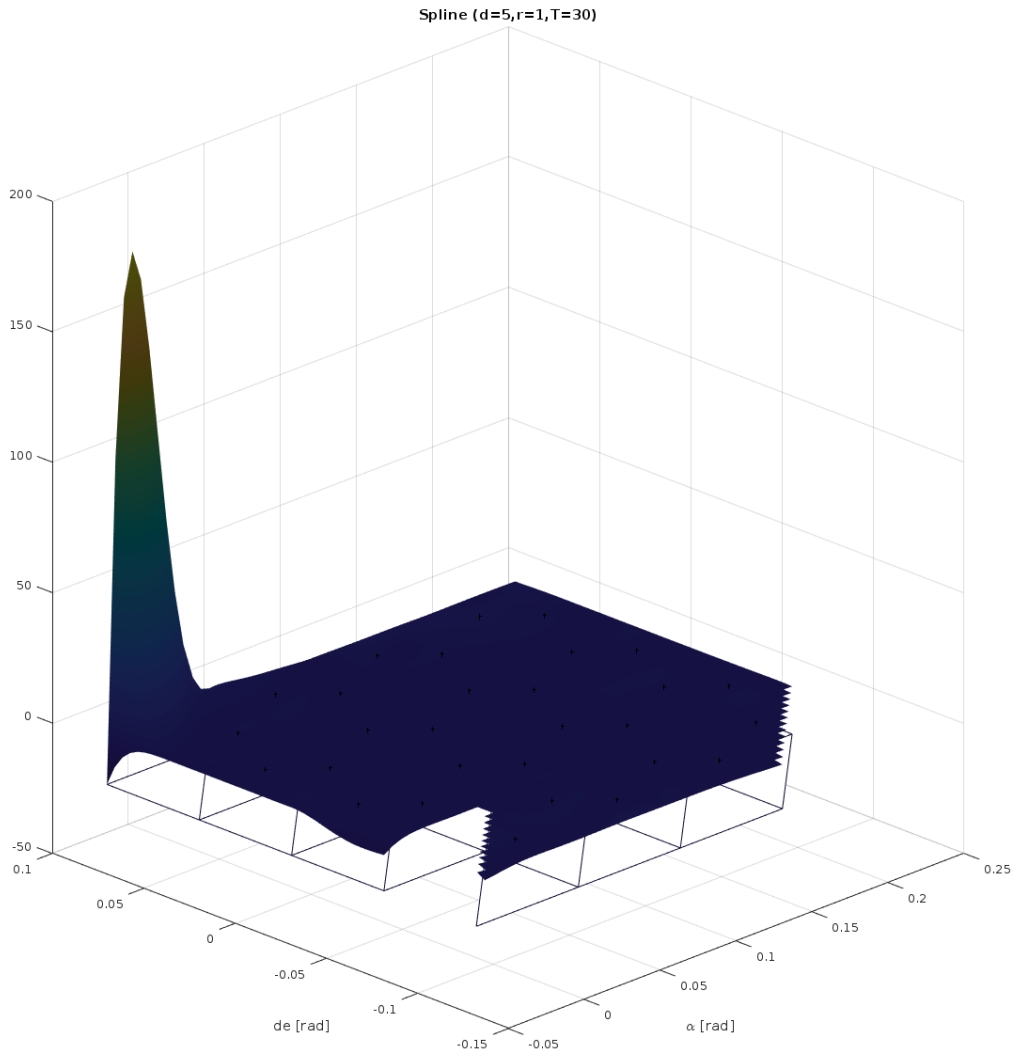


Figure D-1: Spline model of $C_m(\alpha, \delta_e)$ using flight-test data with no constraints added to the optimisation process

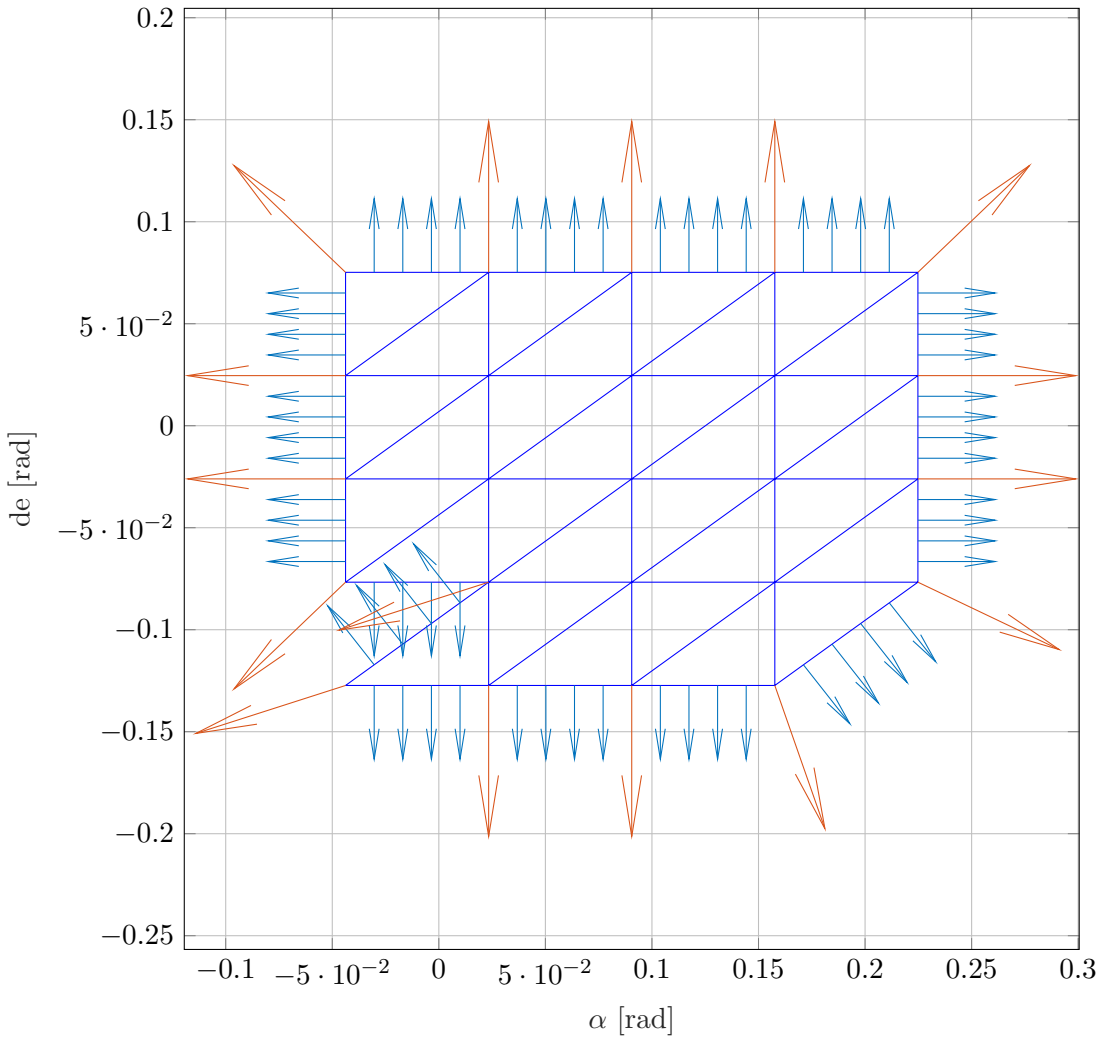


Figure D-2: The triangulation including the differential constraints location and direction. Orange vectors indicate shared differential constraints.

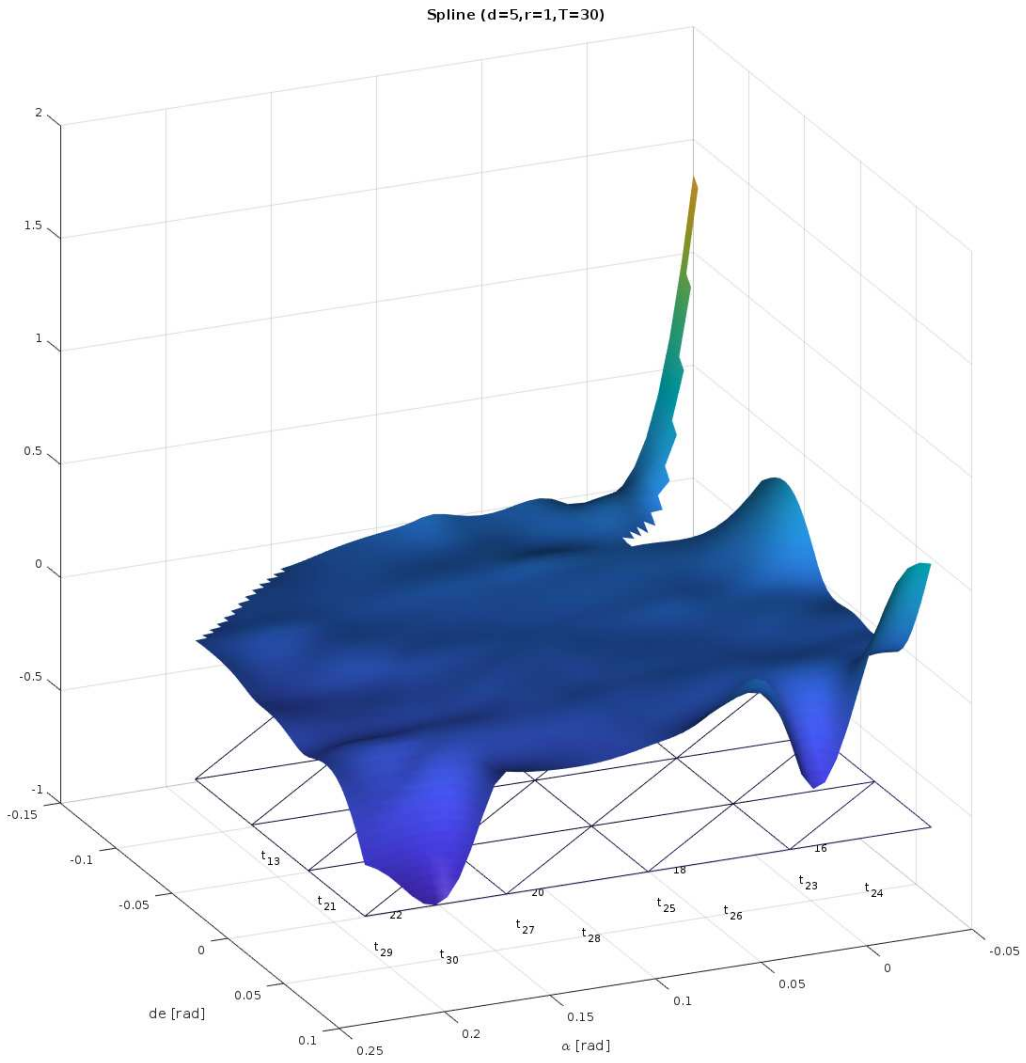


Figure D-3: Spline model of $C_m(\alpha, \delta_e)$ using flight-test data with second order boundary facets differential constraints added to the optimisation process

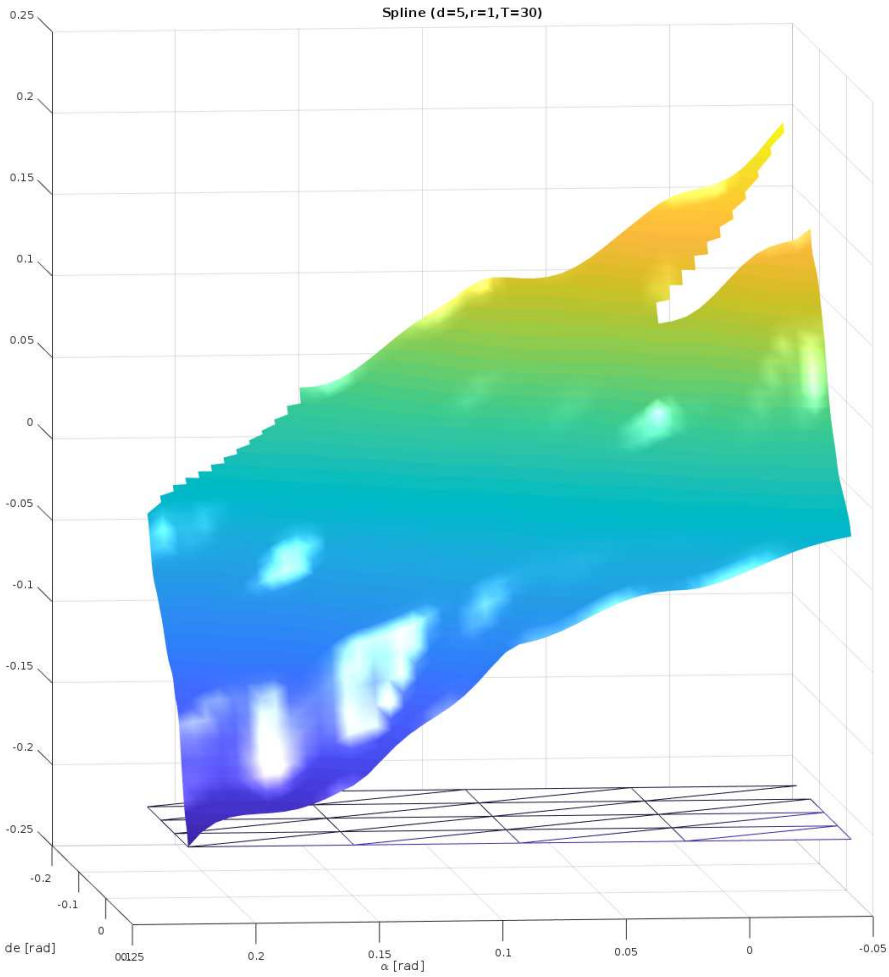


Figure D-4: Spline model of $C_m(\alpha, \delta_e)$ using flight-test data with bounded physical inequality constraints added to the optimisation process. The physical inequality constraints are based on advance aerodynamic information of the aircraft using the stability derivatives.

the convex optimisation problem is preferable it is decided to use the grid-approach. In the experiment the grid density of $\tau = d$ was found to be appropriate. The grid, in combination with this density, controls the stability derivatives of the aerodynamic model very well during the optimisation process.

Appendix E

Test Case Four - Physical Constraints

To obtain better insight knowledge what the physical constraints actually do and how they influence the parameter estimation process and other constraints, a test is set up. A Franke function is created and noise is introduced to the data. The dataset is split in a 60% identification set and a 40% validation set. Before noise is added to the data, a MSBS model was created with degree $d = 5$ and no continuity using a $\mathcal{T}(6, 6)$ hypercube triangulation. The derivative with respect to the physical variables x_1 and x_2 was calculated. These derivatives are used to determine the derivative values based on grid-points on a per-simplex basis. These derivative values are used later on to create bounded physical derivative inequality constraints for each simplex and serve as physical a-priori information. Eight different models are created using the noisy data and compared afterwards. All models have degree $d = 5$, a $\mathcal{T}(6, 6)$ hypercube triangulation and no continuity or C^0 continuity. Models 1 and 2 have no constraints, models 3 and 4 have physical derivative bounded inequality constraints, models 5 and 6 have model structure equality constraints based on a stepwise orthogonalisation methods and a per-simplex validation over-fit check and models 7 and 8 have both constraints combined.

E-1 Results and Discussion

The statistical results are tabulated in E-1. Comparing model 2 with model 1 the influence of continuity is clearly visible. Continuity smooths the model while decreasing the variance and therefore reducing the effect of noise. The other results show that the constraints have a positive effect in improving the model. Especially the model structure constraints prevent the model from over-fitting. Table E-2 shows information about global derivative values, including their maximum, minimum, mean and variance. For comparison the derivative values of the Franke function is given in the bottom row. An interesting observation is that all physical constraints have the ability to lower the variance of the derivative significantly and obtain a near identical mean compared to the reference Franke function without noise. The previous is also clearly visible in figures E-1, E-2, E-3 and E-4, plotting the results for model 1 & 2, model 3 & 4, model 5 & 6, and model 7 & 8 respectively. Smoothness constraints are able

to suppress the noise to a certain extent, however high variance and divergent behaviour on triangulation boundary still exists. Adding physical information in the form of constraints have a great advantage as is visible in the variance plots E-5, E-6, E-7 and E-8, again plotting the results for model 1 & 2, model 3 & 4, model 5 & 6, and model 7 & 8 respectively.

E-1-1 Conclusion

In general over-fitting the model depends on certain aspects. Among them is the triangulation density, i.e. how many hypercubes are combined. Optimising for an optimal triangulation is a difficult non-linear process. Using a Kuhn triangulation and combining multiple hypercubes makes this process easier. However sometimes some parts of the domain do require a more dense triangulation with the consequence other parts of the domain are too dense. Another aspect is the degree of the basis functions. In barycentric space the degree determines the very important B-net. Influencing the model structure in barycentric space is only possible by changing the degree. The continuity, often determined by a model requirement, also influences the degree. Higher order continuity requires a higher degree. A relation between the real approximation power, degree and continuity was found to be $d-r-1$. Similar to the triangulation density a dynamic approximation power for different parts of the domain can be required. The physical-splines formulation turns out to have favourable properties. The B-net can be adjusted via a physical model structure selection using orthogonalisation and therefore a prevention of over-fitting. Furthermore physical information can be introduced to the optimisation in the form of bounds. For aerodynamic modelling these bounds are based on the extensively studied stability derivatives. The latter inequality constraints lower the model parameter variance, prevent over-fitting, removes and prevents propagating erroneous data and introduces a more easy single-optimisation extrapolation approach. Researching the newly obtained properties of the physical-splines the following conclusions are made:

- a good practice for the maximum number of model structure constraints was found to be $\hat{d}_r$,
- it is better to not constraint linear terms,
- if the optimisation is infeasible a good practice to constraint coupled terms only,
- continuity presses the bound of the derivatives,
- therefore do no tighten the derivative inequality bounds too much or,
- stretch these set bounds when increasing the continuity and check afterwards if derivatives are on the bounds,
- the grid used to determine the linear derivative inequality constraints based on the B-net location has a better performance than randomly creating this grid,
- grid points equal to the B-net density turned out to be sufficient to make the derivative constraints effective,
- increasing the grid density showed little effect,

Table E-1: Statistical results using the validation dataset, where R2 is the coefficient of determination and RMSE and RRMSE the root mean squared error and relative root mean squared error.

	Model	R2	RMSE	RRMSE
No constraints	**Model 1**, C^{-1}	0.98	4.2	3.2%
	Model 2, C^0	0.99	3.0	2.3%
Physical derivative bounded constraints	**Model 3**, C^{-1}	0.99	2.9	2.3%
	Model 4, C^0	0.99	3.0	2.3%
Physical Model Structure constraints	**Model 5**, C^{-1}	0.99	3.2	2.5%
	Model 6, C^0	0.99	2.9	2.2%
Combined constraints	**Model 7**, C^{-1}	0.99	3.0	2.3%
	Model 8, C^0	0.99	2.9	2.2%

- test for over-fit, using validation data, per simplex individual, without constraints yields good results,
- the stepwise orthogonalisation preferable order determination shows better results using a normalised dataset,
- derivative bounded constraints can be used for extrapolation,
- by doing so, the extrapolated simplices in combination with continuity constraints have a small influence on the interpolated simplices,
- physical based constraints perform better in terms of over-fit prevention, lowering parameter variance and smoothing compared to continuity constraints,
- physical derivative constraints remove and prevent propagating erroneous data and divergent behaviour at the triangulation boundaries.
- too many constraints can make the optimisation infeasible or will results in a trivial solution, i.e. a constant model or all simplices individual models are equal,
- the Kuhn triangulation can also be used for cluster detection in the flight-test data.

Table E-2: The output of the derivative spline functions with respect to x_1 and x_2 using grid data points on top of the triangulation. The maximum, minimum, mean and variance are calculated.

Model	$dx_{1_{\max}}$	$dx_{1_{\min}}$	$\bar{dx_1}$	$var(dx_1)$	$dx_{2_{\max}}$	$dx_{2_{\min}}$	$\bar{dx_2}$	$var(dx_2)$
Model 1, C^{-1}	44.75	-17.85	-0.16	15.65	23.47	-73.06	-0.55	19.84
Model 2, C^0	44.75	-21.51	-0.33	7.98	10.79	-67.77	-0.51	10.84
Model 3, C^{-1}	2.59	-3.27	-0.41	1.03	2.41	-3.24	-0.33	1.23
Model 4, C^0	2.59	-3.27	-0.41	1.03	2.41	-3.24	-0.32	1.22
Model 5, C^{-1}	14.06	-10.26	-0.42	1.61	2.71	-14.24	-0.34	1.57
Model 6, C^0	4.08	-5.02	-0.43	1.41	3.81	-10.48	-0.29	1.73
Model 7, C^{-1}	2.39	-3.22	-0.42	0.96	2.29	-3.14	-0.33	1.15
Model 8, C^0	2.57	-3.25	-0.41	0.99	2.41	-3.24	-0.32	1.20
Franke Function	2.58	-3.28	-0.40	1.00	2.41	-3.23	-0.32	1.19

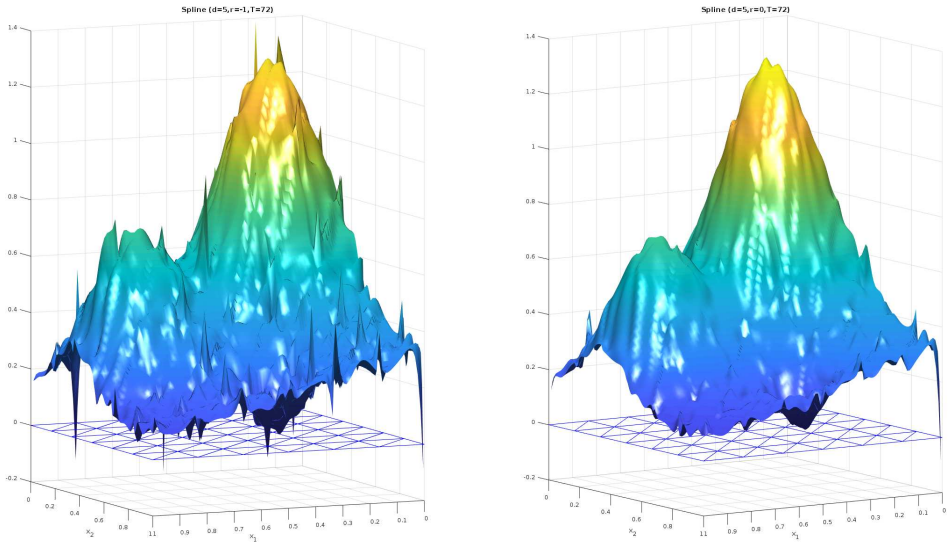


Figure E-1: Franke function containing noise with no constraints and no continuity left and no constraints and 0^{th} order continuity right.

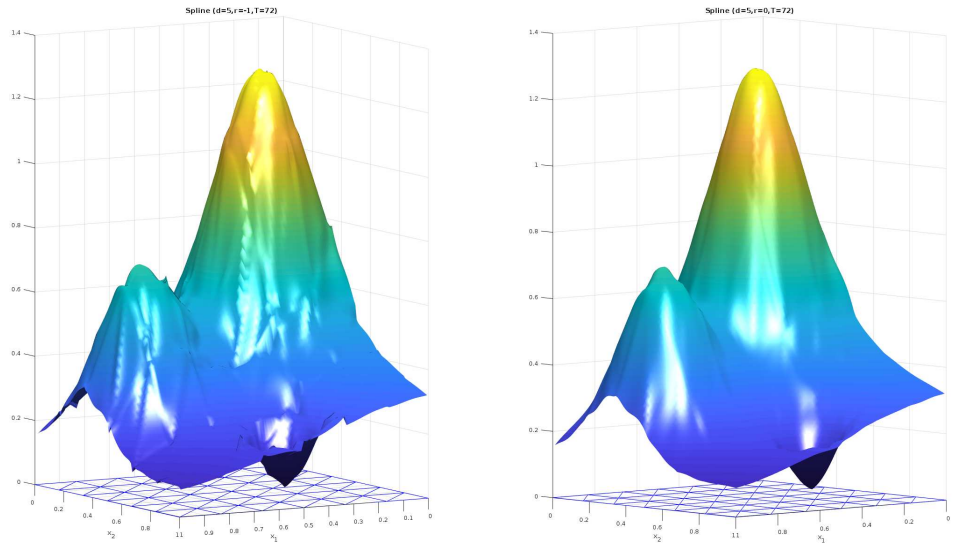


Figure E-2: Franke function containing noise with no continuity left and 0^{th} order continuity right. Both models have physical bounded derivative inequality constraints.

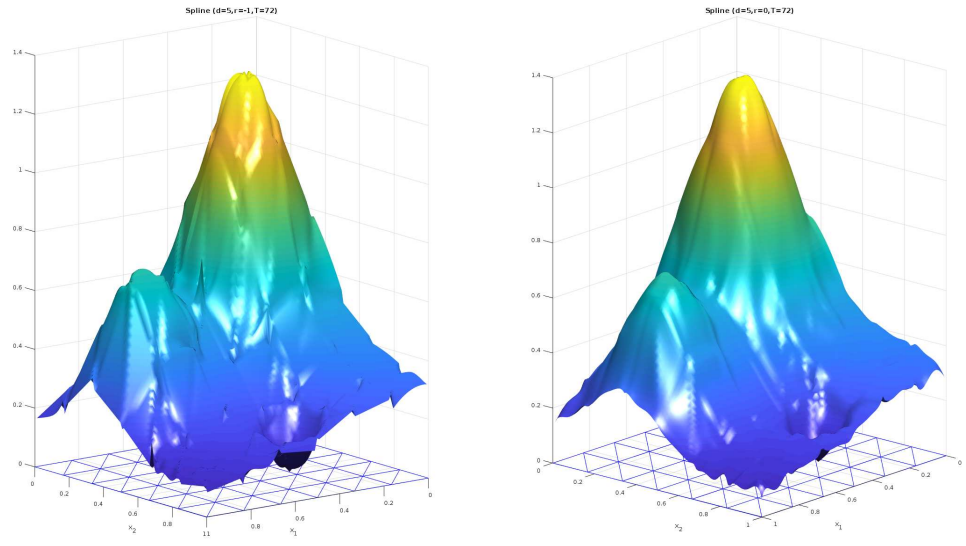


Figure E-3: Franke function containing noise with no continuity left and 0^{th} order continuity right. Both models have physical model structure selection equality constraints.

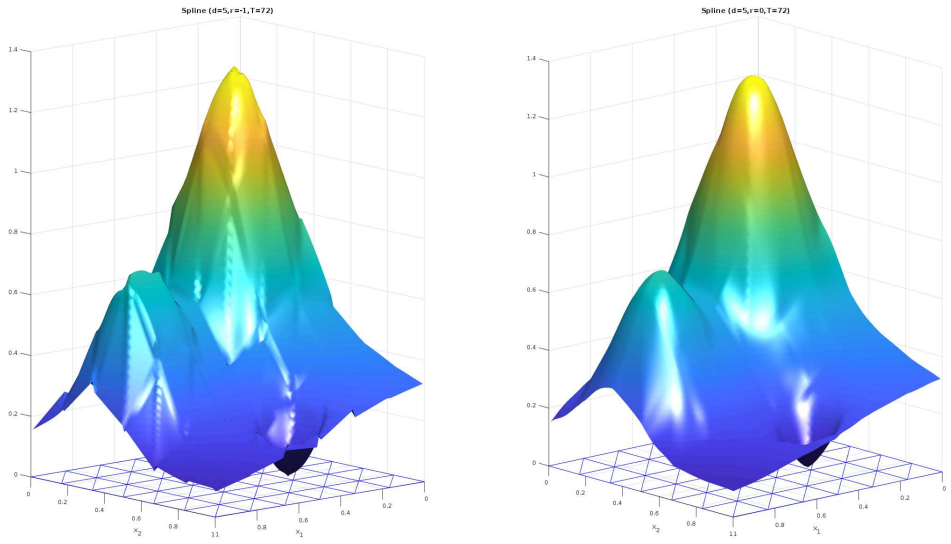


Figure E-4: Franke function containing noise with no continuity left and 0^{th} order continuity right. The two models have both constraints combined.

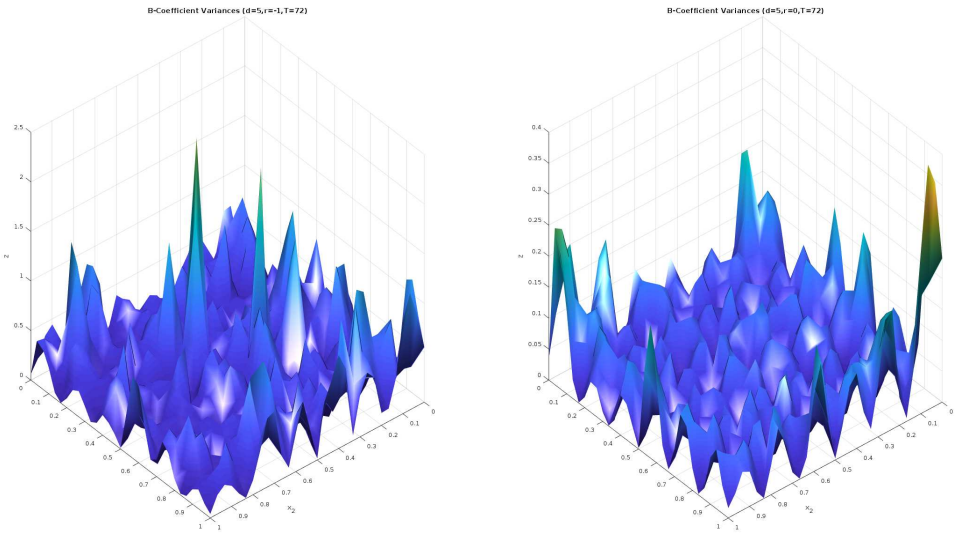


Figure E-5: Variance of Franke function containing noise with no constraints and no continuity left and no constraints and 0^{th} order continuity right.

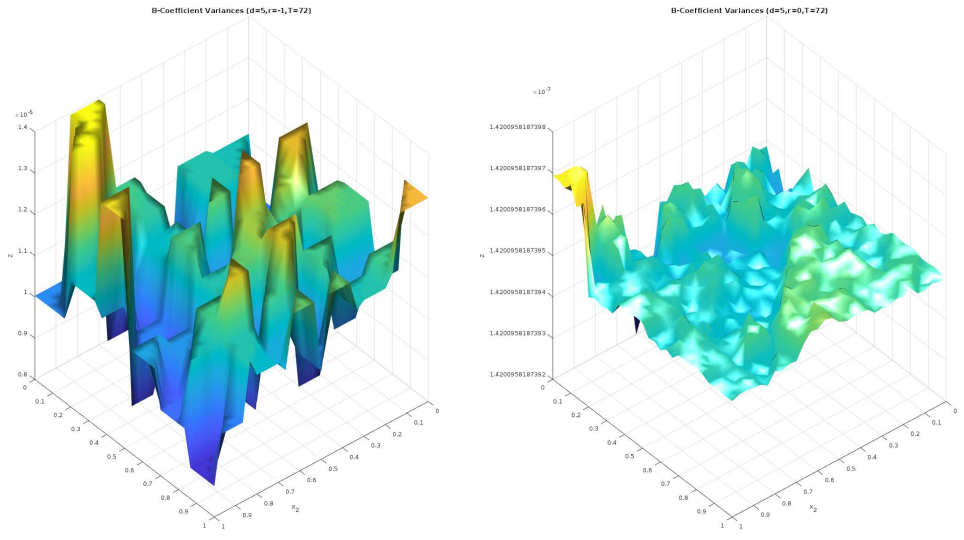


Figure E-6: Variance of Franke function containing noise with no continuity left and 0^{th} order continuity right. Both models have physical bounded derivative inequality constraints.

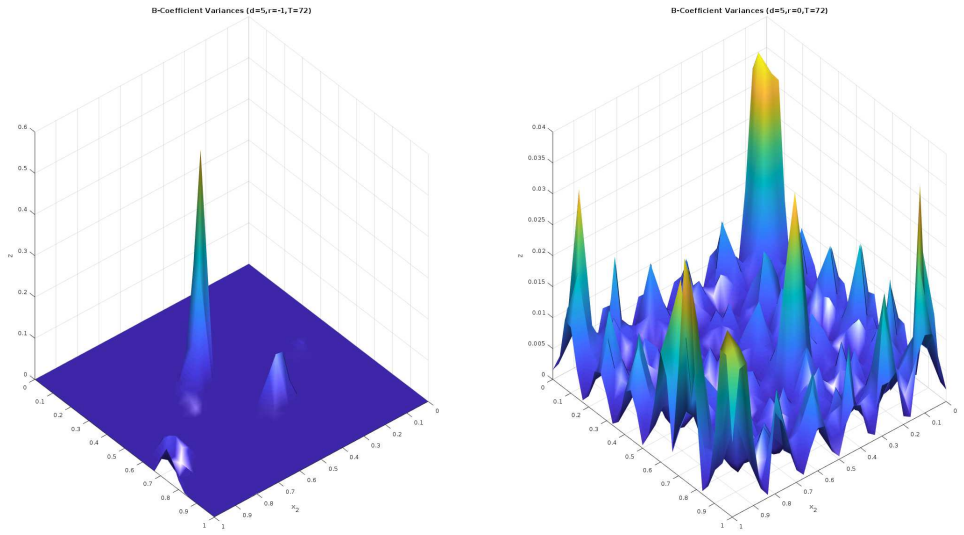


Figure E-7: Variance of Franke function containing noise with no continuity left and 0^{th} order continuity right. Both models have physical model structure selection equality constraints.

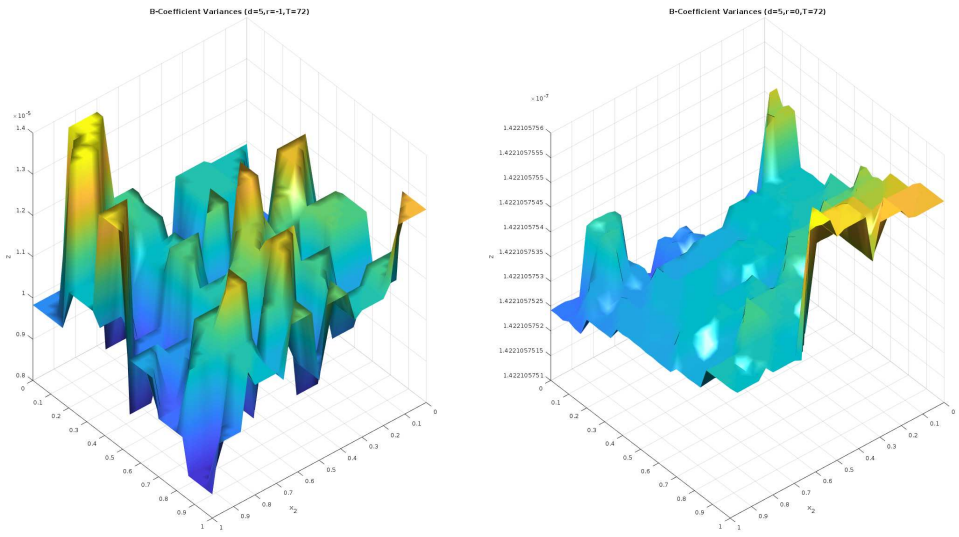


Figure E-8: Variance of Franke function containing noise with no continuity left and 0^{th} order continuity right. The two models have both constraints combined.

Appendix F

Reference Frames

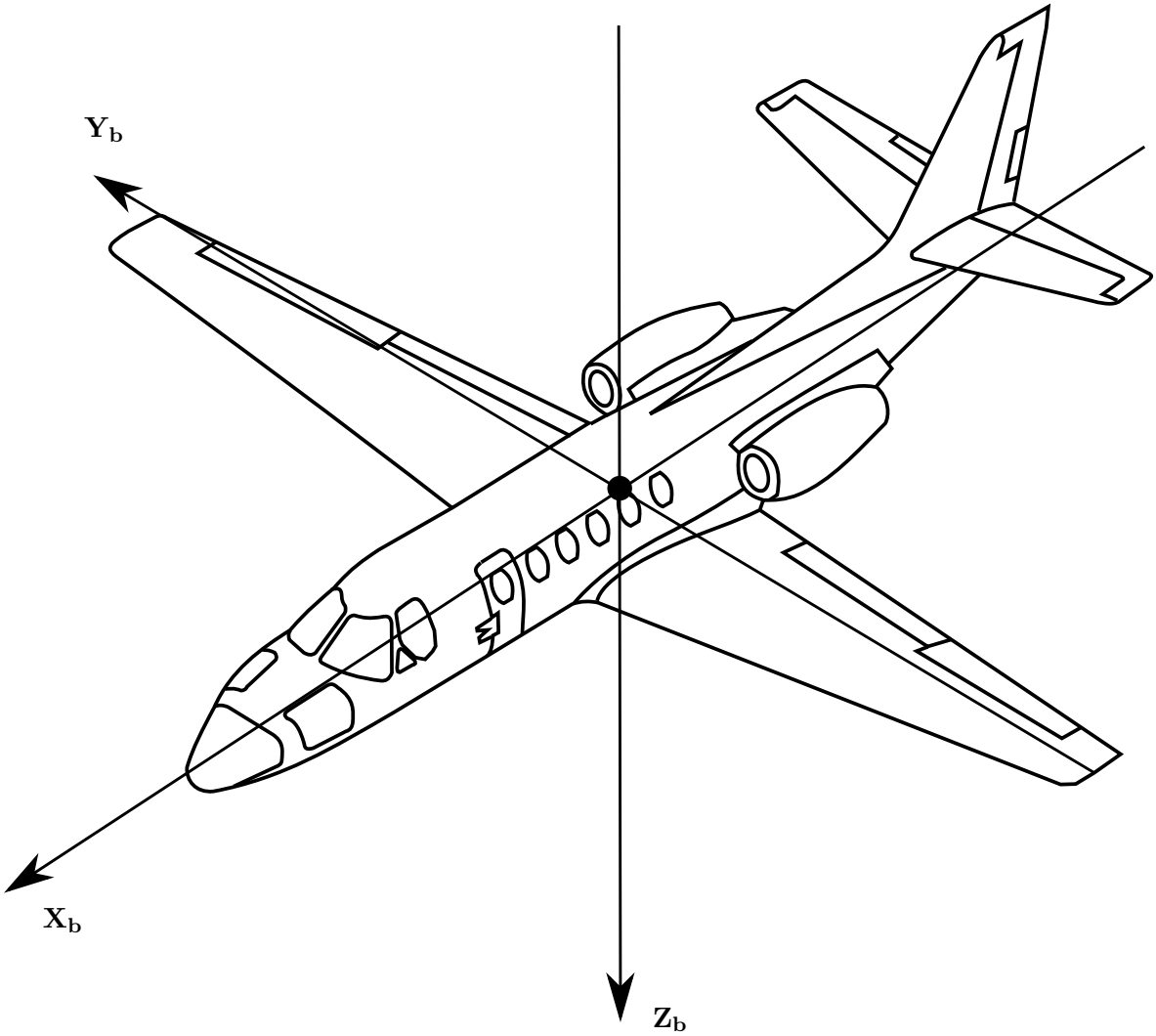


Figure F-1: Aircraft body-fixed reference frame $\mathbb{F}_b$

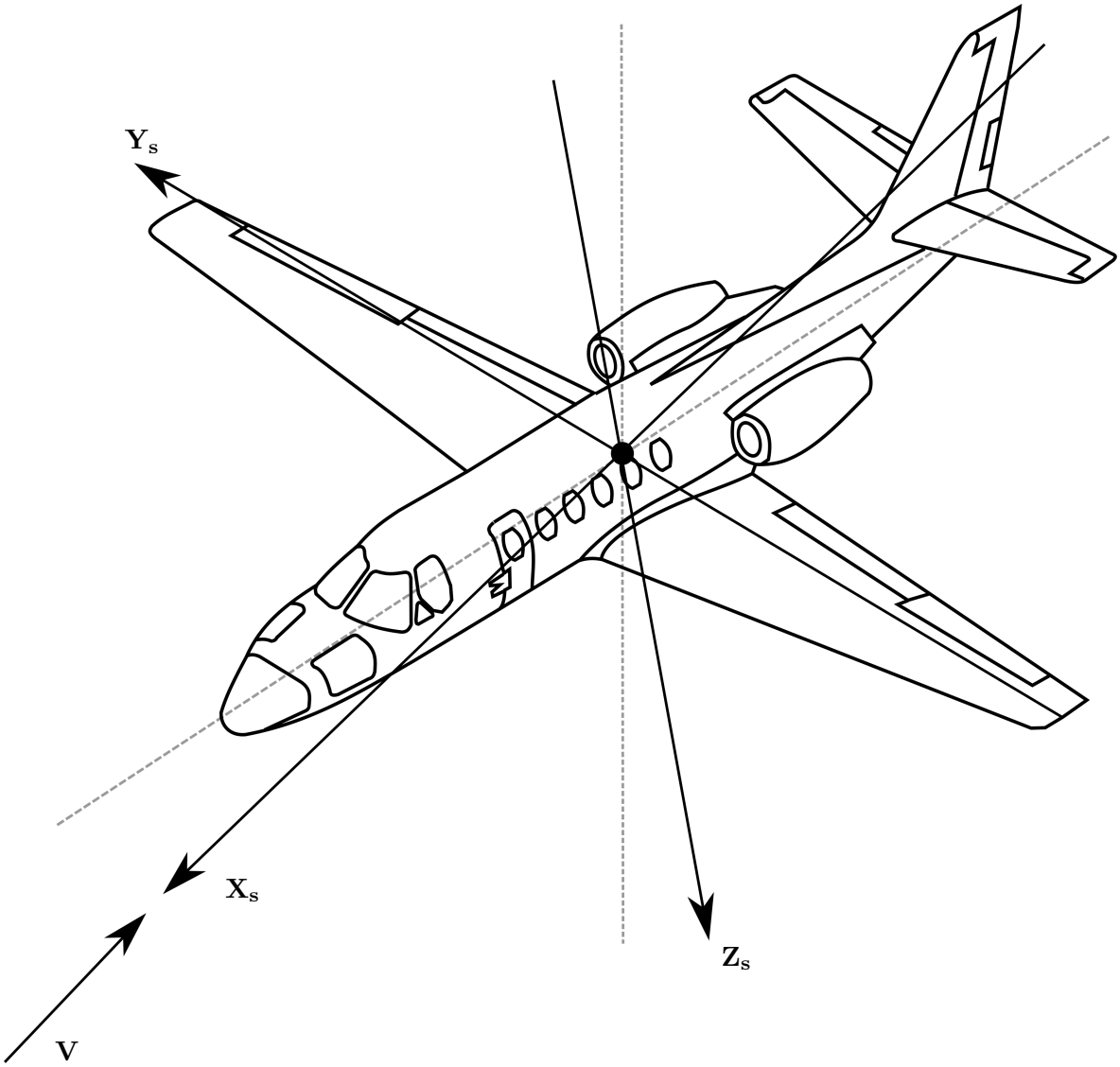


Figure F-2: Aircraft stability reference frame $\mathbb{F}_s$

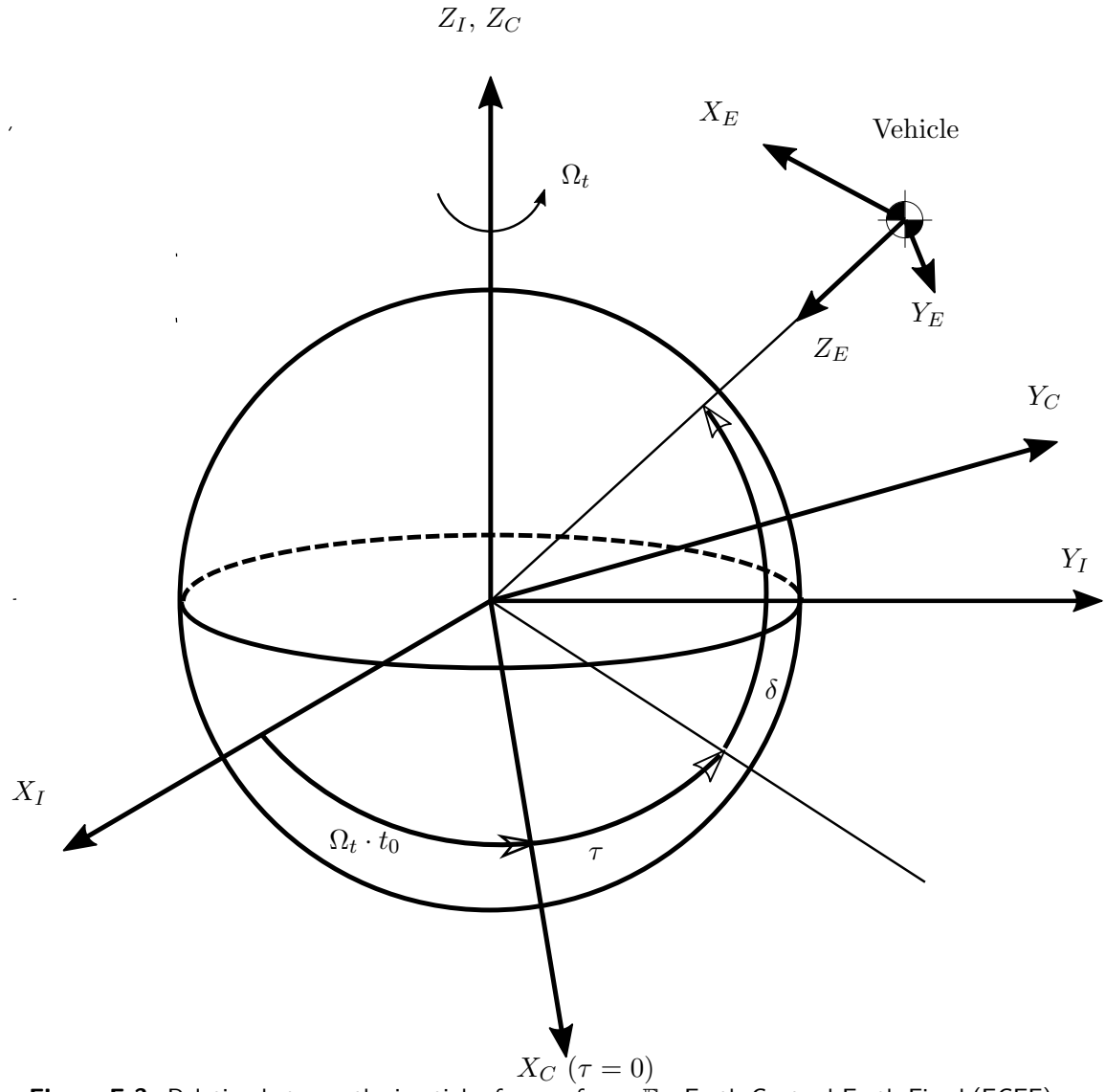


Figure F-3: Relation between the inertial reference frame $\mathbb{F}_I$, Earth-Centred-Earth-Fixed (ECEF) reference frame $\mathbb{F}_C$ and the vehicle-carried normal Earth reference frame $\mathbb{F}_E$

Appendix G

Flight Envelope

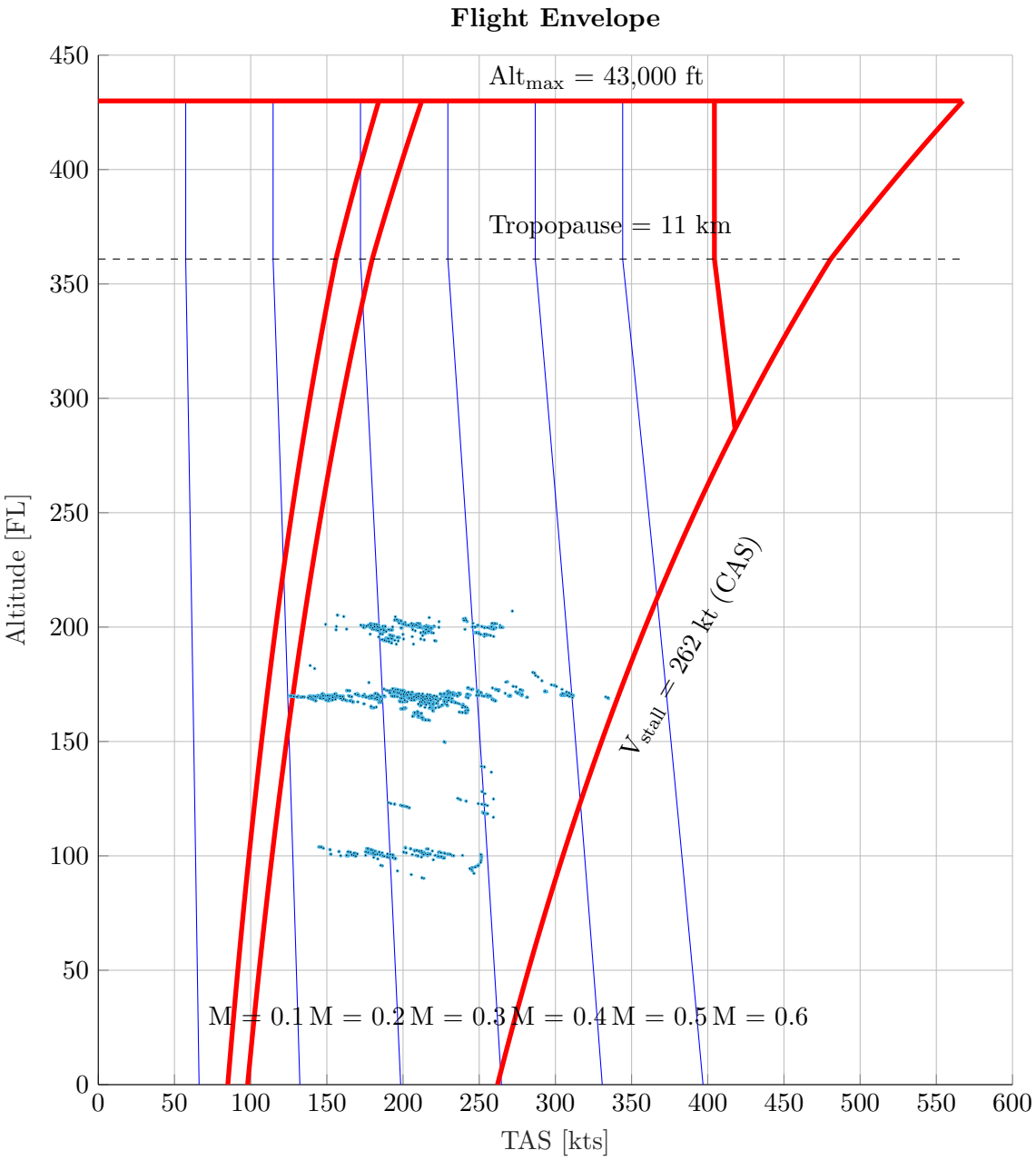


Figure G-1: The Flight Envelope of the PH-LAB Cessna Citation II Aircraft. The blue dots indicate the measured flight test data.

Appendix H

Work Flow Block Diagram

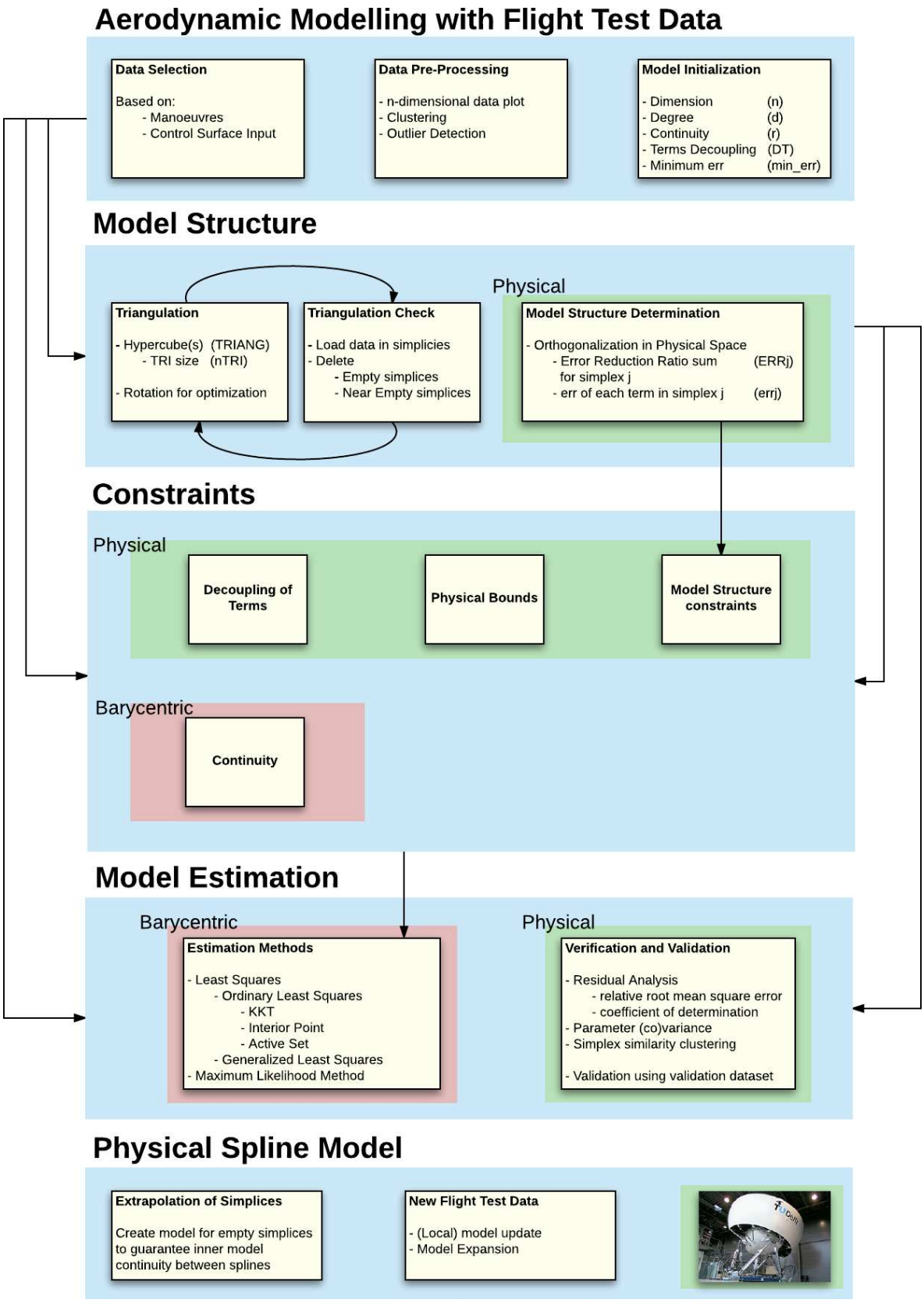


Figure H-1: Block diagram indicating the work flow for aerodynamic modelling with Multivariate Physical Simplex B-Splines

Appendix I

Matlab Code Physical Transformation based on convolution

```
function [Lambda, iLambda] = LambdaF(TRIANG,d)

n = size(TRIANG.PHI,2);
T = size(TRIANG.TRI,1);

% Calculate polynomial basis function multi-index permutations
[pbasis, basisKt] = bsplinen_constructBasis(n, d);
dhat = size(pbasis, 1);

Lambda = zeros(dhat*T,dhat*T);
iLambda = zeros(dhat*T,dhat*T);

% Gamma Indexer
g = (2*d+2).^((1:n)-1)+1;
g = [1,g];

% Indexer reversing Kronecker Substitution
kp = pbasis(:,2:end);
ig = zeros(dhat,1);
for i = 1:size(kp,1)
    ig(i) = ig(i) + sum(kp(i,:).*(g(2:end)-1)) + 1;
end

for t = 1:T
    % Create Cartesian to Barycentric transformation matrix
    % such that b = A*x + k
    simplex = TRIANG.PHI(TRIANG.TRI(t, :), :);
    v0      = (simplex(1, :))';

    [~, An] = bsplinen_cart2bary(simplex, ones(1, n));

    A0 = -ones(1, size(An, 1)) * An;

```

```
34      Kn = -An*v0;
35      K0 = 1 - ones(1, size(Kn, 1)) * Kn;
36
37      Axb = [A0; An];
38      kxb = [K0; Kn];
39
40      % Create Kronecker substituted cells representing the barycentric
            variables
41      Ag = cell(n+1,1);
42      for i = 1:(n+1)
43          Ag{i} = zeros(2*((2*d+2)^(n-1)+1),1);
44          Ag{i}(g) = [kxb(i),Axb(i,:)];
45      end
46
47      % Perform convolution for each column of Lambda using pbasis
48      Lambda_t = zeros(dhat,dhat);
49      for i = 1:dhat
50          Lambda_tc = [];
51          % The next two for-loops have d iterations, i.e. |pbasis| = d
52          for j = 1:(n+1)
53              for k = 1:pbasis(i,j)
54                  if ~isempty(Lambda_tc)
55                      Lambda_tc = conv(Lambda_tc,Ag{j});
56                  else
57                      Lambda_tc = Ag{j};
58                  end
59              end
60          end
61          % Undo Kronecker Substitution and write to Lambda_t
62          % and multiply with multivariate coefficient
63          Lambda_t(:,i) = basisKt(i).*Lambda_tc(ig);
64      end
65
66      % Write to global transformation matrix and determine the inverse
67      Lambda(((t-1)*dhat+1):(t*dhat),((t-1)*dhat+1):(t*dhat)) = Lambda_t;
68      iLambda(((t-1)*dhat+1):(t*dhat),((t-1)*dhat+1):(t*dhat)) = Lambda_t\eye(
            dhat);
69
70  end
71
72  Lambda = sparse(Lambda);
73  iLambda = sparse(iLambda);
74
75  end
```

Appendix J

Active-Set Method

Another method to solve a minimization problem subjected to inequality problems and equality problems is the active-set method. Recalling the minimization problem used to solve for the B-coefficients

$$\min_{c} \frac{1}{2} \parallel \mathbf{B} \cdot \mathbf{c} - \mathbf{Y} \parallel \tag{J-1}$$

subjected to

$$\mathbf{H} \cdot \mathbf{c} = \mathbf{0} \tag{J-2}$$

$$\mathbf{\Lambda} \cdot \mathbf{G} \cdot \mathbf{c} = \mathbf{G_B} = \mathbf{0} \tag{J-3}$$

$$\mathbf{\Lambda} \cdot \mathbf{F} \cdot \mathbf{c} \geq \mathbf{v} \tag{J-4}$$

There are different ways how to use the active-set method. Assumed is that the inequality constraints, regarding the aerodynamic coefficient, are most of the time met without constraints. An efficient way of solving and implementing the active-set method is described by (Kuindersma et al., 2014) and (Harkegard, 2002). This algorithm verifies first if the inequality constraints are violated. For the first iteration, $k = 0$, setup the KKT matrix with only the equality constraints included

$$\begin{bmatrix} \vec{c_{B_{k=0}}} \\ \vec{\lambda_{k=0}} \end{bmatrix} = \begin{bmatrix} \vec{B^T} B & H\vec{G}_B^T \\ H\vec{G}_B & \vec{0} \end{bmatrix}^{-1} \begin{bmatrix} \vec{B^T} Y \\ \vec{0} \end{bmatrix} \tag{J-5}$$

next check if any inequality constraints are violated by solving for c_{B_k} and check if

$$\mathbf{\Lambda} \cdot \mathbf{F} \cdot \mathbf{c_{B_{k=0}}} - \mathbf{v} \geq \mathbf{0} \tag{J-6}$$

The inequality constraints that are violated are added to the next active-set, $k = 1$, as equality constraints. This process is repeated until none of the inequality constraints are violated anymore. This process is explained in the pseudo code below

Data: When c_{B_k}, containing only equality constraints, violates inequality constraints use active-set method to solve for the minimization problem.
Result: An optimal solution for c_B when subjected to constraints
while c_{B_k} *violates* **do**
  Solve for c_{B_k} using KKT and active-set HG_bF_j;
  if $\Lambda \cdot F \cdot c_{B_k} - v \geq 0$ **then**
    add violating constraints j to active-set HG_bF_j;
  else
    if c_{B_k} *all satisfy* **then**
      break;
    end
  end
end

Algorithm 1: Active-Set Algorithm

The active-set method is successfully implemented in Matlab and can be selected within the framework. An advantage of this algorithm is that it transforms into an "ordinary" KKT matrix when there are no inequality constraints. Assumed is that on forehand not many of the inequality constraints are violated, making it a fast method since less iterations are required.

J-1 Matlab Code ExampleActive-Set Method

```
function coefs = qpActiveSet(X,Y,A,b,Aeq,beq,iter)
    f = -X'*Y;

    ciq = false(size(A,1),1); % Check IneQualities
    ciqi = ciq; % Check IneQualities Index
    ceq = true(size(Q,1)+size(Aeq,1),1); % Check EQualities
    stopOpt = false; % Stop optimization
    KKT = [Q Aeq' A';
           [Aeq; A] zeros(size(Aeq,1)+size(A,1))];
    BT = [-f; beq; b];
    ep = 1*10^-6;
    k = 100; % Number of iterations
    Xm3 = zeros(size(BT)); % Solution

    while ~stopOpt
        if iter == 1 % Use iterative solver
            la = ones(size(Aeq,1)+size(A(ciq,:),1),1);
            Xm3 = (2*Q + (1/ep)*[Aeq;A(ciq,:)]'*[Aeq;A(ciq,:)]) \ ...
                (2*X'*Y - [Aeq;A(ciq,:)]'*la + (1/ep)*[Aeq;A(ciq,:)]'*[beq;b(
                    ciq,:)]);
            for i = 1:k
                Xm3 = (2*Q + (1/ep)*[Aeq;A(ciq,:)]'*[Aeq;A(ciq,:)]) \ ...
                    (2*Q*Xm3(1:size(Q,1)) + (1/ep)*[Aeq;A(ciq,:)]'*[beq;b(ciq
                        ,:)]);
            end
        else
            Xm3 = KKT([ceq; ciq],[ceq; ciq])\BT([ceq; ciq]);
```

```
26          end
27          % Check for violating constraints
28          ciq(~ciq,:) = A(~ciq,:)*Xm3(1:size(Q,1))-b(~ciq,:)>0;
29          if sum(ciq(~ciqi,:)) <= 0
30              stopOpt = true;
31          else
32              ciqi = ciq;
33          end
34      end
35      coefs = Xm3(1:size(Q,1));
36  end
```

Appendix K

Interior Point Method

The interior point method, sometimes called the barrier method, can be used to solve convex optimization problems, both linear and non-linear subjected to constraints. To prevent violation of the inequality constraints a barrier term and Lagrangian multipliers λ for equality constraints are added to the objective function J. It is necessary to rewrite/ augment the constraints to standard form such that they are equal to 0 or ≥ 0. By introducing slack variables z, the inequality constraints are transformed into equality constrains $\vec{S}$ and the slack variables form the inequality constraints. μ is a variable becoming each iteration k smaller and approaching 0 when $k \to \infty$ forming the barrier. The interior method introduced is described by (Robere, 2012). First create the following matrix

$$\begin{bmatrix} \vec{W}_k & \nabla_c \vec{S}(c_k) & -\vec{I} \\ \nabla_c \vec{S}(c_k)^\top & \vec{0} & \vec{0} \\ \vec{Z}_k & \vec{0} & \vec{\lambda_k} \end{bmatrix} \begin{bmatrix} \vec{d_k^c} \\ \vec{d_k^\lambda} \\ \vec{d_k^z} \end{bmatrix} = - \begin{bmatrix} \nabla \vec{J}(c_k) + \nabla \vec{S}(c_k) \vec{\lambda}_k \\ \vec{S}(c_k) \\ \lambda_k Z_k \vec{e} - \mu_k e \end{bmatrix} \tag{K-1}$$

where,

$$\vec{W}_k = \nabla^2_{cc} \vec{L} \left(\vec{c}_k, \vec{\lambda}_k, \vec{z}_k \right) = \nabla^2_{cc} \left(\vec{J}(c_k) + \vec{S}(c_k) - \vec{z}_k \right) \tag{K-2}$$

and the Lagrangian with the multipliers and the slack variables is given by

$$L(\mathbf{c}, \lambda_\mathbf{h}, \lambda_\mathbf{m}, \mu_\mathbf{n}) = \frac{1}{2}(\mathbf{Y} - \mathbf{B} \cdot \mathbf{c})^\top (\mathbf{Y} - \mathbf{B} \cdot \mathbf{c}) + \lambda_\mathbf{h}^\top \cdot \mathbf{H} \cdot \mathbf{c} + \lambda_\mathbf{m}^\top \cdot (\mathbf{M} \cdot \mathbf{c} - \mathbf{v}) + \mu_\mathbf{n}^\top \cdot (\mathbf{N} \cdot \mathbf{c} - \mathbf{z}) \tag{K-3}$$

$$\vec{\lambda_k} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix}, \ \vec{Z_k} = \begin{bmatrix} z_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & z_n \end{bmatrix} \tag{K-4}$$

The objective function is minimized for c iteratively for $k = 0, 1, \ldots l$ After each iteration c, the Lagrangian multiplier and the slack variables are updated according to

$$\vec{c}_{k+1} = \vec{c}_k + \alpha \, \vec{d_k^x} \tag{K-5}$$

$$\vec{\lambda}_{k+1} = \vec{c}_k + \alpha \, \vec{d_k^\lambda} \tag{K-6}$$

$$\vec{z}_{k+1} = \vec{z}_k + \alpha \, \vec{d}_k^z \tag{K-7}$$

Each iteration μ and step size α are updated. This, however, is a non-trivial process. A method that can be used is the predictor-corrector technique.

Before the iteration process can start c has to be initialized first. Initialize $\mathbf{c_0}$, such that $\mathbf{c_0}$ = feasible. An option is to initialize at zero, i.e. $c_0 \vec{=} 0$, $\lambda_0 \vec{=} 0$ and $z_0 \vec{=} 0$. If one decides not to initialize at zero, $\lambda_\mathbf{0}$ and $\mathbf{z_0}$ can be calculated with

$$\begin{bmatrix} \vec{I} & \nabla_c \vec{S}(c_0) \\ \nabla_c \vec{S}(c_0)^\top & \vec{0} \end{bmatrix} \begin{bmatrix} \vec{w} \\ \vec{\lambda}_0 \end{bmatrix} = - \begin{bmatrix} \nabla \vec{J}(c_0) - \vec{z}_{L,0} - \vec{z}_{u,0} \\ \vec{0} \end{bmatrix} \tag{K-8}$$

$$\mathbf{z_0} = \frac{\mu}{\mathbf{x_o}} \tag{K-9}$$

An option is to rewrite equation K-1 into a linear symmetric system

$$\begin{bmatrix} \vec{W}_k + \vec{\Sigma}_k & \nabla_c \vec{S}(c_0) \\ \nabla_c \vec{S}(c_0)^\top & \vec{0} \end{bmatrix} \begin{bmatrix} \vec{d}_k^c \\ \vec{d}_k^\lambda \end{bmatrix} = - \begin{bmatrix} \nabla \vec{J}(c_0) + \nabla \vec{S}(c_k) \vec{\lambda}_k \\ \vec{S}(c_k) \end{bmatrix} \tag{K-10}$$

where

$$\vec{\Sigma}_k = \vec{X}_k^{-1} \vec{Z}_k \tag{K-11}$$

After solving for $\vec{d}_k^x$, $\vec{d}_k^z$ is calculated with

$$\vec{d}_k^z = \mu \vec{S}_k^{-1} \vec{e} - \vec{z}_k - \vec{\Sigma}_k \vec{d}_k^c \text{ where } \vec{e} = [1, \ldots, 1]^\top \tag{K-12}$$

Next c, the Lagrangian multipliers and the slack variables are updated using the same equations (K-5), (K-6) and (K-7).

K-1 Interior Point Method for Multivariate B Simplex Splines

In this section the interior point method is used to solve for the B-coefficients. The following multivariate B simplex spline function is minimized, where B is the B-matrix, c are the B-coefficients and Y is the output data

$$\min_c \frac{1}{2} \parallel \mathbf{B} \cdot \mathbf{c} - \mathbf{Y} \parallel \tag{K-13}$$

The spline function is subjected to equality and inequality constraints, where $\vec{H}$ are the equality constraints determined by the continuity/ smoothness, $\Lambda \vec{\cdot} G$ the hard equality constraints on the parameters in physical space transformed to Barycentric space and $\Lambda \vec{\cdot} F$ the inequality constraints on the parameters in physical space transformed to Barycentric space as well. Λ is transformation matrix between physical space and Barycentric space

$$\mathbf{H} \cdot \mathbf{c} = \mathbf{0} \tag{K-14}$$

$$\mathbf{\Lambda} \cdot \mathbf{G} \cdot \mathbf{c} = \mathbf{0} \tag{K-15}$$

$$\mathbf{\Lambda} \cdot \mathbf{F} \cdot \mathbf{c} \geq \mathbf{v} \tag{K-16}$$

The constraints are rewritten/ augmented to standard form and slack variables $\vec{c_s}$ are introduced next to the $\vec{c}$-coefficients in order to rewrite the inequality constraints to equality constraints. Combined they form the new $\vec{c_r}$ vector.

$$\mathbf{H_r} \cdot \mathbf{c_r} = \mathbf{0} \tag{K-17}$$

$$\mathbf{\Lambda} \cdot \mathbf{G_r} \cdot \mathbf{c_r} = \mathbf{0} \tag{K-18}$$

$$\mathbf{\Lambda} \cdot \mathbf{F_r} \cdot \mathbf{c_r} - \mathbf{v} \ = \mathbf{0} \tag{K-19}$$

$$\mathbf{c_s} \geq \mathbf{0} \tag{K-20}$$

The size of other matrices are changed according to the number of slack variables. Creating new matrices and vectors with subscript $_r$

$$\vec{H_r} = \begin{bmatrix} \vec{H} & \vec{0} \end{bmatrix},\ \vec{G} = \begin{bmatrix} \vec{G} & \vec{0} \end{bmatrix},\ \vec{F_r} = \begin{bmatrix} \vec{F} & -\vec{I} \end{bmatrix},\ \vec{c_r} = \begin{bmatrix} \vec{c} \\ \vec{c_s} \end{bmatrix} \tag{K-21}$$

The slack variables are introduced in the cost function (K-22)

$$\vec{J}(c) = \frac{1}{2} \parallel \mathbf{B_r} \cdot \mathbf{c_r} - \mathbf{Y} \parallel = \frac{1}{2} \left(Y - \vec{B}_r \cdot c \right)^\top \left(Y - \vec{B}_r \cdot c_r \right) \tag{K-22}$$

where

$$\vec{B_r} = \begin{bmatrix} \vec{B} & \vec{0} \end{bmatrix} \tag{K-23}$$

The first derivative of the cost function is taken with respect to c

$$\nabla_c \vec{J}(c) = -B^\top \cdot Y + \vec{B^\top} \cdot B \cdot c \tag{K-24}$$

and the second derivative with respect to c.

$$\nabla^2_{cc} \vec{J}(c) = \vec{B^\top} \cdot B \tag{K-25}$$

Note that the second derivative is constant for our minimization problem.

$$\begin{bmatrix} \vec{B_s^\top} B_s + \vec{\Sigma}_k & H\vec{G}F_s \\ H\vec{G}F_s^\top & \vec{0} \end{bmatrix} \begin{bmatrix} \vec{d_k^c} \\ \vec{d_k^\lambda} \end{bmatrix} = - \begin{bmatrix} -B_s^\top Y + B_s^\top \vec{B_s} c_s + HGF_s \lambda_k \\ \vec{0} \end{bmatrix} \tag{K-26}$$

After each iteration the coefficients, the Lagrangian multipliers and the slack variables are updated using equations (K-5), (K-6) and (K-7). An algorithm of the interior point method described above is created in Matlab and successfully applied during regression.

Appendix L

Results Aerodynamic Model C_m Detailed

Table L-1: Simplex information and statistical analysis, $C_m(\alpha,\ q,\ M,\ \delta_e) \in S_1^0(T_{1111=24})$

Simplex	Data Points	ERR	$\sigma(\theta)_{min}$	$\sigma(\theta)_{max}$	Condition Number
1	737	0.77	8.31×10^{-06}	7.88×10^{-03}	13
2	11420	0.69	7.28×10^{-07}	7.88×10^{-03}	16
3	289	0.95	1.10×10^{-05}	1.53	60
4	12	0.99	1.10×10^{-05}	6.77×10^{-03}	52
5	3085	0.87	8.04×10^{-07}	6.17×10^{-03}	9
6	202	0.97	2.99×10^{-06}	2.68×10^{-03}	15
7	1030	0.95	8.26×10^{-06}	4.77×10^{-03}	8
8	5479	0.94	6.61×10^{-07}	4.77×10^{-03}	12
9	894	0.97	5.02×10^{-06}	4.77×10^{-03}	12
10	3674	0.98	1.93×10^{-06}	4.77×10^{-03}	15
11	2432	0.92	6.25×10^{-07}	4.77×10^{-03}	19
12	2800	0.94	1.93×10^{-06}	4.77×10^{-03}	17
13	55	0.98	0	1.53	78.38
14	5	1.00	0	4.67×10^{-03}	49723
15	829	0.96	0	7.84×10^{-03}	16
16	3271	0.96	0	7.84×10^{-03}	15.47
17	424	0.98	0	2.68×10^{-03}	14
18	20876	0.81	0	2.97×10^{-03}	24
19	3266	0.93	6.63×10^{-07}	6.17×10^{-03}	8
20	886	0.92	1.71×10^{-06}	2.68×10^{-03}	6
21	4100	0.88	4.49×10^{-07}	8.71×10^{-03}	16
22	10678	0.87	1.93×10^{-06}	8.71×10^{-03}	26
23	1206	0.93	6.03×10^{-07}	2.68×10^{-03}	8
24	16254	0.85	1.16×10^{-06}	2.97×10^{-03}	18

Table L-2: Parameter coefficients of model $C_m(\alpha,\, q,\, M,\, \delta_e) \in S_1^0(T_{1111=24})$

Simplex	$\mathbf{C_{m_0}}$	$\mathbf{C_{m_\alpha}}$	$\mathbf{C_{m_q}}$	$\mathbf{C_{m_M}}$	$\mathbf{C_{m_\delta}}$
1	-0.1	-0.44	-10.29	0.36	-1.22
2	-0.04	-0.44	-10.29	0.18	-1.07
3	-0.81	-0.44	-43.77	2.65	-1.22
4	-0.87	-0.44	-10.00	2.65	-3.13
5	-0.04	-0.44	-9.12	0.18	-1.13
6	-0.06	-0.44	-10.00	0.24	-1.13
7	-0.1	-0.50	-9.14	0.36	-1.22
8	-0.04	-0.50	-9.14	0.18	-1.07
9	-0.04	-0.31	-9.14	0.12	-1.22
10	-0.04	-0.36	-9.14	0.12	-1.17
11	-0.04	-0.51	-9.14	0.18	-1.06
12	0.01	-0.36	-9.14	-0.01	-1.06
13	-0.26	1.43	-43.77	0.21	-1.22
14	-0.32	1.43	-10.00	0.21	-3.13
15	-0.07	-0.31	-10.42	0.21	-1.22
16	-0.06	-0.36	-10.42	0.21	-1.17
17	-0.04	-0.51	-10.00	0.21	-1.03
18	-0.06	-0.36	-12.89	0.21	-1.03
19	-0.03	-0.53	-9.12	0.18	-1.03
20	-0.05	-0.53	-10.00	0.24	-1.03
21	-0.03	-0.51	-9.62	0.18	-1.03
22	0.01	-0.36	-9.62	-0.01	-1.03
23	-0.04	-0.51	-10.00	0.21	-1.03
24	-0.06	-0.36	-12.89	0.21	-1.03

Table L-3: Parameter coefficients of model $C_m(\alpha,\ q,\ M,\ \delta_e) \in S_1^0(T_{1111=24})$ with optimal rotation

Simplex	$\mathbf{C_{m_0}}$	$\mathbf{C_{m_\alpha}}$	$\mathbf{C_{m_q}}$	$\mathbf{C_{m_M}}$	$\mathbf{C_{m_\delta}}$
1	-0.15	0.29	-24.16	0.26	-1.07
2	-0.14	0.29	-24.16	0.22	-1.1
3	-0.17	0.29	-10.63	0.26	-1.84
4	-0.2	0.29	-9.13	0.36	-1.84
5	-0.48	0.29	-7.63	1.35	-1.1
6	-0.47	0.29	-9.13	1.35	-1.02
7	-0.07	-0.41	-10.63	0.26	-1.07
8	-0.06	-0.41	-10.63	0.22	-1.1
9	-0.09	-0.31	-10.63	0.26	-1.18
10	-0.03	-0.43	-10.63	0.11	-1.18
11	0.02	-0.58	-10.63	0.01	-1.1
12	0	-0.43	-10.63	0.01	-1.26
13	-0.07	-0.42	-10.63	0.26	-1.06
14	-0.1	-0.42	-9.13	0.36	-1.06
15	-0.08	-0.31	-12.72	0.26	-1.06
16	-0.02	-0.43	-12.72	0.11	-1.06
17	-0.02	-0.58	-9.13	0.16	-1.06
18	-0.04	-0.43	-11.92	0.16	-1.06
19	-0.01	-0.64	-7.63	0.13	-1.1
20	0	-0.64	-9.13	0.13	-1.02
21	-0.01	-0.58	-8.92	0.13	-1.1
22	-0.03	-0.43	-8.92	0.13	-1.26
23	-0.01	-0.58	-9.13	0.13	-1.09
24	-0.03	-0.43	-11.92	0.13	-1.09

Bibliography

Aitken, A. C. (1936). Iv.on least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, *55*, 42–48.

Awanou, G., & Lai, M. (2005). On convergence rate of the augmented lagrangian algorithm for nonsymmetric saddle point problems. *Applied Numerical Mathematics*, *54*(2), 122–134.

Awanou, G., Lai, M.-J., & Wenston, P. (2005). The multivariate spline method for scattered data fitting and numerical solutions of partial differential equations. *Wavelets and splines: Athens*, 24–74.

Bava, R., Hoare, G. T., Garcia-Mesuro, G., & Oelker, H.-C. (1999). Recent experiences on aerodynamic parameter identification for eurofighter at alenia, british aerospace, casa and daimier-benz aerospace. In *Nato rto meeting proceedings* (Vol. 11).

Billings, S., Korenberg, M., & Chen, S. (1988). Identification of non-linear output-affine systems using an orthogonal least-squares algorithm. *International Journal of Systems Science*, *19*(8), 1559–1568.

Billings, S., & Voon, W. (1986). A prediction-error and stepwise-regression estimation algorithm for non-linear systems. *International Journal of Control*, *44*(3), 803–822.

Boor, C. de. (1972). On calculating with B-splines. *Journal of Approximation Theory*, *6*(1), 50–62.

Boor, C. de. (1987). B-form basics. *Geometric Modeling: Algorithms and New Trends*, 131–148.

Boor, C. D. (1976). Splines as linear combinations of B-splines. A Survey. *Approximation Theory*, 1–47.

Brandon, J. M. (1998). Fuzzy logic modeling of nonlinear unsteady aerodynamics. , 565–580.

Brandon, J. M., & Morelli, E. A. (2016, feb). Real-Time Onboard Global Nonlinear Aerodynamic Modeling from Flight Data. *Journal of Aircraft*, *53*(5), 1261–1297. Available from `http://dx.doi.org/10.2514/1.C033133`

Cox, M. G. (1975). An algorithm for spline interpolation. *IMA Journal of Applied Mathematics (Institute of Mathematics and Its Applications)*, *15*(1), 95–108.

Curry, H. B., & Schoenberg, I. J. (1966). On Pólya frequency functions IV: The fundamental spline functions and their limits. *Journal d'Analyse Mathématique*, *17*(1), 71–107.

Da Ronch, A., Ghoreyshi, M., & Badcock, K. (2011). On the generation of flight dynamics aerodynamic tables by computational fluid dynamics. *Progress in Aerospace Sciences*, *47*(8), 597–620.

De Visser, C. C. (2011). *Global nonlinear model identification with multivariate splines.* Unpublished doctoral dissertation, Delft University of Technology.

Federal Aviation Administration. (n.d.). *Qualification, Service, and Use of Crewmembers and Aircraft Dispatchers* (Tech. Rep.). U.S. Department of Transportation.

Ferranti & Christoph Hormann, J. de. (n.d.). *Online dem data.* `http://viewfinderpanoramas.org/dem3.html`. (Accessed: 2017-01-25)

González-Vega, L., & Trujillo, G. (1997). Multivariate sturm-habicht sequences: real root counting on n-rectangles and triangles. *Revista Matemática Complutense*, *10*, 119–130.

Gould, N. I. M., & Hribar, M. E. (n.d.). On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM J. Sci. Comput*(4), 1376–1395.

Greenberg, H. (n.d.). A survey of methods for determining stability parameters of an airplane from dynamic flight measurements. (April).

Harkegard, O. (2002). Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. In *Decision and control, 2002, proceedings of the 41st ieee conference on* (Vol. 2, pp. 1295–1300).

Hoek, M. A. van den, Visser, C. C. de, & Pool, D. M. (2017). Identification of a Cessna Citation II Model Based on Flight Test Data. In *Proceedings of the 4th ceas specialist conference on guidance, navigation & control, warsaw, poland.*

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, *31*(8), 651–666.

Jategaonkar, R. (2006). *Flight vehicle system identification: a time domain methodology* (Vol. 216). AIAA, Reston, VA, USA.

Kirkpatrick, K., May Jr, J., & Valasek, J. (2013). Aircraft system identification using artificial neural networks. In *51st aiaa aerospace sciences meeting including the new horizons forum and aerospace exposition, grapevine, tx.*

Klein, V., & Morelli, E. A. (2006). *Aircraft system identification: theory and practice.* American Institute of Aeronautics and Astronautics Reston, Va, USA.

Kuhn, H. W. (1960). Some combinatorial lemmas in topology. *IBM Journal of research and development*, *4*(5), 518–524.

Kuhn, H. W., & Tucker, A. (1951). Nonlinear Programming. *Proceedings of the Second Symposium on Mathematical Statistics and Probability*(x), 481–492. Available from `http://projecteuclid.org/euclid.bsmsp/1200500249`

Kuindersma, S., Permenter, F., & Tedrake, R. (2014). An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *2014 ieee international conference on robotics and automation (icra)* (pp. 2589–2594).

Kyriakides, E., & Heydt, G. T. (2006). Calculating confidence intervals in parameter estimation: a case study. *IEEE transactions on power delivery*, *21*(1), 508–509.

Lai, M.-J., & Schumaker, L. L. (2007). *Spline functions on triangulations* (No. 110). Cambridge University Press.

Lee, T., & Kim, Y. (2001). Nonlinear adaptive flight control using backstepping and neural networks controller. *Journal of Guidance, Control, and Dynamics*, *24*(4), 675–682.

Linden, C. Van der. (1998). Dasmat-delft university aircraft simulation model and analysis tool: A matlab/simulink environment for flight dynamics and control analysis. *Series 03: Control and Simulation 03*.

Lombaerts, T., Chu, Q., Mulder, J., & Joosten, D. (2007). Real time damaged aircraft model identification for reconfiguring flight control. In *Aiaa atmospheric flight mechanics conference and exhibit* (p. 6717).

Maddala, G. S. (1971). Generalized least squares with an estimated variance covariance matrix. *Econometrica: Journal of the Econometric Society*, 23–33.

Monteiro, R. D., Adler, I., & Resende, M. G. (1990). A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Mathematics of Operations Research*, *15*(2), 191–214.

Morelli, E. A. (1998). Global nonlinear parametric modelling with application to f-16 aerodynamics. In *American control conference, 1998. proceedings of the 1998* (Vol. 2, pp. 997–1001).

Mulder, e. a., J. A. (2000). *Flight Dynamics (Lecture Notes): AE3-302*. Delft: TU Delft.

Mulder, J. A. (1986). *Design and evaluation of dynamic flight test manoeuvres*. Unpublished doctoral dissertation, Technische Universiteit Delft.

Müller, B. (2006). Wake penetration a tumultuous farewell of 1st typhoon prototype aircraft.

Obrien, R. M. (2007). A caution regarding rules of thumb for variance inflation factors. *Quality & Quantity*, *41*(5), 673–690.

Pedersen, P. (1991). Multivariate sturm theory. In *International symposium on applied algebra, algebraic algorithms, and error-correcting codes* (pp. 318–332).

Rai, M. M., & Madavan, N. K. (2000). Aerodynamic design using neural networks. *AIAA journal*, *38*(1), 173–182.

Robere, R. (2012). Interior point methods and linear programming. *University of Toronto*.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Rump, S. M. (1979). Polynomial minimum root separation. *Mathematics of Computation*, *33*(145), 327–336.

Schoenberg, I. J. (1946). Contributions to the problem of approximation of equidistant data by analytic functions. part a. on the problem of smoothing or graduation. a first class of analytic approximation formulae. *Quarterly of Applied Mathematics*, *4*(1), 45–99.

Shahbandi, S. G., & Lucidarme, P. (n.d.). Object recognition based on radial basis function neural networks: Experiments with RGB-D camera embedded on mobile robots. *Systems and Computer Science (ICSCS), 2012 1st International Conference on*, 1–6.

Shinbrot, M. (1951). A Least square Curve Fitting Method with Applications to the Calculation of Stability Coefficients from Transient Response Data. *NACA TN 2341*.

Stine, R. A. (1995). Graphical interpretation of variance inflation factors. *The American Statistician*, *49*(1), 53–56.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *Systems, Man and Cybernetics, IEEE Transactions on*, *SMC-15*(1), 116–132.

Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*. SIAM.

Tol, H. J., Visser, C. C. D., Sun, L. G., Kampen, E. V., & Chu, Q. P. (2016). Multivariate Spline-Based Adaptive Control of High-Performance Aircraft with Aerodynamic Uncertainties. *Journal of Guidance, Control, and Dynamics*, *39*(4), 781–800.

Vanderbei, R. J., & Lagarias, J. C. (1990). I. I. Dikin's convergence result for the affine-scaling algorithm. *Contemporary Mathematics*, *114*, 109–119.

Visser, C. de, Mulder, J., & Chu, Q. (2009). Global nonlinear aerodynamic model iden-

tification with multivariate splines. In *Aiaa atmospheric flight mechanics conference* (p. 5726).

Visser, C. de, Mulder, J., & Chu, Q. (2010). A Multidimensional Spline-Based Global Nonlinear Aerodynamic Model for the Cessna Citation II. *AIAA Atmospheric Flight Mechanics Conference*(August), 1–30. Available from `http://arc.aiaa.org/doi/abs/10.2514/6.2010-7950`

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

Visser, C. C. de, Chu, Q., & Mulder, J. (2011). Differential constraints for bounded recursive identification with multivariate splines. *Automatica*, *47*(9), 2059–2066.

Visser, C. C. de, Kampen, E. van, Chu, Q. P., & Mulder, J. A. (n.d.). Intersplines: A new approach to globally optimal multivariate splines using interval analysis. *Reliable Computing*(December 2012), 153–191.

Yu, H., M., B., & Wilamowski. (2011). LevenbergMarquardt Training. *The Industrial Electronics Handbook*, 12–1–12–16.