Knowledge Based Engineering framework for preliminary spaceplane design

Master of Science Thesis
Aerospace Engineering

Tomasz Sadowy-Sadowski

Delft University of Technology

TU Delft

Knowledge Based Engineering framework for preliminary spaceplane design

Master of Science Thesis
Aerospace Engineering

by

Tomasz Sadowy-Sadowski

Student Number

5377463

Supervisor:	Dr.ir. F.F.J. Schrijer
Project Duration:	January, 2022 - September, 2022
Faculty:	Faculty of Aerospace Engineering, Delft
Track:	Flight Performance and Propulsion

Cover photo: Orbital Sciences Corporation X-34 [1]

Preface and acknowledgements

The presented master thesis concludes my 7-year academic journey. It has been a long and, at times, a difficult one, but at the end of the day the struggle was pretty much worth it. Particularly this very project cost me a lot of time, stress and dedication. Combining it with a part-time job, and long and intense preparations for my first 80km and 100km ultramarathons was no easy feat and a challenge on its own, but happily, all the goals have been achieved.

I want to use this space to express my sincere gratitude and immense appreciation to my parents, Dorota and Jacek, without whom this whole thing would not be possible. Their psychological and financial support throughout this entire journey has been invaluable, and I cannot possibly thank them enough for all they have done for me. On a similar note, the support and words of optimism from my sweet sister, Zuzanna, and my brother-in-law, Piotr, gave me much comfort when I needed it most.

But they have not been the only people that made me go through this period alive and well. My friend, Martijn, has been of great support in the moments when I was losing all the hope that any of my simulations would ever converge. He also made the preparations for the ultramarathons that tiny bit easier.

My other thesis-burdened friends, Juan, Anuj, Aditya, and Ula, also certainly made my own suffering not feel as bad as it would otherwise. In the elementary school we were taught this insurgent song, whose lyrics perfectly encapsulate how it felt. The lyrics go "Lepiej razem żyć w niewoli, niż samemu w szczęsnej doli", which can be roughly translated to "It's better to live together in captivity, than it is to live alone in fortune."

On a more serious note, I would like to thank ParaPy B.V., in particular Max Baan, for the license to use their software during this project. Similarly, I want to give my gratitude to ZONA Technology Inc., in particular to Darius Sarhaddi, for allowing me to incorporate ZONAIR as an integral part of the framework created within this project.

My very close friends from back home also deserve a big shout for all the encouragement, mental support and motivation I received from them from the distance. Daniel, Piotrek, Marek, it means the world to me. Similarly, and no less importantly, a huge thank you to Marnix and Alfie for being there for me and listening to my endless complaining about this very project here in Delft.

Lastly, I want to thank my supervisor, Dr Ferdinand Schrijer for his support and guidance that made this project finally come to a successful end.

Tomasz Sadowy-Sadowski
Delft, September 2022

Summary

Spaceplane design is an extremely complex task that involves many different, but interrelated disciplines. Under usual circumstances it can take an excessively long time and huge amount of resources until the design starts taking shape. Therefore, in order to make horizontal-take-off-horizontal-landing spaceplanes a reality, effort must be put into making this process as quick and as cheap as it can be. This is one of the leading drivers that gave birth to this thesis project, which was the creation of a Knowledge Based Engineering framework for preliminary spaceplane design. With the tight time constraints put upon this project, the number of disciplines involved in the framework had to be limited, and thus the aerodynamic analysis of the external shape was chosen as the principal focus of the thesis.

The first chapters of this document are dedicated to the introduction of the most important topics necessary to understand the applied concepts. The idea behind spaceplanes and their capabilities is covered, along with the pros, cons, and the biggest challenges in the design of such a vehicle. Then, this topic is followed by a discussion of various design techniques, both traditional and novel. It is then accompanied by a more focused review of aerodynamic design and analysis, where many different methods, simple and more complex, are presented. Then, it is further extended to the creation of aerodynamic databases by means of single-fidelity and multi-fidelity interpolation techniques in kriging and co-kriging respectively.

The concepts introduced in the opening chapters are then applied to create the framework that is the main subject of this document. It is presented in full detail with a broad discussion of each of the principal modules - the geometry module and the aerodynamic analysis module. For the geometry module, a parametrization is applied that employs only intuitive and easily visualised parameters such as lengths, angles, radii, etc. The aerodynamic analysis module consists of three parts - a low-fidelity analysis software, geometry export for high-fidelity analysis and the interpolation module that employs the kriging and co-kriging to create the aerodynamic database. ZONAIR is integrated into the framework as the low-fidelity tool, and the geometry export allows to use any high-fidelity code of choice. For this project the choice was made for Ansys Fluent.

The validation of the proposed framework consists of the analysis of the results coming from the CFD simulations, ZONAIR simulations, and lastly and most importantly, the interpolation results in the range of Mach numbers from 0.2 to 4, and angles of attack between 0° and 25°. This analysis is limited to the longitudinal motion, with no control surface deflections. The study of the interpolation results is conducted in a few steps. First, pure kriging is studied and used as the baseline comparison for the subsequent analysis of the co-kriging data fusion. Next, multiple cases with an increasing amount of expensive interpolation points are performed. The cases involve between 10 and 70 data points with a step of 10 points. Each case consists of thirty runs, and in each run the points are randomly selected from the pool of the conducted CFD simulations (87 data points in total). The four least-populated cases are highlighted and discussed in detail. The metrics used for this study are the relative error at each data point, the mean absolute error of each run (separated into three speed regimes - subsonic (up to and including M=0.8), transonic (between M=0.8 and M=1.2) and supersonic (above and excluding M=1.2), standard deviation of the error, and a qualitative comparison of the surrogate surfaces of the aerodynamic coefficients.

During the study it is found that just 20 points are enough to match kriging accuracy, provided that the points are well distributed, and 30 points can reach the expected error of less than 10%. With additional 10 points, the expected error drops to around 5% for all the coefficients. Furthermore, it is found that the distribution of the points across the domain plays a vital role in the accuracy of the predictor. Generally, to obtain good level of accuracy and minimise the number of points, the following guidelines should be kept in mind: the points should be heterogeneous, spread out to cover as much of the domain of interest as possible, with higher density around the locations where biggest gradients are expected. With very costly generation of even a single expensive data point (in this project usually between 6 to 12 hours per point, using HPC), any reduction of the amount of simulations translates to big time savings. The current approach shows that as few as 30 data points (without a numerical optimisation of their distribution) would be enough to obtain surrogate models acceptable for preliminary design stage.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

6DOF Six degrees of freedom

AAA Advanced Aircraft Analysis

AoA Angle of attack

ARS Algebraic Reynolds stress

AUSM Advection Upstream Splitting Method

BL Boundary layer

BS Bow shock

CAD Computer-Aided Design

CAE Computer-Aided Engineering

CFD Computational Fluid Dynamics

CFL Courant(-Friedrichs-Lewy) number

CG Center of gravity

CL Center-line

DLR German Aerospace Center

DNS Direct Numerical Simulation

ECW Expansion/compression wave

ESDU Engineering Sciences Data Unit

FEM Finite Element Method

HPC High performance computing

HTOHL/HTHL Horizontal Take-off Horizontal Landing

HTOL Horizontal Take-off and Landing

IS Impinging shock

ISA International standard atmosphere

JAXA Japan Aerospace Exploration Agency

KBE Knowledge Based Engineering

LES Large-eddy Simulation

LLT Lifting-line Theory

LOX Liquid oxygen

LRR Launder-Reece-Rodi

LST Lifting-surface Theory

MAC Mean aerodynamic chord

MAE Mean absolute error

MDO Multi-disciplinary Design Optimisation

MLE Maximum likelihood estimate

NASA National Aeronautics and Space Administration

NASDA Japan's National Space Development Agency

NS Navier-Stokes

OSP Orbital Spaceplanes

PDE Partial Differential Equation

RANS Reynolds-Averaged Navier-Stokes

RBCC Rocket-Based Combined Cycle

RHS Right hand side

RLV Reusable Launch Vehicle

RSM Reylonds-stress Model

SABRE Synergetic Air-Breathing Rocket Engine

SBLI Shockwave-boundary-layer interaction

SJ Supersonic jet

SL Sea-level

SL Shear layer

SP Sonic point

SS1 SpaceShipOne

SSO Space Shuttle Orbiter

SST Shear-stress Transport

SSTO Single-Stage-To-Orbit

STD Standard deviation

TLNS Thin-layer Navier-Stokes

TPS Temperature Protection System

TS Transmitted shock

TSTO Two-Stage-To-orbit

USA The United States of America

USAF United States Air Force

USSR The Union of Soviet Socialist Republics

VLM Vortex Lattice Method

VTVL Vertical Take-off Vertical Landing

WT Wind tunnel

Greek symbols

α Angle of attack

β Angle of sideslip

ϵ Turbulent dissipation rate or Error

μ Mean value

Ω Spatial domain

ω Specific rate of dissipation of turbulence kinetic energy

ρ_∞ Freestream air density

Σ Sum

σ Standard deviation

σ^2 Variance

Latin symbols

Δv Maximum change of velocity

b_{ref} Reference span

C_D Drag coefficient

C_L Lift coefficient or Likelihood

C_M Pitching moment coefficient

C_N Yawing moment coefficient

C_R Rolling moment coefficient

C_Y Side-force coefficient

c_{ref} Reference chord-length

$Corr$ Correlation

Cov Covariance

D Drag force

exp Exponential function

g_0 Gravitational acceleration

I_{sp} Specific impulse

k Turbulence kinetic energy

L Lift force

M Mach number or Pitching moment

m_0 Initial total mass

m_f Final total mass

N	Yawing moment
R	Rolling moment
s^2	Mean-squared error
S_{ref}	Reference area
T	Time interval
v_e	Effective exhaust velocity
V_∞	Freestream velocity
Y	Side-force
$Y(x)$	Normally distributed random variable at point x
$y(x)$	Value at point x

1

Introduction

Sustainability is one of the most important factors of continuous development of virtually any industry. The space industry is no different. This problem is being approached in many different ways. Among others, research is conducted to find more environmentally-friendly fuels, or increase reusability of space launch systems. The goal of the latter is to minimise the amount of expendable components, leading to a drastic waste reduction. In the relatively short history of space exploration engineers had different ideas of how it could be achieved. One such idea is the so-called spaceplane, which is the inspiration and the focal point of this thesis.

This opening chapter is the introduction to the topic treated in this graduation project. First, this curious vehicle is presented in **Section 1.1**, which provides the context to the rest of the report by briefly discussing what spaceplanes are. **Section 1.2** highlights one specific spaceplane, the X-34, the design used as the reference vehicle in the development of this project. Then, **Section 1.3** brings to attention some of the most complex aerodynamic phenomena encountered by spaceplanes along their trajectories, that stress the need for accurate solution methods. **Section 1.4** presents the objective of this research and formulates the questions to be answered in order to solve the problems at hand. Lastly, **Section 1.5** shows the structure of this document, and the order in which the topics here included are discussed.

1.1. What are spaceplanes?

This chapter is a brief introduction to the entire idea behind spaceplanes. What are they? Where do they come from and what do they offer? What are the missions they can perform, and what are the pros and cons to this concept? These and other questions will find their answers in this chapter.

The opening section gives a short historical background of spaceplanes, describing how the idea first came to life. Then, **Section 1.1.2** talks about advantages and disadvantages of such launch systems. The last section, **Section 1.1.3** describes the biggest challenges in spaceplane design.

1.1.1. Historical background

It is impossible to tell the story of spaceplanes without talking about airplanes first. The story of the concept of spaceplanes starts very early into the aviation history. Spaceplanes are just extremely advanced airplanes and can be traced back directly to the first ever heavier-than-air powered flight of Orville and Wilbur Wright on December 17, 1903 (Figure 1.1) [2]. Basic principles of winged atmospheric flight are the same for both aircraft. After the success of the Wright Brothers came many improvements to the design of the airframe and the engine. By

World War I airplanes were already so advanced that first aerial dogfights were taking place during the conflict [3].

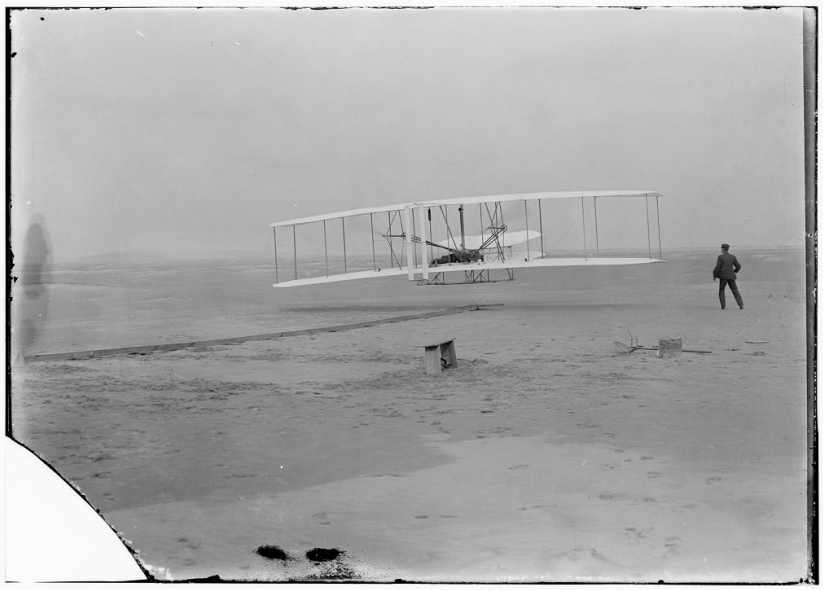


Figure 1.1: First ever flight of the Wright brothers on 17 December 1903. [4]

While the refinement of the airplane continued in the early twentieth century, some people had already started going beyond the atmosphere with their imagination. Even though the first rockets appeared in the thirteenth century in medieval China, it was not until 1861 when a concept of using rockets for spaceflight was first proposed by William Leitch [5]. The idea was further developed by a Russian teacher Konstantin Tsiolkovsky who derived the famous delta-v equation (Equation 1.1) which is the foundation of the present rocket dynamics [6]. When talking about rocketry, it is impossible not to mention the two other founding fathers of the discipline. Herman Oberth, who published two groundbreaking works on rockets, *The Rocket to Interplanetary Space* (Wege zur Raumschiffahrt) in 1923, and further expanded it into *Ways to Space Travel* (Die Rakete zu den Planetenräumen) in 1929 [7]. And Robert Goddard who, among other things, built and successfully launched the first ever liquid-fueled rocket in human history on March 16, 1926 [8].

In the 1930s, the first visions of spaceplanes started appearing in the minds of the engineers. Silbervogel, *The Silverbird*, was one such concept conceived by an Austrian engineer Eugen Sänger. He imagined a so-called antipodal bomber, which would climb to space and travel extremely large distances to be able to attack very distant areas. Contrary to his contemporary *rocket-men* who contemplated vertically launched rockets, he came up with an idea of designing a *rocket airplane*. This concept was heavier than the conventional rocket designs of the time due to the need for sturdy structures of wing root and the landing gear. This is the first registered recognition of the weight penalty imposed by these structures, that is going to be discussed in more detail in **Section 1.1.3**. In 1933, his work *Rocket Flight Engineering* (Raketenflugtechnik) was published. Later, him and his future wife Irene Bredt worked together on the refinement of the design of Silbervogel until the latter years of World War II. However, no hardware of the design was ever built. [2], [9].

Even though Silbervogel was never built, it was not the only concept of a rocket-powered plane. First such airplanes flew as early as late 1920s, as two gliders with rocket propulsion took to the skies. *Ente* on June 11, 1928 and *RAK.1* on September 20, 1929 [10], [11]. These were, however, designed as only atmospheric vehicles and used solid fuel for their propulsion.

Both machines were designed and built by German engineers. A decade later, the first ever liquid-rocket-powered airplane was created. *Heinkel He 176*'s first flight took place on June 20, 1939. The aircraft's engine was fueled by hydrogen peroxide and hydrazine. It was capable of a fifty-second-long flight [2]. When World War II broke out, it only accelerated the technological development of rocket-propelled aircraft. Less than two years after *He 176*'s first flight, *Messerschmitt Me 163 Komet* became the first operational rocket-powered airplane and set the world airspeed record of 1004.5km/h. Just like its predecessor, the plane was fueled by liquid fuel and could fly for up to eight minutes. It officially entered service for the Luftwaffe in 1944 [12].

After the war, it was Edwards Air Force Base in California in the USA that became the central point of the new wave of rocketplane research. Unlike its subsonic German predecessors, the new-coming American machines were designed for high-speed and high-altitude research, eventually breaking the sound barrier for the first time. The vehicle that first achieved this feat in a horizontal flight was the *Bell XS-1* on October 14, 1947. To save propellant, the plane did not take off by itself but was carried by a B-29 bomber and released at the altitude of 6400m. This groundbreaking experimental aircraft was being developed and modified until the late 1950s [13]. Another important milestone in rocketplane history was marked by *Douglas D-558-2 Skyrocket*, the first aircraft to reach Mach 2, double the speed of sound, on November 20, 1953 [14]. Then, on September 7, 1956, the second in the family of X-planes marked another highly significant milestone. *Bell X-2 Starbuster* was the first ever airplane that flew above 100,000ft (30.5 km), at an altitude of 126,200ft (38.5km), earning the pilot of the vehicle, Cpt. Iven C. Kincheloe, the title of "America's First Spaceman". That same aircraft was also the first to fly triple the speed of sound only 20 days later [13].

Another vehicle that needs to be mentioned when discussing spaceplanes is *the X-15*, shown in Figure 1.2. In 1959, this research aircraft, dedicated to studying high-speed and high-altitude flight, started its operations. The aircraft broke multiple records, such as the highest speed by a crewed, powered aircraft, when it flew at Mach 6.7. This record remains unbroken to this day. On top of the speed records, it was also the first winged aircraft to fly above the 100km mark, i.e., the first winged vehicle to fly in space. It happened on two separate occasions in 1963, on July 19 and August 22, when the aircraft flew at 105.9km and 107.8km altitude respectively [15].

In the meantime, on August 2, 1955, the competition known as *the Space Race* began. On that day, four days after the US announced their intention of launching its very own satellite, the Soviet Union responded. This is now recognised as the official beginning of the race. This event led to a great acceleration of space technology that resulted in many enormous accomplishments from both sides [2], i.a.:

- *Sputnik* - the first artificial satellite (October 1957, USSR)
- Yuri Gagarin - the first human in space (April 1961, USSR)
- Alexei Leonov - the first space-walk (March 1965, USSR)
- *Apollo 11* - the first manned Moon landing (July 1969, USA)

Figure 1.2: *The X-15* in flight. [16]

Between the years 1963 and 1975 NASA's Flight Research Center ran a joint effort to validate the concept of a wingless vehicle that would be able to fly back from space and land at a selected site after having successfully re-entered the atmosphere. There were 6 unique aircraft with very unconventional aerodynamic shapes. The lift was provided by the shape of the vehicles themselves instead of the traditional wings. All the lifting bodies had fins and control surfaces to stabilize and control the aircraft. They were designated *M2-F1*, *M2-F2*, *M2-F3*, *HL-10*, *X-24A*, and *X-24B*. Together, all the lifting bodies had 221 flights over this period. Much information gathered from the Lifting Body Program later helped in the development of the Space Shuttle [17].

Figure 1.3: Three of NASA's Lifting Bodies, *X-24A*, the *M2-F3*, and the *HL-10*. [17]

After the success of the *Apollo 11* mission in 1969, NASA started considering the next major goal - a manned space station. In order to make the goal achievable, transportation costs of the components needed to be reduced. A major part of the space launch expenses came from dropping off costly, expendable hardware. Hence, a reusable launch system was proposed, that eventually evolved into *the Space Shuttle Program*. It was not the first ever concept of a reusable launch system, but certainly the most successful till this day. The initial design was supposed to be completely reusable, with a first stage in a form of a fly-back vehicle of the size of a *Boeing 747* and the second stage, an Orbiter, that would continue to the orbit and then return safely to the ground. However, the project was found to be too expensive to develop, and, to accommodate different requirements, it ended up as a 3-stage system [18]. It consisted of two solid rocket boosters, an external fuel tank, and the Orbiter. The Orbiter had a double-delta low-wing, no horizontal tail and one vertical fin. For additional stability control it used a body flap located at the rear side of the fuselage. Apart from the Space Shuttle, only two other winged vehicles were ever successfully launched to the orbit and came back after re-entry: Russian *Buran* and the *X-37* [18], [19], [20]. *The Space Shuttle* is without a doubt the most successful of the lot, having operated for 30 years between 1981 and 2011 during which it flew more than 120 times [21]. However, till this day there have not been a single vehicle that would be capable of both going to space, and taking off and landing horizontally. Thus, for now, *the Space Shuttle* is the closest humanity has been to a spaceplane.

1.1.2. Spaceplanes: pros and cons

As of yet, there has been no true and fully operational spaceplane. So why is this concept still alive? Why do engineers still try to design a spacecraft that would be able to fulfill such a difficult mission? Also, what are the drawbacks of such a complex vehicle? This section tries to answer these questions by briefly talking about pros and cons of spaceplanes.

The entire idea of a spaceplane seems pretty straightforward. It is a horizontal-take-off-horizontal-landing, advanced airplane that is also capable of flying outside of the atmosphere, then re-enter and land safely with no hardware being wasted in the process. What takes-off, comes back to the airport. Obviously, excluding the payload that is delivered to the orbit. But why not just use the rockets as it has been, and still is, done? There are a few aspects that can explain the advantages of this idea over the traditional rockets.

Rockets always consist of multiple stages that, up until very recently, have been single-use and have been discarded once they served their purpose. This is slowly changing thanks to *SpaceX*. Even though theirs were not the first rockets to have ever landed vertically on Earth (*McDonnell Douglas' DC-X* was [22]), they are the most successfully operating ones and the most promising in that regard. But even those are not yet fully reusable, with both *Falcon 9* and *Falcon Heavy* having only certain parts of the system able to be recovered and relaunched [23], [24]. The first fully reusable rocket is supposed to be their *Starship*, which is still in the development phase [25]. In the same context, the most important feature of spaceplanes that makes them such a welcome invention is their reusability. Just like traditional airplanes, spaceplanes are designed to be used often and repeatedly. They land with all the parts they took off with, and, contrary to *SpaceX*'s rockets, they can always land at the same place they took off from. The rocket-stages, depending on their speed at separation, would need to land many kilometers downrange from the launch site. Furthermore, in case of an in-flight failure, ballistic vehicles have no chance of recovery and might fall almost anywhere. Spaceplanes, thanks to their wings, can be glided and guided back to the airport, even if a failure occurs.

Considering a crewed flight, spaceplanes would also cause drastically lower stress on the crew, given their wings and the lifting re-entry mission profile. These lower G-forces would mean the vehicle would be much more comfortable and safer for the people inside.

Another important characteristic of spaceplanes is their increased safety relative to vertically launched vehicles. To put this into perspective, statistically, there is a chance of only one in five million for an airplane passenger to die once they board. In contrast, almost one in seventy launches of *the Space Shuttle* ended up killing the entire crew. 2 failures in 135 total launches. Both of them associated with the way the vehicle's vertical take-off [21]. To add to that, the statistics for 2021, as of December 8, show that 7 out of 126 overall orbital launch attempts ended in failures [26]. It shows that the reliability and safety of vertically launched vehicles is incomparably lower. Vehicles capable of taking off horizontally would drastically improve both of these aspects. The idea behind the spaceplane is to combine airplanes' safety and reliability with the capability of rockets to deliver payload to space.

Operational simplicity of spaceplanes is yet another aspect of the design that would make them a desired option to replace traditional vertically launched vehicles. What is meant by operational simplicity is that well designed spaceplanes would consist of a single part. Contrary to current launch systems that operate as multistage vehicles, spaceplanes would retain all their parts from taking off till landing. Thus, the complexity of preparing it for another launch would be drastically decreased. Furthermore, maintenance should also be significantly easier. On top of that, thanks to the spaceplanes' capabilities of horizontal take-off and landing, they would be able to use the already available infrastructure such as runways, fuel stations, repair shops. No exclusive launch zones or airways would be required. Hence, spaceplanes would operate in harmony with the currently existing air traffic.

Last main point of a good spaceplane design is that, unlike rockets, they would use the atmosphere to their advantage instead of treating it as an obstacle on their way out. Depending on the exact design, spaceplanes could use it in three different ways: for aerodynamic lift, as oxidizer for sustained combustion, or even as *working mass* (a propellant), if an air-breathing engine is installed on the airframe. Firstly, the air would be used during take-off, minimizing the gravity losses essentially to zero, as well as during glide-back, just like *the Space Shuttle*. The other two applications would depend on the type of engine used by the spacecraft. If an air-breathing engine is part of the powertrain, it would help limit the amount of fuel and oxidizer that need to be carried by the vehicle. Air consists of around 20% oxygen, and thus can be used to sustain combustion inside engines, just like in conventional airplane engines. Moreover, the atmosphere provides air as the *working mass* that is accelerated by the engine, and thus thrust is produced and the vehicle is propelled forward. Once a high enough altitude is reached, and the air cannot be used for any of these applications, the spaceplane would go into rocket-mode to fly ballistically into space. By then it would have already gained a lot of potential energy (altitude) and kinetic energy (velocity), what would minimize the additional energy it would need to obtain in order to reach the orbit. Hence, working with rather than against the atmosphere could provide many benefits.

However, there are also a few disadvantages to spaceplanes. First and foremost, they pose a very complicated design problem. Which is why the world has not seen one yet. Unlike airplanes, they do not have a design point for which engineers could optimize. There are many requirements, all of which need to be properly balanced in order for the design to work. Inside its flight envelope, a spaceplane experiences many drastically different flight conditions, speed regimes, or angles of attack. Finding equilibrium between low subsonic, supersonic and even hypersonic performance is no easy task. Add to it the extreme temperatures that the vehicle experiences on during the high-speed portion of its mission, and the complexity of its propulsion system, and the puzzle seems almost unsolvable.

Another important drawback that only adds to the intricacy of the problem is all the weight that becomes unnecessary once the vehicle leaves the sensible atmosphere. Wings, stabilizers, landing gear, air-breathing engine. All these components are useless in the vacuum and only

limit the amount of payload that can be carried to space. The real challenge is, therefore, finding a design that allows the cons of carrying this extra weight be outweighed by the advantages these subsystems provide.

1.1.3. Challenges in spaceplane design

The disadvantages mentioned towards the end of the previous section can mostly be seen as challenges that spaceplane designers must face. Once resolved, the design might finally become feasible. However, they are not the only ones that the engineers need to tackle. This section further expands upon these drawbacks and tries to explain that solving these problems might actually be possible.

As already discussed, one of the major obstacles is that an air-breathing engine, wings, stabilizers and landing gear only add unnecessary weight to the vehicle once it is outside of the atmosphere. But it does not mean that these parts are useless for the mission. On the contrary, they help the spaceplane in the most crucial part of its mission, which is to reach this altitude in the first place. Then, they help the vehicle find its way back to the airport after a successful re-entry during the glide-back. The challenge is to find a design that saves more mass on the fuel not taken aboard than is added by carrying all the weight coming from these components. To add to it, high temperatures experienced during re-entry require it to have some kind of temperature protection system (TPS). Not only do the wings increase the weight by themselves, the increased frontal area caused by their presence makes the TPS only grow heavier.

Then, this vehicle needs to be able to perform in a variety of different flight conditions, which means that it does not suffice to just add any wings and stabilizers to the fuselage. They also need to be made in such a way that low speed, high speed, and high temperature performance is balanced well enough for the spaceplane to carry out its mission. All these conditions have different, and frequently opposite, requirements. Consider wings as an example. Generally, for good low-speed performance the leading edge should have a relatively large radius and the airfoil should be rather thick. On the other side of the spectrum are the requirements for adequate supersonic performance. Sharp leading edges and thin airfoils are preferable in this case. But as the speed increases and the vehicle enters the realm of hypersonic flows, very high temperature effects come into play and drag minimization is no longer the leading factor. Keeping the temperature as low as possible is added to the equation, and what helps with that is, again, a blunt leading edge. Bluntness helps with reducing the surface temperatures by producing thick boundary layers, which in turn increase the efficiency of surface-radiation cooling [27]. However, spaceplanes are designed for both ascent and re-entry. That is, two flight stages with opposing requirements. During ascent the vehicle needs to have low drag in order to accelerate quicker. For that, bluntness is not desired. But during the re-entry it should produce more drag for more effective breaking, and thus less time spend in high-speed and high-temperature conditions. In both of these stages temperature should be kept below a certain level, which depends on the materials used. Furthermore, the choice of materials themselves is a story of its own, with balancing heat resistance, stress resistance and weight. Finding balance between such extreme conditions is what makes spaceplane design so complex.

Spaceplanes' powertrain is yet another aspect of this puzzle that might give the designers headaches. How do you create a system that is able to use the atmosphere to its advantage but is also able to perform perfectly well when there is none? Conventionally, what goes to space is propelled only by rocket engines and relies on staging to reach the orbit. The atmosphere is not used by the propulsion system at all, and everything the vehicle needs in order to accelerate is carried on board. However, since the winged launch system is already going to use the atmosphere for lift instead of treating it as a barrier that must just be crossed,

what if it could be used by the power train too? This thesis project is not propulsion oriented, thus it suffices to say that it would require a system that saves enough weight on fuel to justify the complexity and added mass of said design.

Whenever space launch technologies are discussed, the famous rocket equation of Konstantin Tsiolkovsky should be brought up. It is needed to fully understand the next big challenge of spaceplane design. In Equation 1.1 Δv is the maximum change of velocity of the vehicle, m_0 is the initial total mass, m_f is the final total mass and $v_e = I_{sp}g_0$ is the effective exhaust velocity. This equation shows that in order to produce a certain change in velocity, a certain ratio between the full and dry masses is needed. The higher the ratio, the bigger the change. This is why rockets employ the often-mentioned staging technique, as it is the easiest way of achieving a high mass ratio. Once all of the propellant has been used by a stage, it is discarded so as not to carry the unnecessary weight, i.e., the tank that contained the propellant and the oxidizer of a stage. Spaceplanes, however, are ideally supposed to be single stage. And this is the big challenge for the designers. How to get the highest possible mass ratio contained within a single stage?

$$\Delta v = v_e ln\left(\frac{m_0}{m_f}\right) = I_{sp}g_0 ln\left(\frac{m_0}{m_f}\right) \tag{1.1}$$

1.2. The Orbital Sciences X-34

When the Cold War ended in the fall of 1989, the funding of space-related technology was greatly impacted. As a result, NASA, the Department of Defence and other interested parties lost much of its investments, what made them need to cut down on both new and old launch systems. Therefore, the need for cheaper space delivery drastically increased. The Space Shuttle Program was becoming increasingly difficult to support, with great portion of its costs coming from expendable boosters. It resulted in three major programs, all of which were designed as SSTO vehicles: *the Orbital Sciences X-34* (Figure 1.4), *DC-XA Clipper Graham* (Figure 1.5), and *the Lockheed Martin X-33* (Figure 1.6).

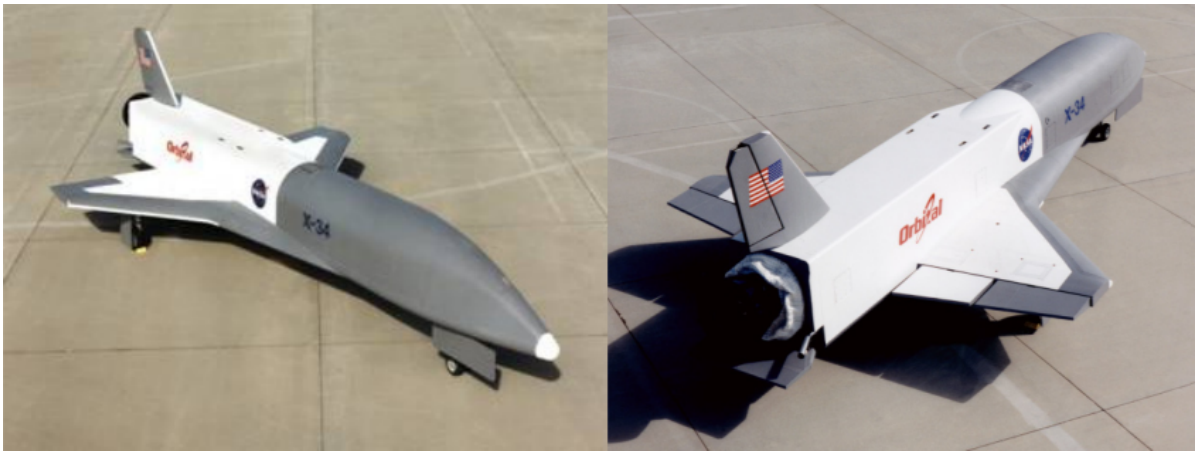


Figure 1.4: Two views of Rocket Sciences' advanced technology demonstrator X-34. [28]

Figure 1.5: DC-XA firing its engines. [22]

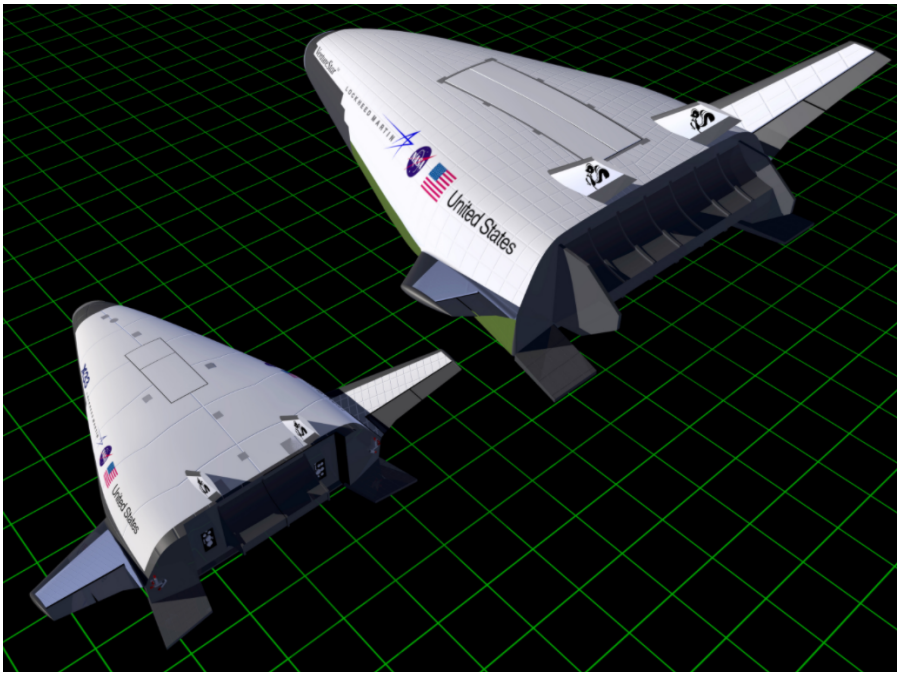


Figure 1.6: Visualisation of NASA's VentureStar and its scaled technology demonstrator X-33. [29]

The X-34 was a joint effort of Orbital Sciences Corporation and Rockwell International, started in 1994. It was being designed as a SSTO HTOL vehicle. In its shape very reminiscent of the Space Shuttle Orbiter with a double-delta-wing, flat bottom and a single vertical tailplane at the rear of its slender fuselage. Although at first very promising, the program had to be cancelled in 2001 because of lack of funds to continue the research. During the years the program ran, various wind-tunnel test campaigns were conducted resulting in a relatively big aerodynamic database of the vehicle [30], [31]. On top of that, a couple of prototypes were built but they were only ever used for towing and captive flight tests, and no real launch was ever performed. [32]

1.3. Spaceplane aerodynamics and computational modeling

Along their trajectories spaceplanes experience a wide variety of different aerodynamic phenomena. Depending on the underlying assumptions and simplifications of the aerodynamic analysis tools, they might *disappear* from the aerodynamic model. This section offers a brief overview of some of these phenomena that can have a big impact on the aerodynamic performance of spaceplanes, but might be missed by some lower-fidelity tools. In this section references are made to **Section 3.2**, where various flow aerodynamic analysis tools and flow models are introduced. The phenomena discussed in this section are limited to four of the most probable to occur and of highest influence along a spaceplane's trajectory. These are: Wing-body junction flow, Shockwave-boundary-layer interactions, Shock-shock interactions and High angle of attack aerodynamics. Figure 1.7 gives a schematic overview of where these phenomena are most likely to occur.

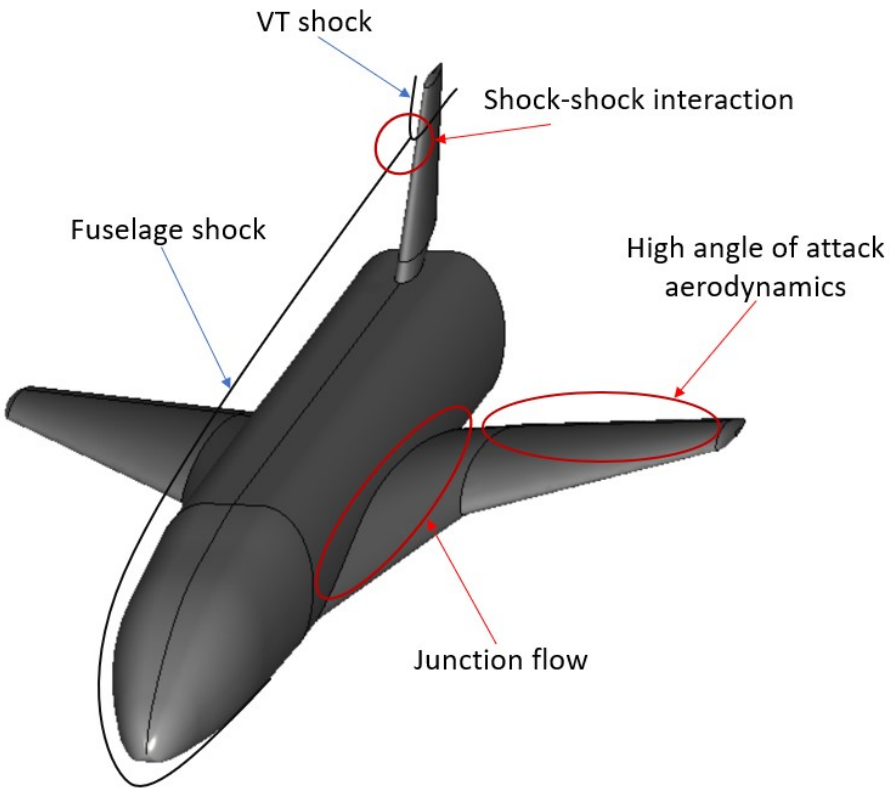


Figure 1.7: Schematic overview of the complex flow phenomena discussed in this section. Remark: high angle of attack aerodynamics are not limited to wing trailing edge, and SBLIs can happen virtually anywhere on the vehicle (depending on the exact flow conditions).

1.3.1. Wing-body junction flow

The first phenomenon that is important to look at is wing-body junction flow. Just like the name suggests, this is the flow at the intersection of a wing and a body (e.g., a fuselage), where there is a strong interaction between the boundary layers generated on their respective surfaces. The vortices created in such flows can lead to buffeting and decreased aerodynamic performance. The drag induced by such an occurrence can contribute to even 10% of the total drag of civil aircraft [33].

The problematic nature of a junction flow comes principally from the wing disrupting the vortices of the boundary layer that developed around the body, and from merging of this vorticity with the one that formed over the wing surface. This interaction gives rise to a complex and highly anisotropic flow, caused by the fact that the two boundary layers are not oriented in the same direction. The complexity of the flows is all the higher with the multitude of secondary flows that can be present within the corner. Figure 1.8 demonstrates a typical wing-body junction flow with a horseshoe vortex and a corner separation.

Modeling of such a phenomenon is impossible without modeling the fluid flow around the object with a volume mesh (vide **Section 3.2.4**). And even that does not guarantee to fully resolve this issue. For example, Coombs et al. [34] studied various turbulence models, from one- or two-equation models (i.a., *Spalart-Allmaras*, *realisable* $k-\epsilon$, $k-\omega$ *SST*) to Full Reynolds Stress Models such as *Launder-Gibson* and *LRR*. They came to the conclusion that frequently the simpler closure models are not fully capable to resolve the junction flows. Without going into too much detail, as it would be outside of the scope of this project, it suffices to say that the complexity of such a flow requires precise (and costly) computational modeling to accurately predict the flow field around any flying vehicle.

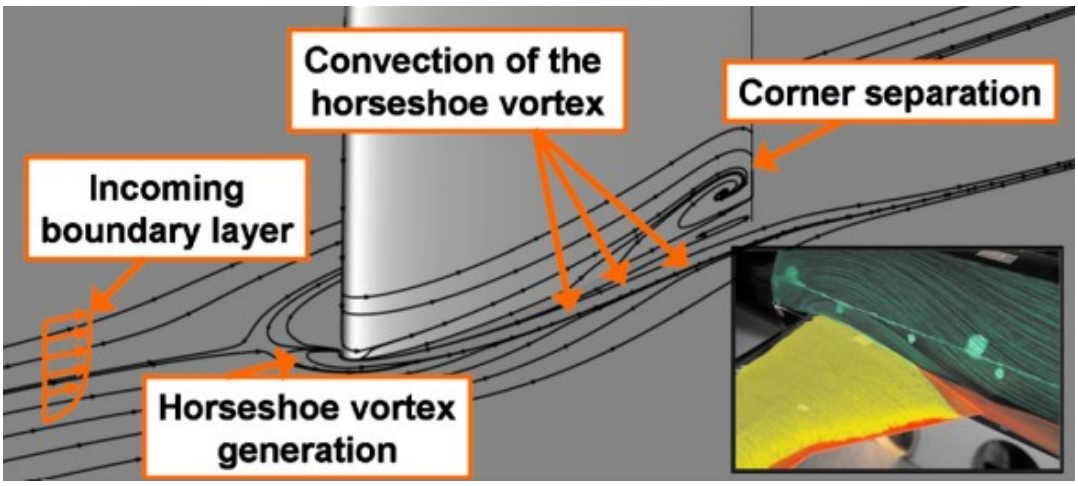


Figure 1.8: A sketch and a wind-tunnel visualisation of a junction flow with interesting flow structures. [33]

1.3.2. Shockwave-boundary-layer interactions

A shockwave-boundary-layer interaction (SBLI) is any occurrence when a shockwave meets a boundary layer. Both boundary layers and shockwaves can be observed in virtually all transonic, supersonic and hypersonic flows, which make this phenomenon a particularly frequent event during a spaceplane mission. They can arise in many different situations, i.a. when a shock generated *externally* to a surface with a boundary layer impinges onto it, or when a sharp compression is produced as a result of the change of the slope of the surface - in a supersonic flow this compression would usually originate from within the boundary layer. Whenever such an interaction occurs, a highly adverse pressure gradient is applied by the shock on the boundary layer. It leads to the thickening of the boundary layer, and might even

cause separation. Even if it does not separate, the viscous dissipation of the flow is still increased during such an interaction. Viscous dissipation is an irreversible process, in which the work realized by the fluid's shear forces is converted into heat, which strongly influences heat transfer, particularly in high-speed flows. In short, higher viscous dissipation is undesirable if the temperature of the flow is to be minimized, which becomes more and more important as the velocity of the flow increases. In hypersonic flight, it can lead to severe concentrated heating, which is highly detrimental to the vehicle's structure. Another unwanted effect of SBLIs is the flow unsteadiness, which can cause buffeting, particularly in transonic flows. Some SBLIs, visualised with Schlieren photography, can be seen in Figure 1.9.

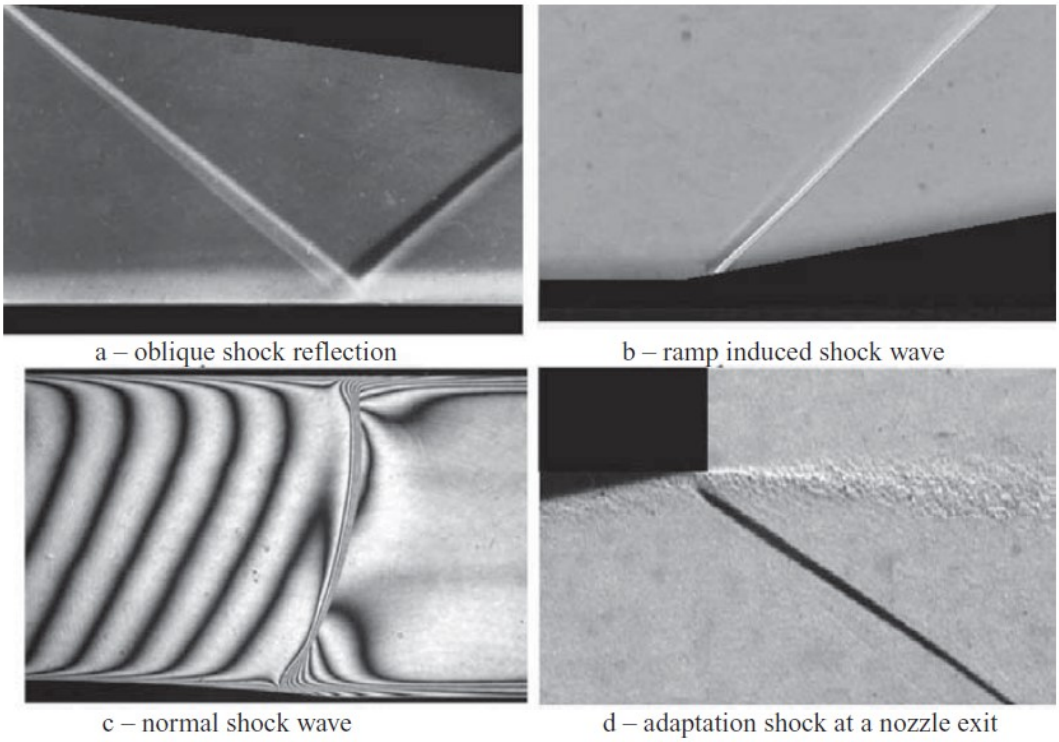


Figure 1.9: Basic shockwave-boundary-layer interactions, visualised with Schlieren photography. [35]

There is a variety of ways shockwaves can interact with boundary layers, most of which are detrimental to the vehicle's performance. They can lead to boundary layer separation, a rise in temperature, flow instabilities, enhanced turbulence, etc. The SBLIs can greatly impact the entire flow field, with intense vortex formation and a generation of complex patterns of secondary shockwaves. The complexity of this phenomenon makes it especially hard to model. Simple, low-fidelity calculation methods such as the ones described in **Section 3.2.2** are unable to predict this phenomenon. And, similarly to junction flows, they require accurate discretization to be properly resolved by even higher fidelity CFD methods. [35], [36]

1.3.3. Shock-shock interactions

In many high-speed flows, shockwaves can interact with other shockwaves, leading to so-called *shock-shock interactions*. For example, a shockwave generated at the nose of the fuselage can travel downstream and impinge on the shock produced by the leading edge of the wing or the vertical tail. Such interactions can be relatively straightforward and can be computed with algebraic methods based on the classical compressible flows theory (e.g., cases involving only straight oblique shockwaves), or highly complex so that they can only be resolved with time-marching numerical methods (e.g., cases with a combination of curved and straight shockwaves that generate a mix a subsonic and supersonic flows). In any case,

viscosity plays little to no role in shock-shock interactions, as they are primarily driven by inviscid phenomena.

This group of phenomena can be very detrimental to the vehicle's performance and can easily lead to its destruction if not adequately accounted for. The patterns resulting from shock-shock interactions depends on the location where the *incoming* shock hits the curved bow shock. These patterns can be divided into six distinct types, and thus, the bow shock can be divided into six sections - as illustrated in Figure 1.10.

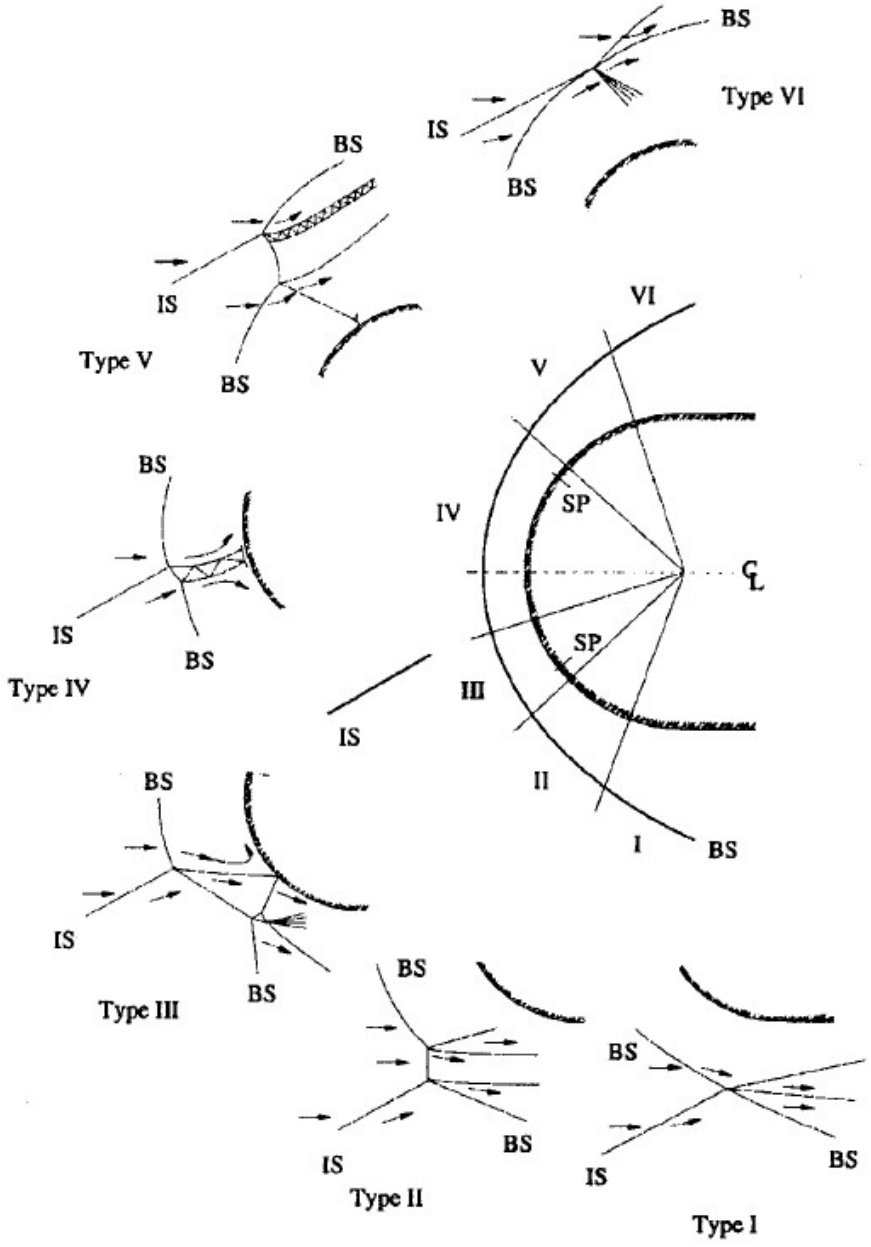


Figure 1.10: The six types of shock-shock interactions. [37]
SP - sonic point, IS - impinging shock, BS - bow shock, CL - center-line.

A detailed description of all the shock-shock interaction types is outside of the scope of this project. However, it is important to highlight the most severe, type-IV interaction (shown in more detail in Figure 1.11). It occurs when the impinging shock strikes the stronger part of the bow shock, typically, at a location significantly upstream of the sonic point. The supersonic jet,

formed behind the point where the incoming shock impinges on the bow shock, penetrates the flow field towards the body and finishes in a normal shock in close proximity of the surface of the body. It leads to big peaks in the pressure distribution, and increased aerodynamic heating. The observed and calculated pressure and heating rates can reach even 30 times the rates expected for a case with no interaction.

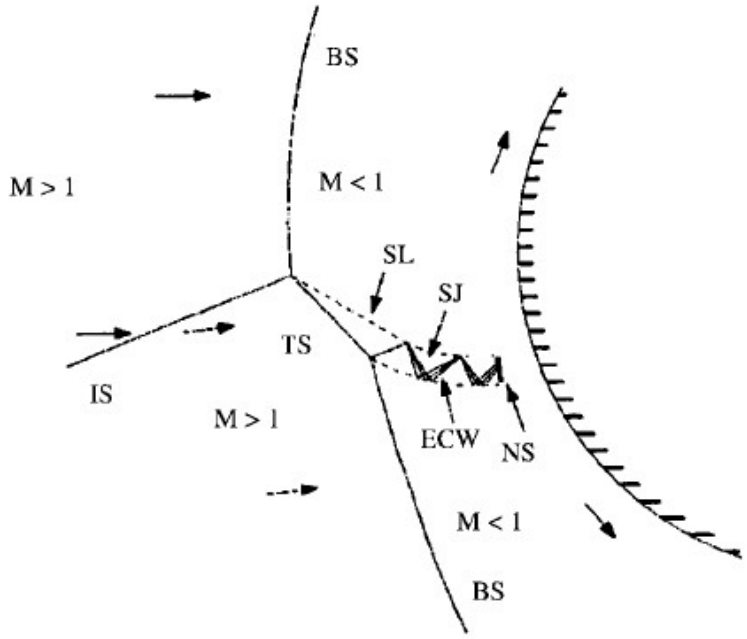


Figure 1.11: Type-IV shock-shock interaction. [37]
IS - impinging shock, BS - bow shock, ECW - expansion/compression waves, SL - shear layer, SJ - supersonic jet, TS - transmitted shock.

In a three-dimensional flow around complex geometries these interactions can be very hard, if not impossible, to predict by simplified methods. Their appearance in the flow field can highly affect the performance of the analysed vehicle. Thus, high-quality domain discretization with adequate solver settings might become indispensable to properly account for these phenomena. [38]

1.3.4. High angle of attack aerodynamics

This section gives a brief overview of some phenomena that can be found in flow fields around vehicles flying at high angles of attack, and how these phenomena are sometimes hard to predict precisely, even with higher-fidelity methods. The characteristics of high-angle-of-attack flows are driven by, i.a.,

- the free shear layers produced at sharp, or highly swept, leading edges
- viscous layers separating from surfaces of wings and bodies
- flow inside the wakes generated behind wings and bodies

The complicated flow field resulting from the strong interactions between such vortical flows is highly three-dimensional, and, unsurprisingly, frequently hard to predict. Phenomena such as the rolled-up vortices or separating vortex sheets are a more and more frequent occurrence as the angle of attack increases, and they do have a large influence over the pressure distribution and the viscous effects around the vehicle. Accurate modeling of such effects is paramount when high angles of attack are expected along the vehicle's trajectory, e.g., such as during

the spaceplanes' re-entry phase.

The complex three-dimensionality with strong viscous-inviscid interactions makes the theoretical analysis of the entire field virtually unrealistic. There do exist, however, some simplified, non-linear methods and approximate theories that allow to compute the aerodynamic characteristics of flying vehicles to a reasonable degree without resolving to high-fidelity numerical methods. Some panel methods can *reproduce* the non-linearity of the aerodynamic characteristics at high angles of attack through boundary conditions dependent on the rolled-up wakes, their shape and position. They model rolled-up vortex sheets originating from the leading edges and wingtips, what allows a better approximation of these effects than a simple, flat vortex sheet behind the trailing edge (**Section 3.2.2.2**).

However, these methods have their limitations, and their application to the calculation of vortical flows must be approached with some reservations. Especially, since these effects do *behave* differently in different speed regimes, and once the compressibility effects start coming into play, and the flow field becomes even more complex (e.g., shockwave-boundary-layer interactions - **Section 1.3.2**), the simplified methods drift farther and farther away from the *true solution*. [39]

1.3.5. The need for more accurate solutions

This section introduced some complex aerodynamic phenomena that can highly influence the performance of spaceplanes. As already frequently stated, if they are not accounted for, the predicted aerodynamic characteristics can give a very wrong image of the true behavior of the analysed configuration. And as important as they are, they are also very hard, if not impossible, to model using only simple methods. And while such methods do give some insight into the vehicle's aerodynamic characteristics, they do not show the full picture. They do tend to overlook some issues that can be detrimental to the configuration's performance. And the sooner such problems can be found, the easier it is to fix them. This is why more advanced aerodynamic analysis methods should be used in the design process as early as possible (vide **Section 2.2**). The problem is, early stages of vehicle design do require a high iteration count, which is why the simple methods are still very much needed. Therefore, this thesis project will aim at creating a *bridge* between low- and high-fidelity methods. The project's goal is to create a framework that would allow to compute an aerodynamic database with high accuracy and relatively low computational overhead.

1.4. Problem statement

This section consists of two main parts: **Section 1.4.1** discusses the objective of this thesis project and **Section 1.4.2** presents the main research questions that need to be answered in order to fulfil the research objective.

1.4.1. Research objective

The research pursued in the graduation project aims at developing a framework that provides efficient and flexible parametrization of spaceplane configurations, later to be analysed in terms of their aerodynamic performance. What is meant by flexible, is that the parametrization should allow for creation of many different geometries representative to spaceplanes. The application should also allow for modeling the control surfaces such as ailerons, rudders, elevons, or body-flaps. All this should lead to an improved process of preliminary design by providing aerodynamic analysis with increased accuracy at this design stage. Therefore, the primary research objective is:

> Develop a framework to assist in the preliminary design stage of horizontal-take-off-horizontal-landing single-stage-to-orbit spaceplanes combining 3D-model generation with high-fidelity and low-fidelity aerodynamic analysis tools by using a Knowledge Based Engineering system.

On the way to achieve the main goal of this project, there are some sub-goals that first need to be completed. To start, an efficient and flexible parametrization of the geometry of the spaceplanes needs to be found and implemented. An adequate set of constraints needs to be put in place in order to allow the creation of only feasible configurations. Modeling of control surfaces should be included in the framework. The generated geometry then needs to be exported as a 3D-model to be analysed in terms of its aerodynamics. A proper scheme needs to be found and applied that would allow for an accurate correlation of low- and high-fidelity methods with only a small amount of high-fidelity data added, in order to keep the computational time as low as possible.

1.4.2. Research questions

The following research questions need to be solved in order to reach the goal of this graduation project:

1. What is a geometrical parametrization that allows for high flexibility in configurations representative of spaceplanes?
2. Which low-fidelity and high-fidelity methods of the aerodynamic analysis best suit the intended framework in terms of accuracy and computational time?
3. What are the typical ranges of Mach numbers and angles of attack that HTHL SSTO spaceplanes experience during their missions?
4. How can low-fidelity methods be correlated with high-fidelity methods to improve the global accuracy of the generated data?
    - Is there a scheme that correlates low-fidelity and high-fidelity analysis in an accurate and efficient way when a small amount of high-fidelity data is added?
    - What is the number of high-fidelity data-points needed that provides the best balance between computational speed and the accuracy of the data?
5. What is the accuracy, and what are the limitations of the created framework?

1.5. Report structure

The report culminating this graduation project is structured in the following way. The first chapters are dedicated to the discussion of the topics necessary to understand the fundamental disciplines and ideas underlying the present study. The last three chapters present the current work and thoroughly study its results.

Chapter 2 introduces the general concepts of aircraft and spacecraft design. **Section 2.1** looks into the traditional design approach, and **Section 2.2** focuses on some modern techniques that can help accelerate the process. Following is **Chapter 3**, where one specific aspect of aerospace vehicle design is explored, where **Section 3.1** looks at the principles of aerodynamic design, and **Section 3.2** discusses multiple tools that can be employed for aerodynamic analysis of such vehicles. Then, **Chapter 4** introduces the topic of aerodynamic database. In this very chapter two interpolation techniques used for building databases are discussed: single-fidelity technique called kriging in **Section 4.1**, and its extension - a multi-fidelity data fusion technique called co-kriging in **Section 4.2**.

Chapter 5 explains all the details about the framework created to realize the task at hand. In it, **Section 5.1** discusses in detail the architecture of the framework. Then follows **Section 5.2**

where the implemented geometry parametrization is presented. The next two sections focus on the aerodynamic analysis tools employed for the low-fidelity (**Section 5.3**), and the high-fidelity calculations (**Section 5.4**). The two tools are discussed in terms of their capabilities, limitations, setup, etc. The chapter is concluded with the implementation of the module to build the aerodynamic databases in **Section 5.5**.

Chapter 6 is the study of the results obtained during the realization of the current project. The methodology is given in **Section 6.1**. Then, the mesh independence study is curried out for both low- and high-fidelity tools (**Section 6.2**). An overview of the results before interpolation is provided in **Section 6.3** for Ansys Fluent, and in **Section 6.4** for ZONAIR. Lastly, an in-depth discussion of the results of the implemented interpolation techniques is the subject of **Section 6.5**. The study is concluded in **Chapter 7**, where the results of the study are summarised (**Section 7.1**), followed by recommendations for future work (**Section 7.2**), and lastly the potential outlook and implementation of the entire framework is discussed in **Section 7.3**.

2

Aircraft and spacecraft design

This chapter is dedicated to the aircraft and spacecraft design, which is one of the most complex processes in engineering. Traditionally, it is approached in a relatively discontinuous manner consisting of three nearly separate stages, which are outlined in **Section 2.1**:

- Conceptual design (**Section 2.1.1**)
- Preliminary design (**Section 2.1.2**)
- Detail design (**Section 2.1.3**)

But this conventional approach has its obvious shortcomings, having developed a compromise between design freedom and complexity [40]. These limitations gave rise to novel design techniques that are the subject of Section 2.2, such as Multidisciplinary Design Optimisation (**Section 2.2.1**) and Knowledge Based Engineering (**Section 2.2.2**).

2.1. Traditional approach

Traditionally, the design process is divided into three main stages: conceptual design, preliminary design and detail design [41], [42]. Figure 2.2 illustrates the general overview of the design process and what these stages typically consist of. Figure 2.1 shows these principal phases and milestones for a traditional development process of an aircraft. As can be seen, this is rather discontinuous, with a marginal overlap between the design phases. This distinction between phases is not just illustrative but is also still in use by big aerospace companies such as Airbus, which uses technical milestones to discriminate between these phases in their development process [40]. It is true that the technological progress and the development of the new techniques have moved high-fidelity methods a bit more toward the front of the design process, making the boundaries between the phases less pronounced. However, the reality of the current approach in the industry is still closer to what is shown in the graphic.

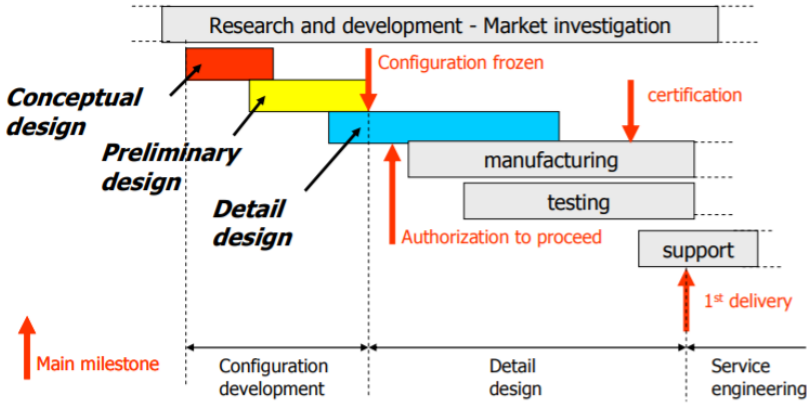


Figure 2.1: The three main phases and milestones in the traditional aircraft development process. [40]

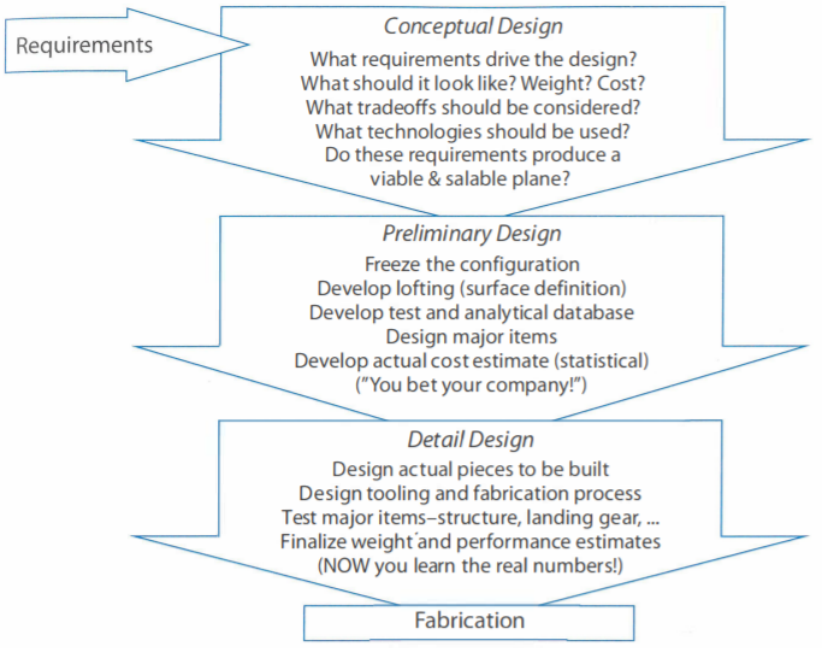


Figure 2.2: Three main aircraft design stages [42]

2.1.1. Conceptual design phase

Conceptual design, which is the first of the traditional design phases, is obviously the most unrestricted one. Here, the general requirements are defined with respect to the function and form of the vehicle, but also the requirements themselves might be impacted and changed. The designers consider a vast range of concepts, configurations and solutions to specific problems. Figure 2.3 shows an example of various concepts considered for the Space Shuttle program. Eventually, at the end of this phase, only one configuration has to be chosen for further refinement in the preliminary design phase. This is done through trade studies of designs as well as requirements, frequently with significant input from potential customers. A typical loop of the conceptual design phase of an air-/spacecraft is presented in Figure 2.4.

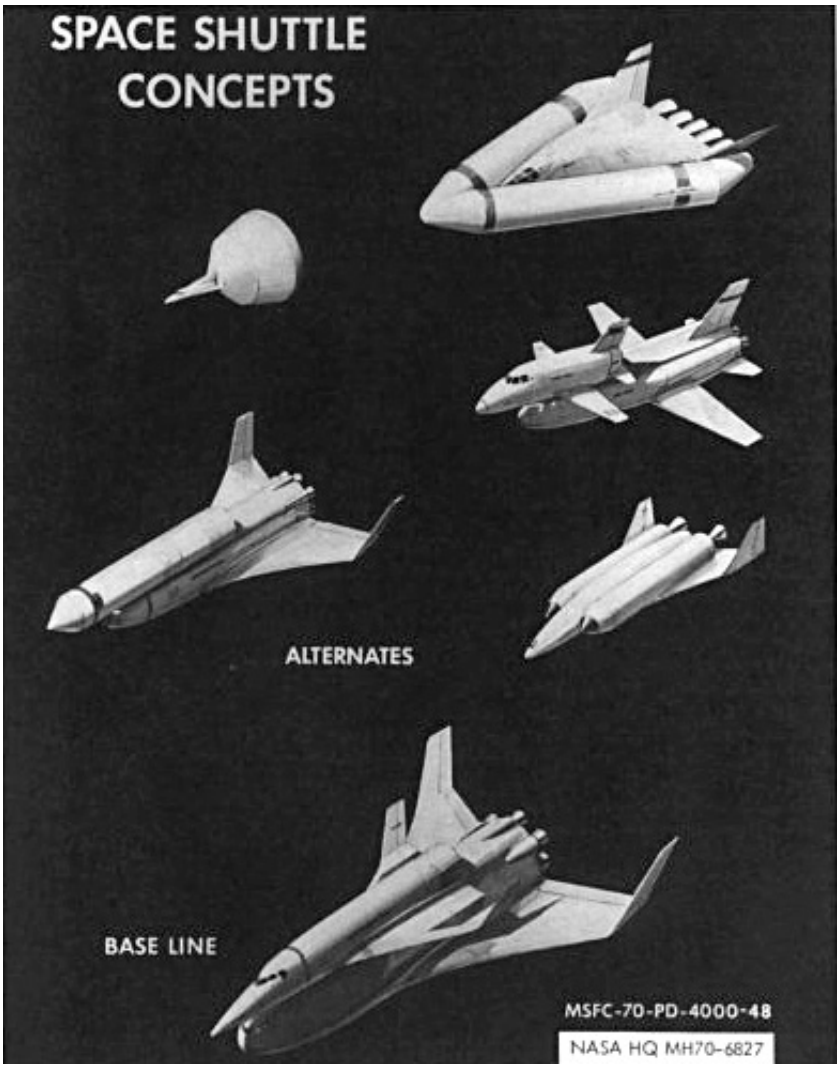


Figure 2.3: Various Space Shuttle concepts visualised. [43]

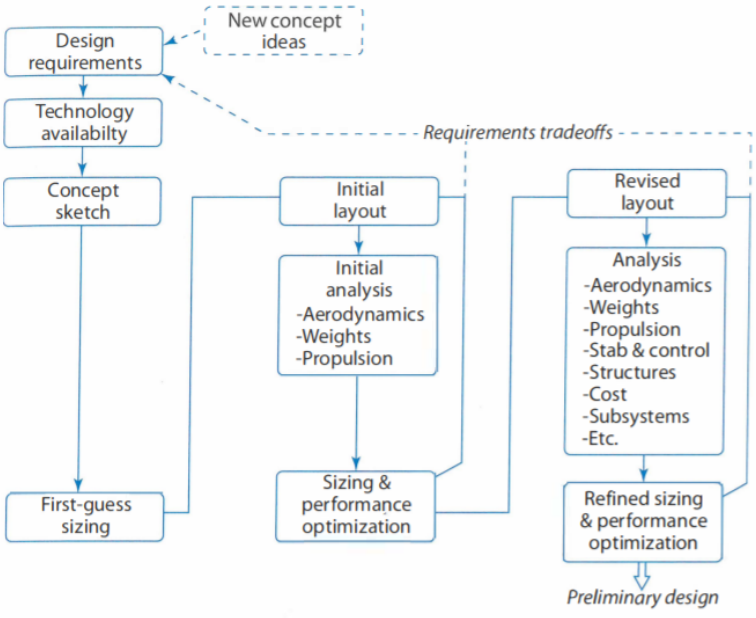


Figure 2.4: Typical loop of the conceptual design of air-/spacecraft. [42]

The main areas that influence the design process at this stage are aerodynamics, propulsion and mission performance. The concepts that result from the analysis need to meet the pre-selected specifications, and they need to do so in the ”closest-to-optimal” way. The aspects such as structures and control systems are not considered in detail, but still kept in mind. Quoting J.D. Anderson, ”no part of the design process is ever carried out in total vacuum unrelated to the other parts” [41]. Thus, many areas impact the evaluation of the competing concepts, but at this stage some can be more important than others.

The aspects of the configuration that are determined at this stage include, but are not limited to, the wing geometry (its planform, sweep, etc.), the wing’s position with respect to the fuselage, type and locations of horizontal and vertical stabilizers (if any), propulsion system, and the internal locations of the vehicle’s subsystems (crew, payload, equipment, etc.). The end product of this stage is a visualisation of the resulting configuration and arrangements in form of sketches or a low-level 3D geometry. The level of detail at this moment is not very high, and the final layout is not set in stone. Some flexibility is carried over to the next phase, preliminary design. [41], [42]

2.1.2. Preliminary design phase

By definition, the next stage of the design process, preliminary design, brings few and relatively small changes to the configuration. The most important questions need to have been answered at the end of conceptual design stage. If significant changes to the configuration are required as a result of the preliminary design analysis, it means that the conceptual design process was seriously flawed.

In this phase a significant refinement of the conceptual design needs to happen. Preliminary design brings about a significant increase in detail and understanding of the design through more advanced, more expensive and more time-consuming techniques such as wind-tunnel testing and extensive CFD simulations for its aerodynamics, or FEM and others for structural analysis. This is also the phase in which structural design and control system design substantially enter the process. That is, both of these areas are now critical to the design. At this point a more detailed mock-up is usually to be created. A process known as *lofting* is carried out. This is a precise mathematical definition of the external shape of the vehicle that ensures that all sections and components of the aircraft fit well together.

Preliminary design ends when the designers are confident that the design meets the requirements after a proper analysis is carried out, involving time and costs necessary to build the vehicle. At this stage the design can be considered precisely defined and the configuration can be *frozen*. Freezing of the design means that the other designers, responsible for detail design, can start their work without fear that it will be nullified by changes to the overall configuration. [41], [42]

2.1.3. Detail design phase

Once the design is frozen, and the project can proceed to full-scale development, the detail design phase can begin. It is at this stage that the actual components are designed for manufacturing. Until now, no single part has been designed to actually be built, only concepts with varying level of detail were created.

The top level decisions have already been made. This phase focuses on all the little details that make the vehicle eventually come to life. These details include things that were not even considered in the previous stages, such as the exact placement of cutouts in the ribs, the placement of holes for assembly, the arrangement of cables, pipes, actuators for landing gear, fuel, propulsion, and all the other systems. The ultimate goal of this stage is to prepare the vehicle for fabrication. A very high level of detail of the parts is required, so that they can be

sent to manufacturing. An important task of this process is production design, where specialists decide how exactly the components are going to be built and assembled. Some minor modifications can be allowed if it means the production is easier and/or cheaper, as long as their impact on the performance is not too big.

Whereas the two previous phases could be realized by relatively small teams, this phase is different. The amount of people necessary to carry out all the tasks in a realistic time is drastically increased and can reach even thousands.This is the stage that consumes the most resources, time and money. High-quality conceptual and preliminary design phases are needed to keep these costs to the minimum.

2.2. Novel approach

In the rapidly moving market the traditional approach has many limitations. Starting with the already mentioned compromise between freedom and complexity of the design [40], through the effect on the life-cycle cost [44], to the actual time it takes the product to reach the market [45].

Figure 2.5 shows how freedom decreases with increasing knowledge about the design. This creates a paradoxical situation where the more the designer knows about the concept, the less influence they can have over its final form. This problem originates already in the conceptual design phase.

Then, Figure 2.6 illustrates how the percent contribution to the entire life-cycle cost changes as the development process progresses, and how the opportunity to reduce the total cost decreases. It can be seen that the trend abides the so-called Pareto Principle, or the 80/20 Rule [46], which states that nearly 80% of the outcomes are an effect of 20% of the causes. This phenomenon was first observed by Vilfredo Pareto in his Cours d'Économie Politique, where he showed that 80% of the land in Italy was owned by 20% of the population [47], and later it was found that also many different natural occurrences exhibit similar distribution. Here, it is seen that 70% of the total costs are already determined at the stage of conceptual design, and the opportunity to reduce these costs significantly drops, whereas by itself this stage generates the least amount of cost, which can be as little as around 1% [40], [42], [48].

The development time is the time it takes from the start of the project to the product's introduction to the market. Generally, the risk of failure increases as the development time increases. This is due to the fact that an idea for a new product rises from an analysis of future needs. In the time it is being developed, it might easily become obsolete, and this is why it is essential to shorten the development time as much as possible [45]. Figure 2.7 visually compares how different approaches, namely concurrent engineering and front-loading, may influence the *time-to-market* of a product.

A closer look at the aforementioned issues leads to very straightforward conclusions. The conceptual design stage is unreasonably short, given how much influence it has over the next stages, and most importantly, over the final form of the design. Even though it consumes the smallest portion of the total project costs, it determines the biggest portion of the budget needed. The 80/20 Rule only highlights how important this design stage is for the success of the development program. Moreover, since this is the most flexible stage, it offers the most possibilities for design changes with relatively marginal effort. That is, the philosophy of concurrent engineering, or the front-loading, can most easily find their applications in the earlier design phases. It can be achieved by a wider implementation of the more modern design techniques that pull high-fidelity validation methods forward in the product development [49], leading to a representation of the development process closer to the one showed in Figure 2.8.

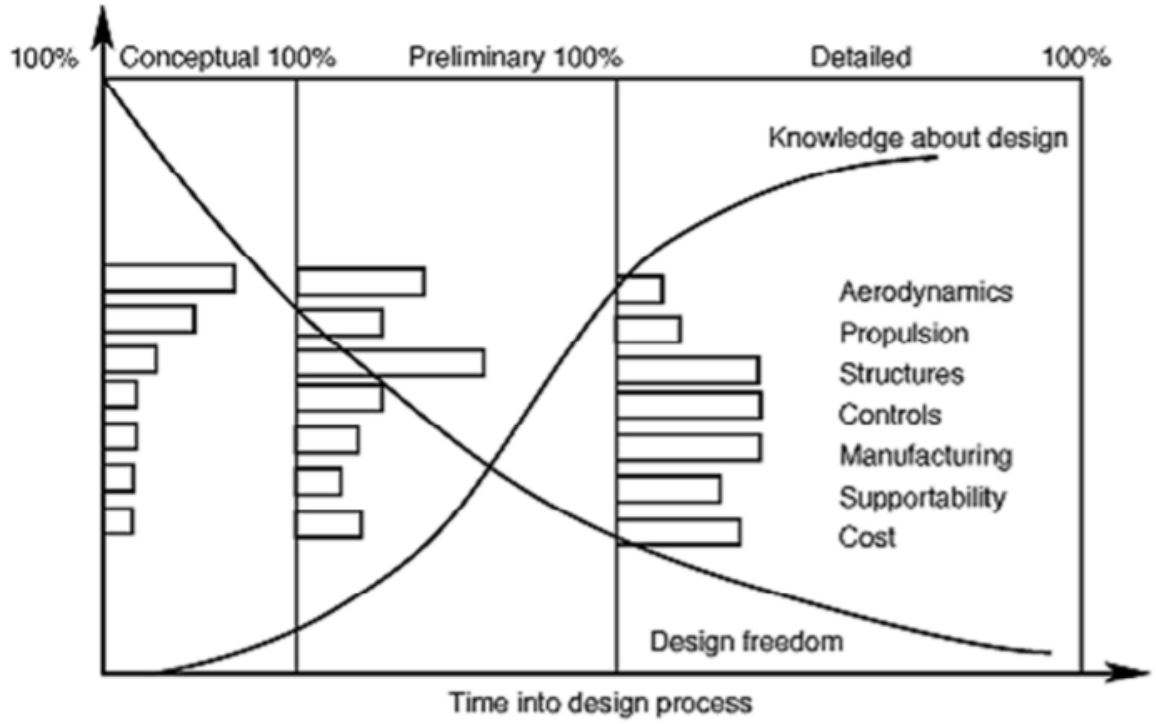


Figure 2.5: Re-distribution of the disciplines with curve symbolising knowledge about the design and design freedom. [50]

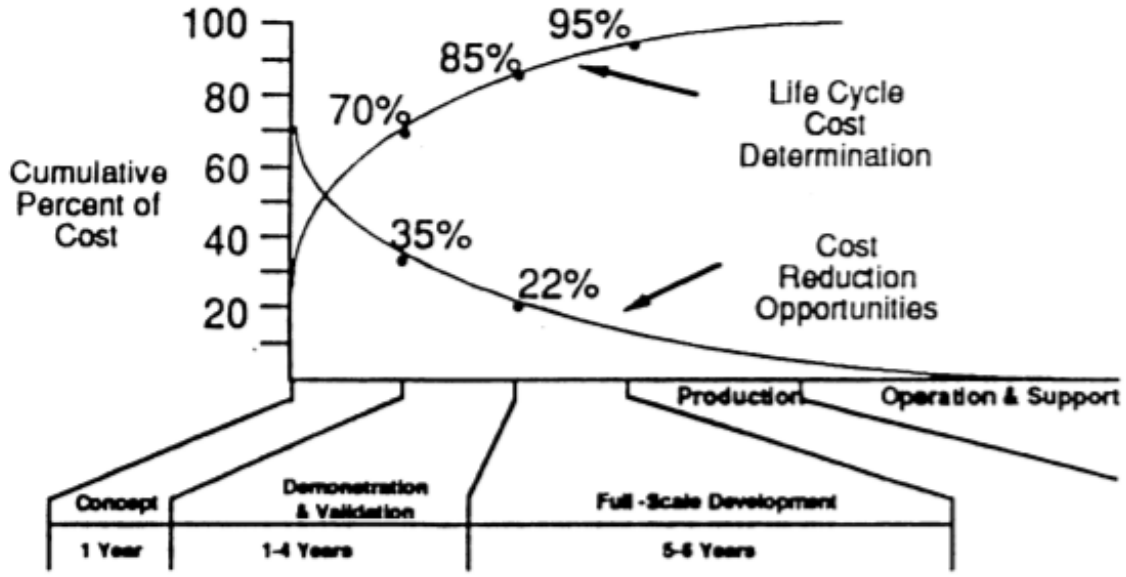


Figure 2.6: Cost of an aircraft as a product across its life cycle, with curves symbolising the reduction opportunities and cost determination. [44]

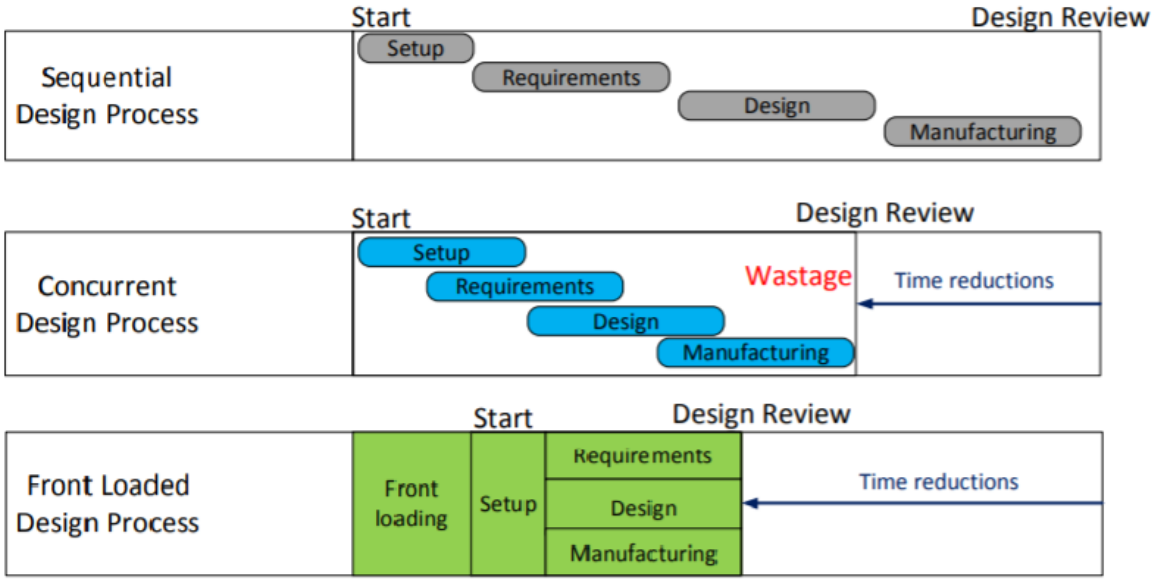


Figure 2.7: Time-to-market for various design strategies. Redesigned from [51]

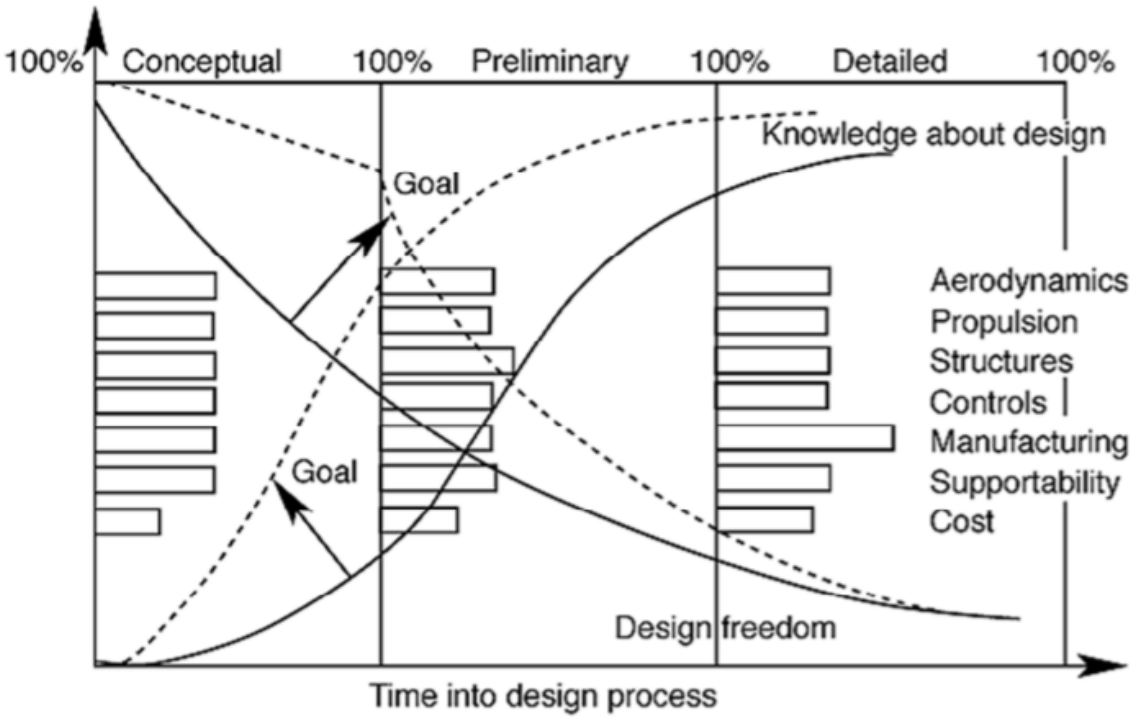


Figure 2.8: Goal re-distribution of the disciplines with curve symbolising knowledge about the design and design freedom using modern design techniques. [50]

2.2.1. Multidisciplinary Design Optimisation

The need for integration of more disciplines earlier in the design process while accounting for their interdependence is not a new observation. Up until the 1960s, the design teams would be responsible for specific disciplines and their work would simply be passed on to the next team. For example, the aerodynamicists would outline the external shape of the aircraft for the structural engineers to later fit the internal structure into the specified shape. Back then the goals of said teams were predominantly performance-based, e.g., minimised drag, or maximised speed [40]. However, whereas the purely sequential design process used to be true in the early days of aviation, it no longer holds. It slowly started changing after the invention of UNISURF, the first 3D CAD/CAE software created by the French engineer Pierre Bézier in 1968 for the automobile manufacturer Renault [52], which was a pioneering tool and the foundation of the next generations of CAD software to come. Another major influence on the change of this outlook was the migration from the performance-centered policy in the early 1970s, when other features like manufacturability, maintainability or operational costs and their influence on the life-cycle costs started being more appreciably considered [40]. The origins of what is now known as Multidisciplinary Design Optimisation started at structural design with Schmit in the 1960s [53], [54] and later Haftka [55], [56], who extended it to include more disciplines.

By the 1990s, now with more advanced CAD technology and experience gained during the *performance-related-design era*, the design objective was shifted to a more balanced one, where life-cycle costs are equally as important as the product's performance. The MDO methodology is now more widely used. A methodology that is capable of a significant reduction of the development time while also allowing for more design freedom for longer, and providing more knowledge about the design earlier in the process [40], [50]. The main idea behind MDO is to simultaneously incorporate various relevant disciplines into a design problem to find the optimum instead of sequentially optimising each discipline one by one. However, in the 30 or so years since the first application of MDO in conceptual and preliminary design phases, the results were relatively mediocre. There are two most probable reasons that could explain why [40]:

- the analysis tools used for specific disciplines in the MDO framework were of low fidelity, thus giving frequently poor and inaccurate results
- the computers of the time did not have the necessary computational power to handle problems of such complex and multi-leveled nature, limiting the number of variables to just 5-10, which is significantly too low to properly handle an MDO task

Giesing and Barthelemy in their paper from 1998 analysed industrial (not just aerospace related) applications of MDO, which is summarised in Figure 2.9 [57]. As can be seen, higher-fidelity methods could only be used at the level of trade studies. In order to use *full* MDO, the fidelity of the employed tools had to significantly drop. And this is in direct opposition to what MDO should be about, which is pushing as much knowledge about the project as much forward as possible. For the computers of the time it was next to impossible to increase the automation level of the high fidelity tools to make use of them in the MDO process. And even these tools by themselves, not as a part of MDO, required a long computational time for their analyses.

In 2007, Weck et al. published a paper where the current state-of-the-art of MDO was discussed [58]. Even though the application of the methods had been spreading and becoming more and more common, the implementation was still rather limited and generally only started at the stage of detail design. Still, the biggest factor restricting the wider application of MDO methods seemed to be the computational power necessary to solve such complex problems at the conceptual, or at least the preliminary design level.

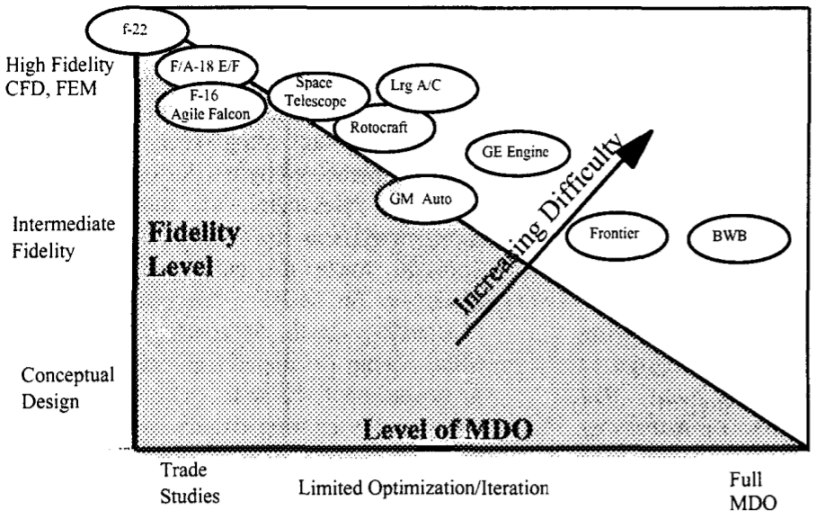


Figure 2.9: Distribution of design process fidelity and level of MDO [57]

Although huge progress has been made in aircraft design where the interactions between different disciplines are taken into consideration, what led to significant increase in their performance, the traditional sequential approach is still what is being used by the designers. Not a systematic MDO approach. While the computational power can still be partially found guilty for this limited use of MDO in the industry, there are other aspects that also need to be addressed. According to Hoogreef et al., some of the reasons are the intrinsic complexity of the mathematical foundations of MDO, or the lack of proper understanding of the available algorithms and architectures [59] - a problem that KBE could solve.

2.2.2. Knowledge Based Engineering

MDO is a very promising tool to help with the design process. However, this method can support the design framework only if the ideas are formalized in the design space defined by input parameters and design variables. A decomposition of the design problem, accompanied by search and analysis strategies are needed to create a working MDO system. But this kind of formulation is hard and time-consuming. And this is where the relatively new concept of Knowledge Based Engineering (KBE) can be extremely helpful [60].

KBE can be viewed as a combination of computer aided design, programming and artificial intelligence. It belongs to the group of so-called knowledge technologies. These are emerging computer-based tools that combine applications from different fields, such as psychology, philosophy, engineering, business studies or computer science. What all knowledge technologies have in common is that they identify, capture and represent the relevant knowledge from various disciplines, and manage it through embedded computer systems [61], [62].

Hence, KBE can be used to support the formulation of a working MDO system by a design formalization in terms of inputs and variables, optimization with an objective function and design constraints, and design analysis. KBE can be implemented in cases with complex geometry generation and manipulation by building blocks that contain optimization, and analysis modules. KBE is not used as a replacement of these modules, but rather to provide solutions to create elaborate systems that translate a given model into particular data sets required to run these modules in an automated way. Moreover, this concept can be used as an evolved CAD system, i.e., provide hands-off manipulation of a product's geometry with an optimizing algorithm that captures both configurational as well non-configurational information. The non-configurational information is usually what is not taken into account by traditional CAD systems, but is no less necessary to build a multidisciplinary system [60].

There are some similarities between CAD and KBE systems what sometimes leads to a misconception regarding what KBE really is, but there is one fundamental difference between the two. CAD systems are primarily a means to store a design in terms of its geometry, to give sufficient information for the product to be manufactured. KBE systems store it in terms of the knowledge about the rules governing a design that later lead, among others, to the creation of its geometry. This information can be the design standards, regulatory codes, manufacturing constraints, safety protocols, material characteristics, etc. Simply put, CAD systems are focused on what the design looks like, while KBE systems are focused on what leads to the design, how it is created and why [61]. Geometry is simply an input that can be passed to a KBE system, or an output that can be generated by it. Geometry-related aspects of KBE can constitute less than 50% of the entire application [63]. A well-built KBE application captures enough knowledge to create a software capable of generating all the necessary information about the product. Its geometry and drawings, and non-geometry related data, such as bills of materials, cost proposals, tool design [60].

A development of a KBE application is also not an easy task by itself. All the knowledge needs to be made explicit so that it can be captured by a programming language to create such an application. Time and resources need to be invested to make it happen. Once such an application is available it makes a life of an engineer significantly easier, ridding them of all the boring and repetitive tasks. Therefore, a KBE application should only be created when it is needed to quickly generate many different variants and configurations of a product, and/or when it is required to vastly explore its design space. A KBE application can be used to automate certain tasks so that an engineer has more time to do the quality work. The advantages of such an approach are most notable in the conceptual design phase, where topology variations can be big and frequent. It can also significantly help at the stage of detail design once the design is frozen. The application could relieve repetitive manual rework associated with small last-minute changes. That way the designers can keep more freedom even in the latter stages of the process.

On top of all that, a KBE application can fit nicely into an MDO process offering a great deal of support. It can be used to generate parametric models with adaptive modifications to allow changes in configuration without limiting the designer's freedom. The topology can be automatically translated into models compatible with different analysis tools, which can be integrated into the process. Hence, the engineer does not have to spend their precious time on laborious model generation. Furthermore, the optimizer does not need to take care of spatial integration constraints since they are inherently guaranteed by models correctly defined by the KBE application. In short, KBE generative models can be the solutions that allow the development of MDO systems that do not need to trade multidisciplinarity for analysis fidelity. [60]

Currently, there are several KBE systems available on the market that can be used for design, analysis and other engineering purposes. Some of the most popular systems are listed below.

- ParaPy from ParaPy [64]
- Adaptive Modeling Language from TechnoSoft, Inc. [65]
- Enventive from Enventive Engineering, Inc. [66]
- Gendl (open-source) [67]
- Genworks GDL from Genworks International [68]
- Pacelab Suite from PACE Aerospace Engineering and Information Technology [69]
- PCPACK6 from Tacit Connexions [70]

3

Aerodynamic design and analysis

Aerodynamic design is the backbone of anything that flies, spacecraft in particular. Different types of spacecraft have varying requirements for their aerodynamics. Rockets need to be designed in such a way so as to minimize their drag on the climb to lowest possible energy loss, as the atmosphere is only an obstacle on their way to the orbit. Spaceplanes, on the other hand, use the atmosphere to their advantage to create aerodynamic lift as well as potentially use it for their propulsion system. It means that these two kinds of vehicles will experience different aerodynamic loads acting on them. Rockets need to handle primarily longitudinal loads, and spaceplanes additionally experience large transverse stresses. A big portion of a spaceplane's mission takes part in the atmosphere, in conditions that vary significantly as the vehicle climbs. It makes the aerodynamic design of such a spacecraft extremely complex and the need for accurate aerodynamic analysis is of paramount importance. However, as discussed in **Chapter 2**, the accuracy of the tools used sometimes needs to be traded off for the time it takes for the results to be ready and available to the engineers. A brief introduction to aerodynamic design is offered in **Section 3.1**. Followed by **Section 3.2**, which discusses different methods, techniques and tools designers can use to analyse their concepts' aerodynamics.

3.1. Aerodynamic design

The aerodynamic design is an intrinsic exercise belonging to the entire aircraft design. It is focused primarily on the shape of the vehicle and layout of its lifting surfaces. The external shape of the aircraft is the most important consideration in the aerodynamic design, but the engineers cannot forget about other important factors that influence it, such as its structural integrity, weight, propulsion system among others. All of which are fully dependent on the actual requirements of the design. This section is focused on the role of aerodynamics in the design process.

There are four most important aspects that an aerodynamic designer needs to address [71]:

- How to create the necessary aerodynamic force to keep the aircraft in the air, i.e., lift?
- How much propulsive force is required to keep the vehicle moving? This is influenced by how much air resistance acts on the aircraft flying through it, i.e., drag.
- How do the forces acting on the vehicle and their distributions around it change during flight? In other words, how to make the vehicle stable and controllable in the air?
- What are the loads the vehicle experiences during flight and how do they change the shape of the aircraft?

Only the first three aspects are covered in this section, treating the last one, aeroelasticity, as a related, yet separate discipline.

General aircraft design starts with the definition of the mission requirements for the vehicle. For an airliner mission these could be the range, the amount of payload, or its design speed. For a fighter jet it could be the maximum speed, turn rate, or the amount of payload such as the ammunition. For an unconventional vehicle such as a spaceplane, these requirements could be the maximum altitude it is to reach, whether or not it is to enter an orbit, the amount of payload it is to deliver to space, etc. These requirements, along with the performance, cost and airworthiness goals, lead to the initial sizing, most often conducted using handbook and empirical methods which are the subject of Section 3.2.1.

Aircraft performance is a denomination that is used to represent the capability of an aircraft to realize actions and feats that would make it useful to accomplish certain goals [72]. There are many factors that define an aircraft's performance, e.g., the take-off and landing distances, the amount of payload it is able to carry, its top speed, rate of climb, range, maneuverability, but also its turn rate, turn radius, etc. All these metrics depend on the airframe and its powerplant specifications. The aerodynamic characteristics dictate the power and thrust requirements at the expected flight conditions. These characteristics are derived from the aerodynamic forces acting on the airframe [71]. In order to properly assess an aircraft's performance, an engineer needs to know these forces inside the aircraft's flight envelope. This aerodynamic analysis can be carried out using a plethora of different methods with varying level of fidelity. The following section, Section 3.2, talks about some of these methods in more detail.

The major task in aerodynamic design is the definition of the shape of the vehicle so that it is able to meet the performance requirements and demands inside its flight envelope. The extremes of the flight envelope define the load cases that are crucial for structural dimensioning. Furthermore, the above-mentioned stability and control need also be analysed. Generation of aerodynamic data is what allows for integration with these and other disciplines, and for diagnosis of possible issues when performance requirements are not achieved [71]. Figure 3.1 outlines the overall tasks and results of aerodynamic design and analysis.

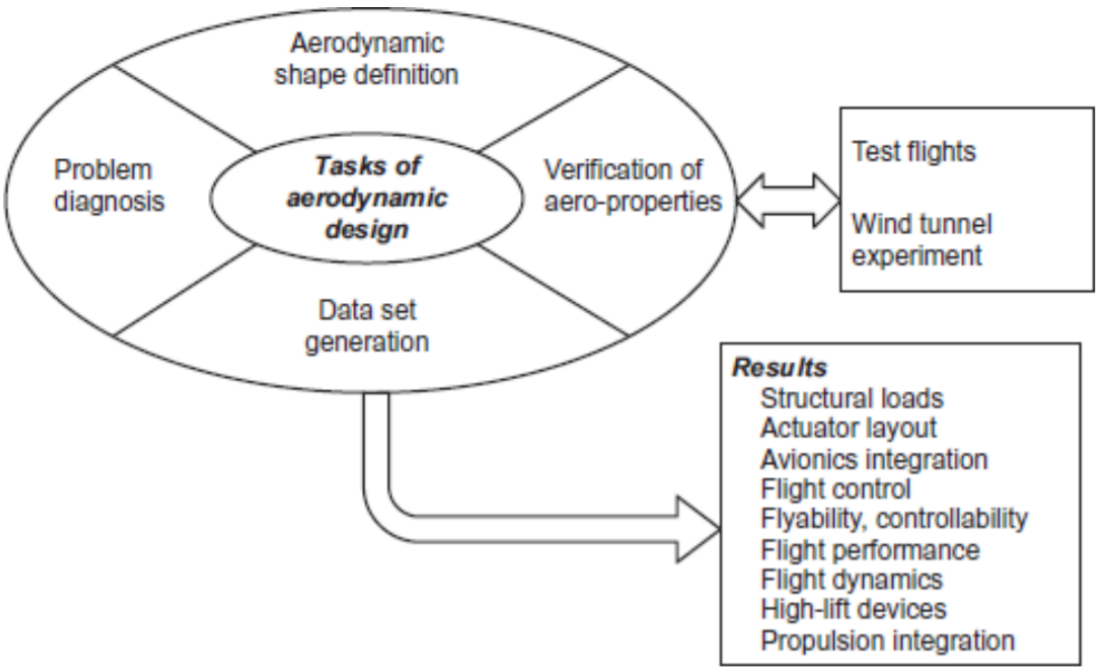


Figure 3.1: Tasks and results of aerodynamic design and analysis. [71]

What is understood by the tasks presented in Figure 3.1 is explained in some more detail below:

1. Aerodynamic shape definition:
    - Airfoil selection, wing lofting, empennage configuration, fuselage shape, etc.
    - External shape definition of the airframe to meet aerodynamic demands for lift, drag, loads, flyability, etc. through sizing and positioning of its components taking into consideration powertrain integration, structures, volumetric requirements, etc.
2. Data set generation:
    - Prediction of forces and moments acting on the airframe inside the aircraft's flight envelope for structural sizing, aeroelastic analysis, etc.
    - Establishment of flight cases that determine the forces, moments and their derivatives for stability and control analysis, and sizing of the control surfaces
3. Problem diagnosis:
    - Finding and solving issues discovered in wind-tunnel or flight testing such as failure to meet performance goals, poor handling characteristics, problems with stability, etc.

3.2. Aerodynamic analysis

Aerodynamic analysis, the subject of this section, is a sub-task of the aerodynamic design process. It is employed to calculate aerodynamic forces acting on the proposed vehicle and verify whether the design requirements are met. Many different tools with varying level of fidelity can be used for the process. The methods discussed in this section are presented in the order of growing fidelity:

- **Section 3.2.1** discusses lowest-order methods such as textbook and empirical methods
- **Section 3.2.2** talks about more advanced numerical methods, which are various linear solution methods
- **Section 3.2.3** offers a brief overview of Computational Fluid Dynamics (CFD) methods which can go from low-/medium-fidelity up to high-fidelity depending on the assumptions and actual modeling used
- **Section 3.2.5** briefly discusses wind-tunnel testing which can be employed for the aerodynamic analysis as a verification tool to the engineers' calculations, or to identify areas for possible improvement in the aerodynamic design

3.2.1. Handbook and empirical methods

Handbook methods, such as the ones presented in the classic textbook of Raymer on conceptual design [42] or Roskam's Airplane Design series [73], are the lowest-level methods of calculations used mostly for sizing of the baseline configuration. They can provide useful information on weights, wing geometry, and propulsion from the very basic mission requirements through explicit mathematical formulas and equations. Both mentioned texts have been automated into workable software such as RDS [74] and Advanced Aircraft Analysis (AAA) [75] respectively. Such methods are usually employed to synthesize the first concept, an initial sizing of the configuration and early estimations of the performance level. At the level these methods are used even the proposed mission itself can be modified if the considered concepts call into question its feasibility. Then the design process takes iterations until a mission profile and a matching design are reached that can be believed to gain the wanted market share with adequate investment return [71].

On the same level of fidelity are empirical methods. Such methods employ the correlation between test-data and theoretical concepts to find prediction equations. An example of such methods may be the USAF Stability and Control DATCOM [76] which is empirical in nature. It is a collection of equations, algorithms and rules based on physics and data of existing aircraft. The compendium has also been digitized into Digital DATCOM which is self-sufficient and is able to provide estimations of weights, inertia, forces and moments for all flight conditions and compute static and dynamic stability and control characteristics for any flight condition and configuration (with certain limitations). It can, however, be used in tandem with other methods. Another program that implements such methods is Engineering Sciences Data Unit (ESDU) [77]. [78], [71]

3.2.2. Linear solution methods

This section introduces some computational methods that are of higher order than the textbook and empirical methods discussed in the previous section. The subject of this section are a selection of methods based on the linear potential flow theory such as Lifting-line theory, Lifting-surface theory and vortex lattice method, and Panel methods. The mathematical formulations are outside of the scope of this report, thus only the basic principles behind these methods, their assumptions and limitations are presented.

3.2.2.1. Lifting-line theory

This section offers a short introduction to the principles of lifting-line theory (LLT). This whole section is based on J. D. Anderson's *Fundamentals of Aerodynamics* [79].

The lifting-line theory is a mathematical model used to calculate a finite wing's basic aerodynamic properties. Even though the method was developed in the early XX century, independently by Frederick W. Lanchester in 1907 [80] and Ludwig Prandtl in 1918 [81], it is still in use in preliminary aircraft design. However, its applicability is limited to straight wings with moderate to high aspect ratios.

The basic principle of the method is modeling of the lifting surface, such as a wing, as a vortex filament of some strength, bound to a location fixed in a flow, contrary to a free vortex, which travels with the fluid flow. This filament spans along the entire wingspan and is placed at the aerodynamic centers of the airfoils forming the wing. Given Helmholtz's theorem, a vortex filament cannot end in the fluid, hence it is assumed to continue as two free vortices downstream from the wingtips to infinity, one from each wingtip. Due to the shape of this vortex model it is known as a horseshoe vortex. This is visualised in Figure 3.2.

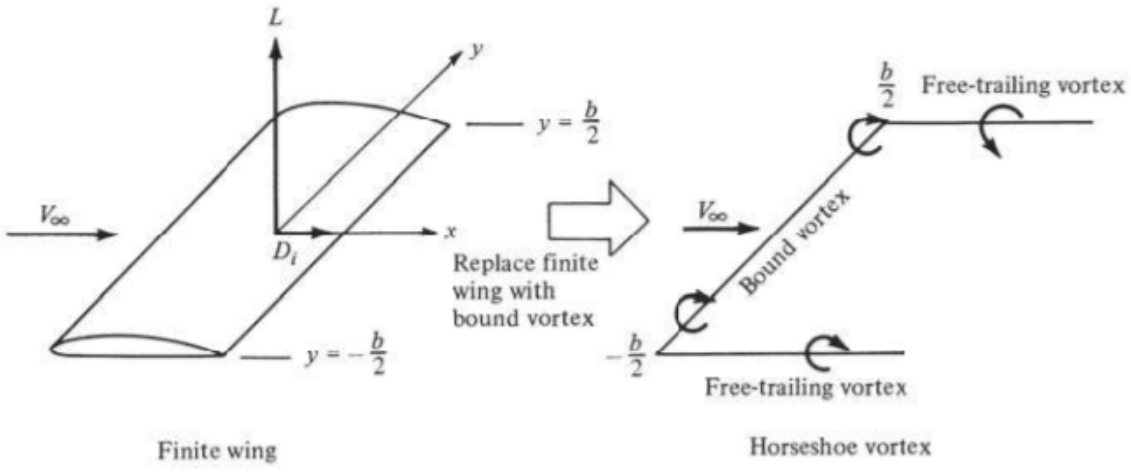


Figure 3.2: A finite wing is replaced with a bound horseshoe vortex. [79]

However, such a single horseshoe vortex is unable to realistically model the downwash distribution of a finite wing since for the model the downwash at wingtips approaches infinity. To solve this, the wing is represented by superposition of a number of such horseshoe vortices with varying length of the bound vortex. Each coincident along a single line - the lifting line (Figure 3.3). Hence, lifting-line theory. If an infinite number of horseshoe vortices is placed along the lifting line, the vortex distribution is continuous and a so-called vortex-sheet is produced downstream of the wing. From this sheet aerodynamic properties of the wing can be calculated, such as lift distribution, lift force, or induced drag. The main limitations of the method is that it does not take into account compressibility, viscosity or unsteadiness of the flow.

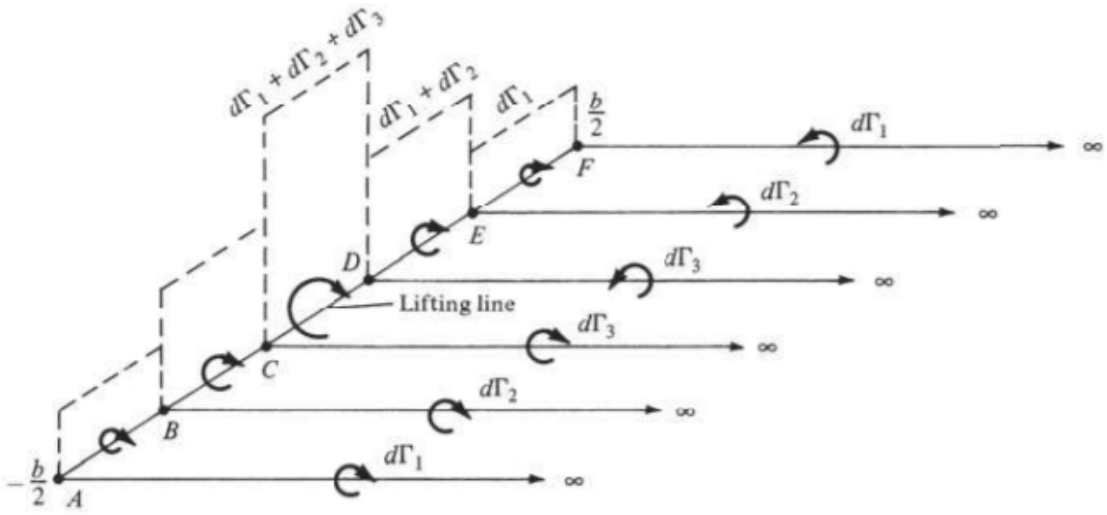


Figure 3.3: Lifting line created from superposition of horseshoe vortices. [79]

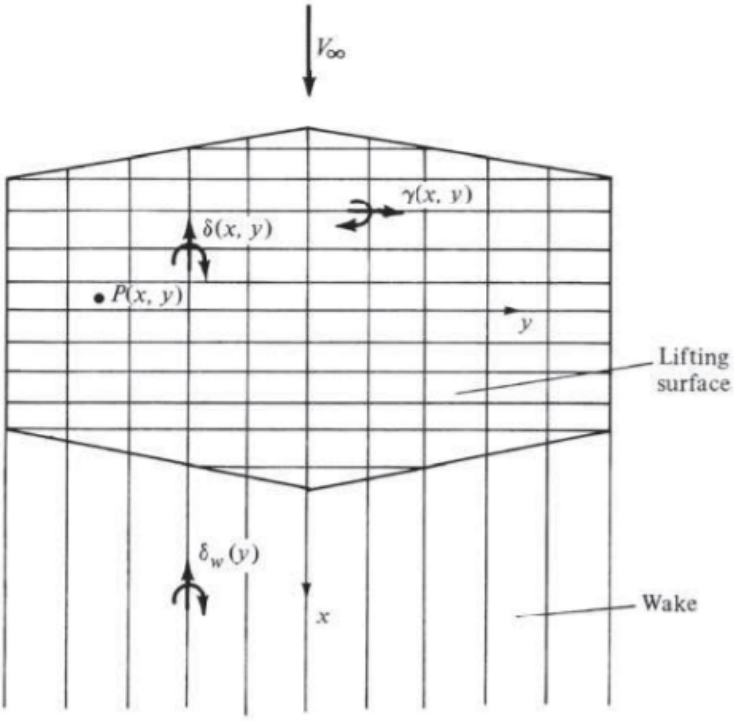


Figure 3.4: Lifting surface created from superposition of lifting lines. [79]

3.2.2.2. Lifting-surface theory and vortex lattice method

There are two similar methods that are, in a way, evolutions of lifting-line theory. They are lifting-surface theory (LST) and vortex lattice method (VLM), and they are the subject of this section, which is again based primarily on J. D. Anderson's *Fundamentals of Aerodynamics* [79].

As mentioned above, lifting-line theory is limited to wings with no sweep and at least a moderate aspect ratio (as a rule-of-thumb at least 4). To overcome this limitation and make the method applicable to other wing geometries an infinite number of lifting lines is distributed along the chord, what yields a vortex sheet. Trailing vortices from all these lifting lines form another vortex sheet. The strength of the vortices is dependent on the x- and y-coordinates on the wing. These two vortex sheets form a lifting surface, hence, lifting-surface theory. From the trailing edge downstream to infinity a wake vortex sheet is produced. A normal velocity is induced by both the lifting surface and the wake vortex sheet following the Biot-Savart law. The boundary condition of tangency of the flow gives that the sum of this induced normal velocity and the normal component of the freestream, which is the result of downwash induced at the wingtips, needs to be zero. Then, the aerodynamic properties of the wing can be calculated. The geometrical limitation of lifting-surface theory is that there is no thickness of the wing, i.e., thin airfoils are assumed.

Just like LST, VLM models a lifting surface as a thin vortex sheet and neglects the wing's thickness. It takes, however, a simpler approach to the concept. The lifting surface is divided into a number of quadrilateral panels with a horseshoe vortex placed on the quarter-chord of each of them and a control point on the panel's centerline at a half-chord distance behind the horseshoe vortex. The velocity induced by a single panel at any point can be calculated from the Biot-Savart law by treating each vortex filament separately. The flow tangency condition, applied at all the control points over the wing, produces a system of algebraic equations to be solved for the unknown strength of each horseshoe vortex. Consequently, resulting lift distribution, lift force and induced drag can be calculated. The same limitations as for lifting-line theory are also true for both LST and VLM.

Whereas strictly VLM is applied to lifting surfaces, similar reasoning, using sources and sinks, can be extended to model fuselages and nacelles as slender bodies to model the entire aircraft.

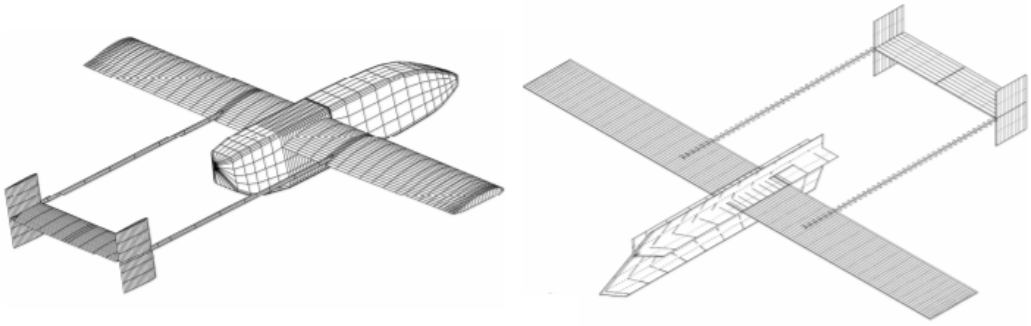


Figure 3.5: Comparison of a panel model to be solved with a panel code (left) and VLM (right). [82]

3.2.2.3. Panel methods

An even more advanced family of methods to solve potential flow equations are panel methods. Contrary to the ones discussed above, panel methods are able to model completely arbitrary geometries. A comparison of panel models to be used by a panel code and a VLM software is shown in Figure 3.5. Panel methods accept models built from tri- or quadrilateral elements. Panel codes' numerical schemes are based on the superposition of surface distribu-

tions of basic analytic solutions to the Prandtl-Glauert equation (sources, sinks, doublets and vorticity singularities) over small panels that approximate the aircraft external shape. Panel methods solve the Prandtl-Glauert equation for linear, inviscid and irrotational flow at subsonic, transonic, supersonic, or even hypersonic freestream velocities. There are lower-order and higher-order panel methods depending on the *under-the-hood* calculations they perform. That is, lower-order methods employ constant strength singularity distribution over all panels, which are usually flat. Higher-order codes use singularity distributions that are of higher order than constant, e.g., linear or quadratic, and the panels can also be curved. The higher-order panel methods often contain some approximations to account for physics neglected by the Prandtl-Glauert equation, such as flow viscosity, the boundary layer, or even flow separation off highly swept wings. The latter is usually modeled with wake panels that emanate from wing leading and trailing edges, whose correct position and shape is later solved with iterative relaxation techniques. [83]

3.2.3. Computational fluid dynamics

The most advanced and complex methods of fluid flow calculations are computational schemes known as Computational Fluid Dynamics (CFD). Contrary to the methods discussed so far, CFD is used to calculate the entire flow field in a domain, and not just forces and moments acting on the aircraft. CFD is the science of numerically solving systems of partial differential equations (PDEs) that describe flows of fluids. Inside the CFD family there are numerous different models with varying level of complexity to solve these equations. The models use different assumptions and simplifications to reduce the complexity of problems at hand and lower the computational cost associated with their solution. However, an engineer must be aware of the limitations these simplifications introduce. Some of the most popular models and schemes are the topic of this section. Figure 3.6 presents the hierarchy of the models treated in this section.

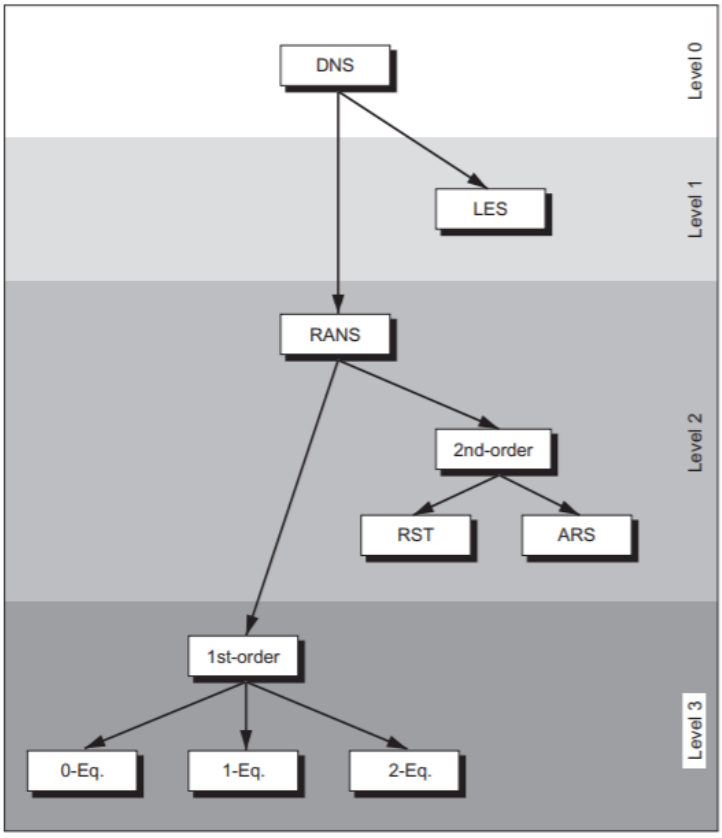


Figure 3.6: Hierarchy of turbulence models. [84]

3.2.3.1. Direct numerical simulation

Direct numerical simulation is the most accurate method of numerically solving the Navier-Stokes (NS) equations, that is, without any turbulence models. The scheme does not offer an exact solution since discretisation steps are required and possible numerical errors may appear. The method, however, does not average, approximate nor neglect any terms in the NS equations. Its objective is to simulate the entire range of turbulent fluctuations at all relevant physical scales. The obtained flow field is analogous to a single execution of a flow or an experiment realised in a laboratory. The extremely high accuracy associated with the method has a major drawback, which is a very high computational cost. So high, that DNS simulations for realistic Reynolds numbers are currently out of reach for even the world's best super-computers. [85], [86]

3.2.3.2. Large-eddy simulation

The smaller scale motions are normally much less energetic than the large scale ones. Due to the size and strength of large scale motions they transport significantly more of the conserved properties, i.e., mass, momentum, etc. Because of this, large-eddy simulation (LES) can overcome the issue of the excessive computational cost and lower it by resolving only large eddies and modeling the smaller scales with a turbulence model. Additionally, large eddies are most affected by the problem's geometry, with the smaller scales being relatively more universal, what makes them fit to be described by sub-grid scale models [87]. Even though this approach allows for big reduction in the computational cost of fluid flow simulations, it still remains too high for LES to be applied in conceptual or even preliminary design. Currently, this method is usually only applied to solving flows around some crucial parts and not around the entire geometry. [85]

3.2.3.3. Reynolds-averaged Navier-Stokes

Although the implementation of LES brings down the computational power necessary to solve the NS equations and simulate the flow field, the cost is still too high to employ this method in preliminary design phase. Particularly for high Reynolds numbers, such as can be found around full scale aircraft or spacecraft. Another simplification of the solution of the flow field is by solving for the Reynolds-averaged Navier Stokes (RANS) equations which model all scales. This approach allows for an even further reduction in computational demands and makes the method applicable to practical purposes of aircraft design.

The (steady) RANS equations are derived by decomposing the flow variables into their mean and fluctuating parts. This is the so-called *Reynolds decomposition* (Equation 3.1 shows an example for velocity components).

$$v_i = \bar{v}_i + v_i' \tag{3.1}$$

Then, the governing equations are solved for the mean values. There are three different ways of the *Reynolds averaging*:

- Time averaging (Equation 3.2) - applicable to stationary turbulence (statically steady turbulence); the mean value then only varies in space and does not change in time
- Spatial averaging (Equation 3.3) - applicable to homogeneous turbulence; the mean value is uniform in space but does vary in time
- Ensemble averaging (Equation 3.4) - applicable to general turbulence; the mean value is still time- and space-dependent

$$\bar{v}_i = \lim_{T\to\infty} \frac{1}{T} \int_t^{t+T} v_i dt \tag{3.2}$$

$$\bar{v}_i = \lim_{\Omega\to\infty} \frac{1}{\Omega} \int_\Omega v_i d\Omega \tag{3.3}$$

$$\bar{v}_i = \lim_{N\to\infty} \frac{1}{N} \sum_{m=1}^{N} v_i \tag{3.4}$$

The fluctuating part disappears when Reynolds-averaged. However, due to the non-linearity of the full NS equations the velocity fluctuations are still present in the RANS equations in the non-linear term from convective acceleration even after the process of Reynolds-averaging. This term is known as the Reynolds stresses and needs to be replaced by semi-empirical turbulence models to close RANS equations and provide the so-called turbulent viscosity. Various turbulence models exist to solve this closure problem, and are briefly discussed below. [84]

Closure models

As mentioned above, RANS equations do not resolve the turbulence but rather model it. After the Reynolds-averaging, a closure problem emerges, meaning that without proper models the velocity fluctuation terms would still be present in the equations. There is a wide variety of models, known as closures, based on empirical data that can be applied to resolve this issue.

The closure models provide a value for the Reynolds stresses either directly, through Reynolds stress models (RSM), or indirectly. The indirect approach follows from the Boussinesq assumption that relates the Reynolds stresses to the average flow parameters; the Reynolds stresses scale with the mean strain rate [88]. As a result the effect of turbulence can be replaced with increased flow viscosity. Thus, the viscosity is divided into two components, one is the dynamic viscosity and the other is the turbulent viscosity. The latter can be computed with various turbulence models [89].

The first-order, indirect, methods are also known as linear eddy-viscosity models. They need fewer equations than the higher-order methods to treat the closure problem. Hence, the computational power needed for simulations is reduced. These models are more widely used in practice [90]. They can be grouped as zero-, one- and multiple-equation models, depending on the amount of transport equations they use to model the Reynolds stresses. The first group, the zero-equation models cannot take history of the flow, like convection and diffusion of turbulent energy, into account due to them being fully empirical, what causes problems with the prediction of separation. The two higher-level groups of models do account for history effects. Some of the more popular are:

- Spalart-Allmaras model [91] - a one-equation model; extensively validated and usually able to accurately predict aerodynamic forces up to moderate angles of attack. The main issues are caused by its one-equation formulation. Time and length scales of turbulence are not defined as well as in higher-level models, what leads to problems with solving shear flows, under-prediction of separation and decaying turbulence.
- k-ϵ [92] - a two-equation model; the model predicts turbulence with two PDEs for turbulence kinetic energy (k) and turbulent dissipation rate (ϵ); it is relatively reliable for free-shear flows (when small pressure gradients are present). However, it should not be applied to problems where adverse pressure gradient, big separation or complex flows are expected.

- k-ω [93] - a two-equation model; the model predicts turbulence with two PDEs for turbulence kinetic energy (k) and specific rate of dissipation of k (ω); it is best suited for near-wall treatment, i.e., when good accuracy of the boundary layer is required, and for external aerodynamics. It is also good at resolving regions with adverse pressure gradient. It is sensitive to inlet freestream turbulence conditions. Most commonly used as k-ω SST (shear stress transport) which is a combination of the k-ω and k-ϵ models. This, however, can cause problems in the stagnation regions or regions with big flow acceleration due to the model producing high turbulence levels in areas with high normal strain.

Eddy-viscosity models can be expected to have a reasonable accuracy in attached-boundary-layer flows considering that only one component of the Reynolds stress tensor is significant. However, more complicated flows can pose significant problems, and the validity of this approach may be dubious. Such models are unable to properly capture anisotropic normal stresses and their effects. Moreover, they cannot correctly represent the impact of additional strains and body forces on turbulence. As shown in Figure 3.6, the direct methods such as Reynolds-stress model (RSM) and algebraic Reynolds-stress (ARS), are the second-order closures. These methods include these effects exactly. These schemes are more flexible but also more complex and more computationally expensive. They require additional models for a few turbulence processes, such as pressure-strain correlations, turbulent diffusion of Reynolds stresses, dissipation. [71]

3.2.3.4. Thin-layer Navier-Stokes

Between the viscous Navier-Stokes equations and the inviscid Euler equations (discussed in Euler equations) are the thin-layer Navier-Stokes (TLNS) equations. The philosophy behind this approach is that for sufficiently high Reynolds numbers the viscous effects parallel to the flow are negligible. The pressure is allowed to change across the boundary layer (in the direction perpendicular to the flow), these equations can simulate separation of the flow, given that it is not too big. Eliminating the viscous derivatives in the flow-direction from the NS equations allows for a big reduction in computational power required to simulate the flows.

3.2.3.5. Boundary-layer equations

Similarly to TLNS equations, boundary-layer (BL) equations follow from the assumption that at high enough Reynolds numbers, the viscous effects outside of a small region of the flow close to the wall (the boundary layer) can be neglected. BL equations are derived by dropping terms of smaller orders of magnitude from the full NS equations. The low order of magnitude makes them of *little importance* to the final result. In this way, the created set of equations is reduced and more manageable. One of the major simplifications is neglecting the viscous diffusion parallel to the flow. Depending on other assumptions, such as incompressibility, steadiness, or two-dimensionality of the flow, the equations take different forms. The major assumption of the model is that the pressure remains constant across then boundary layer, which is the biggest difference between BL equations and TLNS equations. It means that the BL equations cannot see flow separation. Just like other simplifications to the full NS equations, BL equations need additional closure relations, which are analogous to the turbulence models discussed in **Section 3.2.3.3**.

The location of the edge of the boundary layer is, obviously, a priori unknown, and so is the velocity of the flow at this edge. The coupling between viscous and inviscid layers is obtained though an iterative process called a viscous-inviscid interaction method.

3.2.3.6. Euler equations

The Euler equations follow from yet another simplification to the flow. They represent the mass conservation equation and a particular case of Navier-Stokes equations with no viscosity and

no thermal conductivity. This approximation holds relatively accurate in flows where viscosity effects are usually very small, like flows far away from the wall. Because the fluid viscosity is neglected, slip at solid boundaries is allowed. At high-speed flows, where the Reynolds number is very high, and the importance of the viscous and turbulent effects is limited to a small region near the walls. Hence, the Euler equations are frequently used to analyse high Mach number compressible flows. Apart from the simplification of the fluid flow equations themselves, the Euler equations have another major advantage that allows for a reduction in the computational power required for simulations. Due to the lack of the boundary layer caused by neglected viscosity, the grid can be significantly coarser and can be unstructured. Both of these aspects are very relevant in automation of the analysis process. [85]

3.2.4. Domain discretisation

In order to apply any of the models discussed above, the computational domain needs to be discretised. The process of discretisation of a computational domain is called meshing or gridding. A mesh, or a grid, divides the geometry into many inter-connected elements. This way the domain is no longer continuous but is represented by numerous such elements. The CFD code can then solve adequate equations in each element to effectively simulate what happens in the full domain. The basic terminology (Figure 3.7) used in meshing is:

- Cell - control volume into which the domain is divided
- Face - cell boundary
- Edge - face boundary
- Node/vertex - grid point
- Cell centre
- Zone - a group of nodes, faces, and/or cells

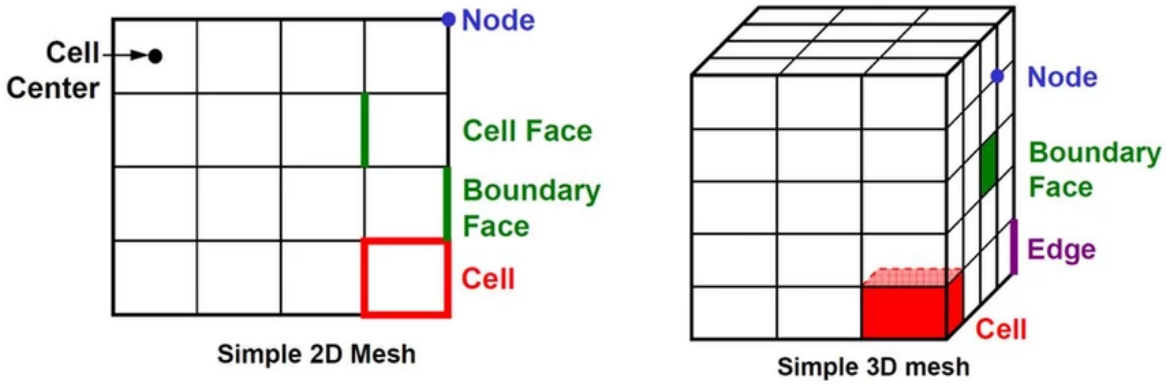


Figure 3.7: Basic mesh terminology. [94]

Depending on whether the problem is two- or three-dimensional, the grids are either surface grids or volume grids respectively. Surface grids are usually built up from triangles and/or quadrilaterals, and volume grids - from tetrahedra, hexahedra, prisms, and pyramids.

The aim of mesh generation in fluid flow problems is to cover the space between the boundary of the flow domain and the body of interest, e.g., the airframe. The grid should resolve the geometric characteristics of the boundaries. Generally, it should be fine close to the body, and can be coarser farther away from it, where velocities are expected to be much closer to the free stream. As mentioned in **Section 3.2.3.6**, when using the Euler equations to solve the flow field, the mesh can be more isotropic, for the lack of the boundary layer. However,

what is true for models both with and without viscosity, the grid should be very fine around the locations where large gradients of flow properties are expected, e.g., around where a shockwave would appear. Typically grids can be divided into structured, unstructured, mixed, or multi-block structured.

In a structured grid each node is uniquely identified by i, j, k indices and their Cartesian coordinates $x_{i,j,k}$, $y_{i,j,k}$, $z_{i,j,k}$. In 2D structured grid consist of quadrilaterals, and in 3D - of hexahedra. An unstructured grid's cells and vertices do not have a specific ordering. Neighbouring cells and nodes cannot be recognized by their indices directly. On top of the shapes that build up structured grids, unstructured grids can be made of triangles (in 2D) and tetrahedra, prisms, and pyramids (in 3D). The domain can be discretised by structured grids in some areas (e.g., close to the body) and by unstructured grids in others (e.g., far field). In such a case, the mesh is called a mixed/hybrid grid. In a multi-block grid the domain is first divided into a low number of regions, and then each of the regions is further split into cells by means of a structured mesh. This allows for a decomposition of a complex geometry into less complex regions that are more easily meshable in a structured way.

The mesh's quality has a great impact on the accuracy of the solution, particularly for unstructured grids. The basic requirement for a grid is for it to have non-overlapping cells with positive volumes. On top of that, it should be sufficiently fine to be able to resolve the relevant flow features. The two most simply quantifiable mesh characteristics are the size and shape of its cells, and the change in thereof between adjacent cells. The discretisation error grows proportionally as a cell's shape diverges from its *regular* shape. Optimally, the cells should be nearly identical in shape and size (rotation is allowed). However, this would restrict the size of the cells in the whole domain (including the outer boundary), what would greatly increase the computational costs. However, not all flow models are equally sensitive to cell distortions. For example, spatially second-order PDEs are more sensitive to this than first-order PDEs, e.g., the Euler equations. [71]

3.2.5. Wind-tunnel testing

Back when the CFD simulations started being more widely applied, and the available computational power drastically escalated, it was thought that CFD would forever replace wind-tunnel experimental simulations. However, as of today, CFD has not substituted experimental work and wind-tunnel testing is still an integral part of the aircraft design process. This section outlines wind-tunnel testing in brief - what it is, and why and how it is employed.

The most important capability that makes experimental simulations still an indispensable tool is design verification in terms of its performance and/or handling qualities. A good wind-tunnel testing campaign provides results that can be used to verify the computationally generated aerodynamic data. So many different flow models, each with numerous adjustable parameters, make it hard to be confident in the results without verification. And here wind-tunnel analysis comes in very handy. Experimental and numerical simulations should be used as complementary tools in aerodynamic design and analysis. Figure 3.8 shows some of the most important properties of numerical and experimental flow simulations. [78]

The major advantages and disadvantages of wind-tunnel testing are listed below:

- Advantages:
  - Direct simulation of complex flow phenomena
  - Rapid variation of operating conditions
  - Validation data for numerical simulations

- Disadvantages:
  - Cost (facility + models)
  - Limited data resolution
  - Limited flexibility of model configurations
  - Challenging, or even impossible to achieve, similarity to full scale flows

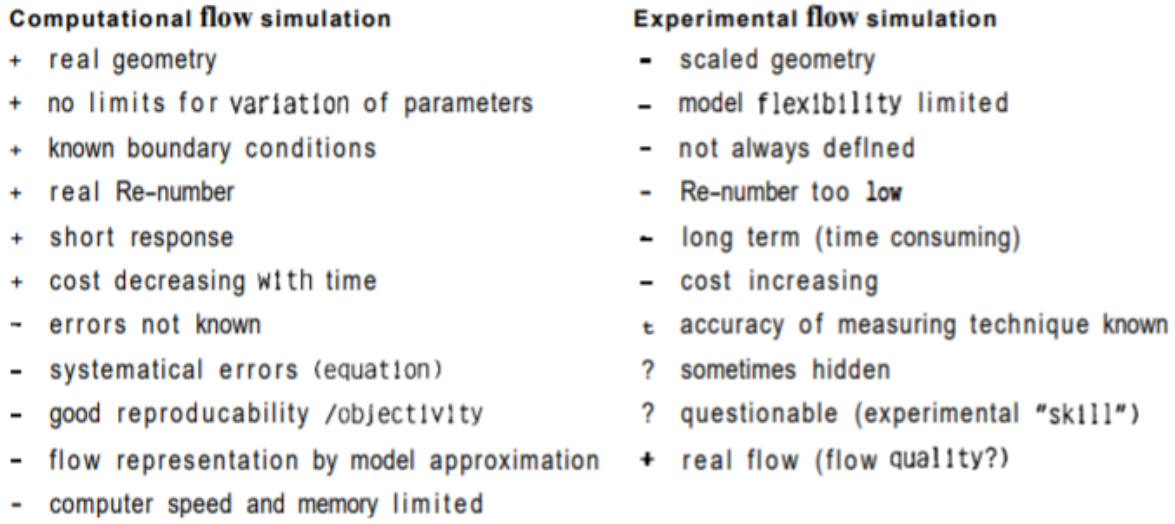


Figure 3.8: Compilation of major characteristic features in computational (CFD) and experimental flow simulations. [78]

4

Aerodynamic database and data fusion

Generation of an aerodynamic database is always needed in the study of performance, stability and control of aircraft. In its broadest sense, it consists of static coefficients, and their static and dynamic derivatives with respect to many different factors the vehicle could experience along its trajectory, e.g., change of flight conditions (speed, angles of attack or sideslip), deflection of control surfaces, change in landing gear position, etc. A full database is produced by examination of all of these factors and their different combinations by means of flight tests, wind-tunnel testing and computational modeling. The creation of a complete database requires an excessive amount of time, thus it is important to limit the number of test points as much as possible. Hence, surrogate models should be implemented to reproduce a continuous database in lieu of simulating and testing *all the points*.

Various forces acting on flying vehicles is what determines their motion. Traditionally, for the linearized stability analysis, it can be separated into two modes of motion: longitudinal motion and lateral-directional motion. The former consists of the forces acting on the aircraft in its longitudinal plane, i.e., the forces of drag and lift, as well as the pitching moment. The latter is concerned with the motion in the lateral and directional planes, that is, the side-force, and rolling and yawing moments. To assess the effect of these forces and moments, engineers usually operate with coefficients such as:

- Drag force coefficient:

$$C_D = \frac{D}{\frac{1}{2}\rho_\infty V_\infty^2 S_{ref}} \tag{4.1}$$

- Lift force coefficient:

$$C_L = \frac{L}{\frac{1}{2}\rho_\infty V_\infty^2 S_{ref}} \tag{4.2}$$

- Pitching moment coefficient:

$$C_M = \frac{M}{\frac{1}{2}\rho_\infty V_\infty^2 S_{ref} c_{ref}} \tag{4.3}$$

- Side force coefficient:

$$C_Y = \frac{Y}{\frac{1}{2}\rho_\infty V_\infty^2 S_{ref}} \tag{4.4}$$

- Rolling moment coefficient:

$$C_R = \frac{R}{\frac{1}{2}\rho_\infty V_\infty^2 S_{ref} b_{ref}} \tag{4.5}$$

- Yawing moment coefficient:

$$C_N = \frac{N}{\frac{1}{2}\rho_\infty V_\infty^2 S_{ref} b_{ref}} \tag{4.6}$$

These coefficients are the non-dimensionalized forces and moments acting on the airframe. In the equations above D, L and Y are the drag, lift and side force respectively, and M, R and N are the pitching, rolling and yawing moments. The values used for the non-dimensionalization are the freestream air density (ρ_∞), the freestream velocity (V_∞), the reference area (S_{ref}), reference chord (c_{ref}) and reference span (b_{ref}).

The combination of the first three coefficients as a function of, e.g., the angle of attack, Reynolds number and Mach number would constitute the longitudinal motion aerodynamic database. Similarly, the latter three coefficients form the lateral-directional motion aerodynamic database. For a complete evaluation of a given design it is paramount to consider both databases and how they interrelate. However, given the time-constraints of this project, only the longitudinal motion will be considered.

Section 3.2 discussed various methods of aerodynamic analysis of aircraft. This section builds upon what was introduced in that section and extends the discussion to creation of aerodynamic databases. There are many methods the surrogate models can be created. There are the response surface models, artificial neural networks, or various interpolation methods. **Section 4.1** discusses a single-fidelity interpolation method called kriging, frequently used for building aerodynamic databases. Then **Section 4.2** introduces co-kriging - a multi-fidelity data fusion method that is an extension to the kriging method.

4.1. Kriging

Kriging is a statistical interpolation technique first used in geostatistics. Its theory was mathematically developed by Georges Matheron in 1960, based on Danie Krige's master's thesis [95]. The method was first introduced into engineering by Sacks et al. [96]. The derivation of the method presented in this report is based on the article of Jones, 2001 [97]. For a standard derivation, one can refer to the technique's original article [96].

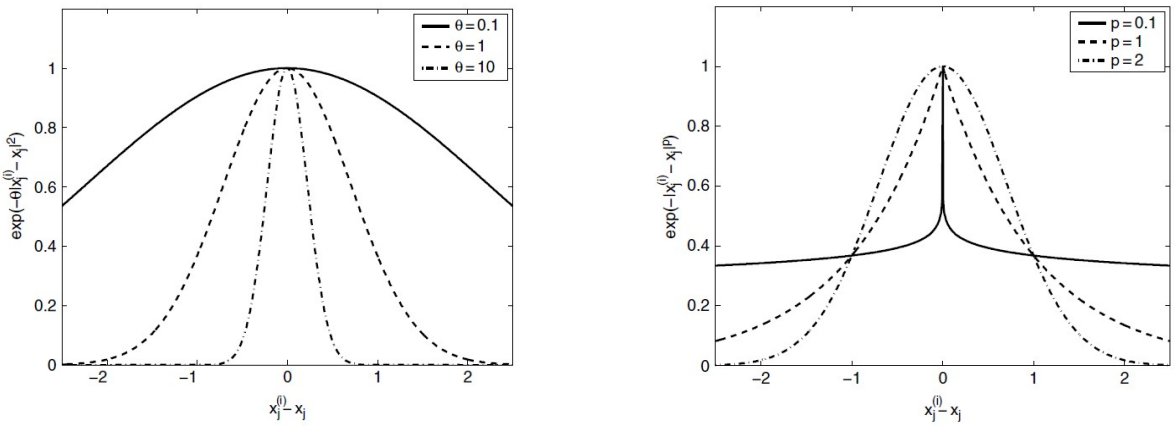
Assume a point $\boldsymbol{x}$ in the domain for which a prediction is to be made. The uncertainty of the value of the function at this point $\boldsymbol{x}$ can be modeled by the realization of a normally distributed random variable $Y(\boldsymbol{x})$ with mean μ and variance σ^2.

For two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and assuming the modeled function is continuous, the values $y(\boldsymbol{x}_i)$ and $y(\boldsymbol{x}_j)$ tend to be close if the distance between the points $||\boldsymbol{x}_i - \boldsymbol{x}_j||$ is small. It can be statistically modeled by assuming the random variables $Y(\boldsymbol{x}_i)$ and $Y(\boldsymbol{x}_j)$ to be highly correlated, given that $||\boldsymbol{x}_i - \boldsymbol{x}_j||$ is small. This correlation is given by Equation 4.7.

$$Corr\left[Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j)\right] = exp\left(-\sum_{l=1}^{k} \theta_l |\boldsymbol{x}_{il} - \boldsymbol{x}_{jl}|^{p_l}\right) \tag{4.7}$$

Intuitively, for $\boldsymbol{x}_i = \boldsymbol{x}_j$, the correlation $Corr\left[Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j)\right] = 1$. Similarly, for $||\boldsymbol{x}_i - \boldsymbol{x}_j|| \to \infty$, the correlation $Corr\left[Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j)\right] \to 0$. The parameter θ_l determines the rate with which the correlation decreases when the distance between the two points in the l^{th} direction increases. Then, for functions that change rapidly with the l^{th} variable, the values of θ_l will be large. The other parameter, p_l, is used to model the smoothness of the described functions along the l^{th}

coordinate. If the the value of p_l is equal or close to 2, the model will be smooth, while the values of p_l near 0 help model rough, non-differentiable functions. The above description of both parameters is illustrated in Figure 4.1.



(a) (b)

Figure 4.1: Correlations with varying θ (fig. a) and p (fig. b). [95]

The uncertainty of the function's values at arbitrary n points can be represented as a vector of random variables:

$$\boldsymbol{Y} = \begin{pmatrix} Y(\boldsymbol{x}_1) \\ \vdots \\ Y(\boldsymbol{x}_n) \end{pmatrix} \tag{4.8}$$

Then, the mean value of the above vector is $\mathbf{1}\mu$ (with $\mathbf{1}$ being an $n \times 1$ vector of ones) and the covariance matrix of said vector is as defined by Equation 4.9.

$$Cov(\boldsymbol{Y}) = \sigma^2 \boldsymbol{R} \tag{4.9}$$

$\boldsymbol{R}$ in Equation 4.9 is an $n \times n$ matrix, where each element (i, j) is the correlation defined in Equation 4.7. Thus, the distribution of the vector $\boldsymbol{Y}$ depends on the parameters μ, σ^2, θ_l and p_l, and characterizes the behaviour of the modeled function.

In order to estimate the values of the four parameters, one needs to find their values that maximize the likelihood of the observed data. The vector of the observed data is defined by Equation 4.10.

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{4.10}$$

Following this notation, the likelihood of $\boldsymbol{y}$ can be written as Equation 4.11.

$$L = \frac{1}{(2\pi)^{\frac{n}{2}} (\sigma^2)^{\frac{n}{2}} |\boldsymbol{R}|^{\frac{1}{2}}} exp\left[\frac{-(\boldsymbol{y} - \mathbf{1}\mu)^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \mathbf{1}\mu)}{2\sigma^2}\right] \tag{4.11}$$

Maximizing the likelihood function means that the created model of the function's behaviour needs to be the most consistent with the observed data. In order to achieve this, it is more convenient to maximize the natural logarithm of the likelihood function, thus:

$$ln(L) = -\frac{n}{2}ln(2\pi) - \frac{n}{2}ln(\sigma^2) - \frac{1}{2}ln|\boldsymbol{R}| - \frac{(\boldsymbol{y} - \mathbf{1}\mu)^T \boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\mu)}{2\sigma^2} \tag{4.12}$$

The maximum likelihood estimates (MLEs) for μ (Equation 4.13) and σ^2 (Equation 4.14) are obtained by taking the derivatives of Equation 4.12 with respect to each parameter and equaling to zero.

$$\widehat{\mu} = \frac{\mathbf{1}^T \boldsymbol{R}^{-1} \boldsymbol{y}}{\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}} \tag{4.13}$$

$$\widehat{\sigma}^2 = \frac{(\boldsymbol{y} - \mathbf{1}\widehat{\mu})^T \boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\widehat{\mu})}{n} \tag{4.14}$$

Substituting Equations 4.13 and 4.14 into Equation 4.12 yields the *concentrated ln-likelihood function*:

$$ln(L) \approx -\frac{n}{2}ln\left(\widehat{\sigma}^2\right) - \frac{1}{2}ln|\boldsymbol{R}| \tag{4.15}$$

Since Equation 4.15 depends only on $\boldsymbol{R}$, it depends solely on the previously introduced correlation parameters θ_l and p_l. Thus, in order to find the estimates $\widehat{\theta}_l$ and $\widehat{p}_l$, Equation 4.15 needs to be maximized. Then, Equations 4.13 and 4.14 are used to compute $\widehat{\mu}$ and $\widehat{\sigma}^2$. To achieve this, usually a numerical optimization technique is used. For *not-too-large* n and k Equation 4.15 should be quick to compute. Usually used bounds for $\widehat{\boldsymbol{\theta}}$ are from 10^{-3} to 10^2. And as for $\widehat{\boldsymbol{p}}$, while tuning it is recommended, to reduce the complexity of the problem it can be set to $\widehat{\boldsymbol{p}} = \mathbf{2}$, assuming a smooth, continuous function. [95]

The above approach presents how to find the kriging correlation. Now follows the derivation of the kriging predictor. With the correlation parameters $\widehat{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{p}}$ already found, a new prediction $\widehat{y}$ at $\widehat{\boldsymbol{x}}$, in order to be consistent with the observed data, needs to be consistent with both these parameters. Therefore, a prediction that maximizes the likelihood of the sample data (the data used to create the model) and the prediction needs to be found. As the first step, the vector of the observed data $\boldsymbol{y}$ is augmented (Equation 4.16) by the new prediction $\widehat{y}$, the value of which has not yet been determined. The vector of correlations between $\boldsymbol{y}$ and $\widehat{y}$ is shown in Equation 4.17, and the correlation matrix for this augmented data set in Equation 4.18. The last element in Equation 4.18 is 1. This is the continuation of the main diagonal of $\boldsymbol{R}$, which is the correlation of a point with itself, i.e., $|\boldsymbol{x}_i - \boldsymbol{x}_j| = 0$ and $Corr\left[Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j)\right] = 1$.

$$\widetilde{\boldsymbol{y}} = \begin{Bmatrix} \boldsymbol{y} \\ \widehat{y} \end{Bmatrix} \tag{4.16}$$

$$\boldsymbol{r} = \begin{pmatrix} Corr\left[Y(\widehat{\boldsymbol{x}}), Y(\boldsymbol{x}_1)\right] \\ \vdots \\ Corr\left[Y(\widehat{\boldsymbol{x}}), Y(\boldsymbol{x}_n)\right] \end{pmatrix} \tag{4.17}$$

$$\widetilde{\boldsymbol{R}} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{r} \\ \boldsymbol{r}^T & 1 \end{pmatrix} \tag{4.18}$$

Hence, the ln-likelihood of the augmented data set is:

$$ln(L) = -\frac{n}{2}ln(2\pi) - \frac{n}{2}ln(\sigma^2) - \frac{1}{2}ln|\widetilde{\boldsymbol{R}}| - \frac{(\widetilde{\boldsymbol{y}} - \mathbf{1}\mu)^T \widetilde{\boldsymbol{R}}^{-1}(\widetilde{\boldsymbol{y}} - \mathbf{1}\mu)}{2\sigma^2} \tag{4.19}$$

It should be noted that in Equation 4.19 only the last term is dependent on $\widehat{y}$. Therefore, this is the only term that is considered in the maximization process. Substituting into it Equations 4.16 and 4.18 yields:

$$ln(L) \approx \frac{-\begin{pmatrix} \boldsymbol{y} - \mathbf{1}\widehat{\mu} \\ \widehat{y} - \widehat{\mu} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{R} & \boldsymbol{r} \\ \boldsymbol{r}^T & 1 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{y} - \mathbf{1}\widehat{\mu} \\ \widehat{y} - \widehat{\mu} \end{pmatrix}}{2\widehat{\sigma}^2} \tag{4.20}$$

The maximization of the above equation follows from the inverse of $\widetilde{\boldsymbol{R}}$ using the partitioned inverse formula [98] giving Equation 4.21, which substituted into Equation 4.20 yields Equation 4.22 (removing the terms not dependent on $\widehat{y}$).

$$\widetilde{\boldsymbol{R}}^{-1} = \begin{pmatrix} \boldsymbol{R}^{-1} + \boldsymbol{R}^{-1}\boldsymbol{r}\left(1 - \boldsymbol{r}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right)^{-1}\boldsymbol{r}^T\boldsymbol{R}^{-1} & -\boldsymbol{R}^{-1}\boldsymbol{r}\left(1 - \boldsymbol{r}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right)^{-1} \\ -\left(1 - \boldsymbol{r}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right)^{-1}\boldsymbol{r}^T\boldsymbol{R}^{-1} & \left(1 - \boldsymbol{r}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right)^{-1} \end{pmatrix} \tag{4.21}$$

$$ln(L) \approx \left(\frac{-1}{2\widehat{\sigma}^2\left(1 - \boldsymbol{r}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right)}\right)(\widehat{y} - \widehat{\mu})^2 + \left(\frac{\boldsymbol{r}^T\boldsymbol{R}^{-1}\left(\boldsymbol{y} - \mathbf{1}\widehat{\mu}\right)}{\widehat{\sigma}^2\left(1 - \boldsymbol{r}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right)}\right)(\widehat{y} - \widehat{\mu}) \tag{4.22}$$

Then, the maximum of the quadratic function from Equation 4.22 can be found by differentiating with respect to $\widehat{y}$ and equalling to zero. After these operations, the MLE for $\widehat{y}$ is the kriging predictor:

$$\widehat{y}(\boldsymbol{x}) = \widehat{\mu} + \boldsymbol{r}^T\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\widehat{\mu}) \tag{4.23}$$

If further $\widehat{\mu} = a$, b_i is the i^{th} element of $\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\widehat{\mu})$, and $\phi(\widehat{\boldsymbol{x}} - \boldsymbol{x}_i)$ is the i^{th} element of $\boldsymbol{r}^T\left(Corr\left[Y(\widehat{\boldsymbol{x}}), Y(\boldsymbol{x}_i)\right]\right)$ (similar to Equation 4.7), the kriging predictor can be rewritten as:

$$\widehat{y}(\boldsymbol{x}) = a + \sum_{i=1}^{n} b_i\phi\left(\widehat{\boldsymbol{x}} - \boldsymbol{x}_i\right) \tag{4.24}$$

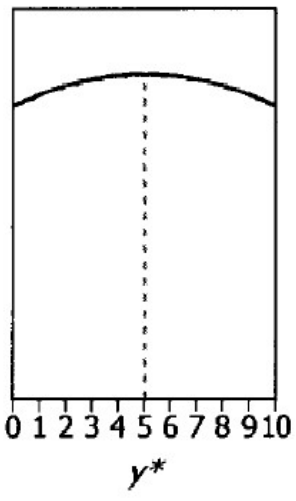
With the kriging predictor as shown in Equation 4.24, one can see that it is basically a linear combination of basis functions and polynomial terms.

Following this derivation, it can be seen that the confidence in the predictor of the ln-likelihood of the augmented data set decreases rapidly as $\widehat{y}$ moves away from its optimal value. Figure 4.2 illustrates this observation. If the curvature of the *augmented ln-likelihood* is small, the confidence in the predictor cannot be high since values of $\widehat{y}$ quite different from its optimal value could perform comparably well. Whereas, if the curvature of this curve is high, subtle differences in the values of $\widehat{y}$ cause substantial inconsistencies with the observed data, thus the confidence in the kriging predictor is high. Following this argument, one can notice that there should be an inverse relation between the potential error of the predictor and the curvature of the augmented ln-likelihood. This curvature can be computed by taking the second derivative with respect to $\widehat{y}$ of Equation 4.22. Remembering that the potential error is inversely related to the curvature, the estimate of the error is shown in Equation 4.25. Then, deriving the mean-squared error of the predictor with the standard stochastic-process procedure gives

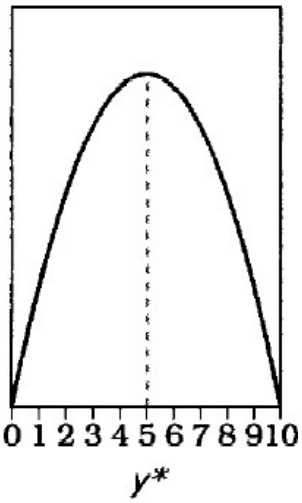
Equation 4.26. Comparing the two equations highlights an *extra* term on the *RHS* of Equation 4.26, which can be understood as the representation of the uncertainty coming from the fact that the mean value is not known exactly, but only estimated from the observed data.

$$\epsilon = \widehat{\sigma}^2 \left(1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r}\right) \tag{4.25}$$

$$s^2(\widehat{\boldsymbol{x}}) = \widehat{\sigma}^2 \left[1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r} + \frac{\left(1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r}\right)^2}{\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}}\right] \tag{4.26}$$



(a) Small curvature of ln-likelihood function - low confidence.



(b) Big curvature of ln-likelihood function - high confidence.

Figure 4.2: Examples of how augmented ln-likelihood might depend upon $\widehat{y}$ (y^* in the figure). [97]

4.2. Co-kriging

Section 3.2 discussed various methods of calculating the flow around a flying vehicle. It is often the case that not one, but many tools of varying levels of fidelity are used to analyse a problem. The data gathered using low-fidelity methods is frequently referred to as *cheap data*, given that the computational time required to obtain it is drastically lower compared to higher-fidelity data. Similarly, high-fidelity data is called *expensive data*. Co-kriging, the topic of this section, is a multi-fidelity data fusion method that can correlate a big quantity of cheap data with a smaller amount of expensive data to create an accurate model of the expensive function. Co-kriging is basically a form of kriging that is able to correlate several data sets. The following derivation shows how to conduct the correction process to correlate the cheap and expensive data. This derivation, for simplicity limited to just two data sets, is entirely based on the work of Forrester, Sóbester, and Keane, 2008 [95].

First, assume there are two data sets - the expensive data with values $\boldsymbol{y}_e$ at points $\boldsymbol{X}_e$, and the cheap data with values $\boldsymbol{y}_c$ at points $\boldsymbol{X}_c$. Furthermore, the set of expensive data points is a subset of the set of cheap data points ($\boldsymbol{X}_e \subset \boldsymbol{X}_c$). To begin with co-kriging, the two sets of points are concatenated:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_c \\ \boldsymbol{X}_e \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{c_1} \\ \vdots \\ \boldsymbol{x}_{c_{n_c}} \\ \boldsymbol{x}_{e_1} \\ \vdots \\ \boldsymbol{x}_{e_{n_e}} \end{pmatrix} \tag{4.27}$$

Similarly to kriging, the values at points in the combined data set are treated as if they were coming from a stochastic process, which gives the random field:

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Y}_c(\boldsymbol{X}_c) \\ \boldsymbol{Y}_e(\boldsymbol{X}_e) \end{pmatrix} = \begin{pmatrix} Y_c(\boldsymbol{x}_{c_1}) \\ \vdots \\ Y_c(\boldsymbol{x}_{c_{n_c}}) \\ Y_e(\boldsymbol{x}_{e_1}) \\ \vdots \\ Y_e(\boldsymbol{x}_{e_{n_e}}) \end{pmatrix} \tag{4.28}$$

Then, the *auto-regressive* model of Kennedy and O'Hagan, 2000 [99] is used, which assumes that if the value of the expensive function is known at $\boldsymbol{x}_i$, nothing new can be learned about $Y_e(\boldsymbol{x}_i)$ from the cheaper data. To put it simply, the assumption is made that the expensive data is correct and the inaccuracies come solely from the cheaper data.

The cheap and expensive models are represented with Gaussian processes Z_c and Z_e respectively. When the auto-regressive model is used, the expensive simulation is approximated as the cheap simulation multiplied by a constant scaling factor ρ and another Gaussian process Z_d (the difference between Z_e and ρZ_c).

$$Z_e(\boldsymbol{x}) = \rho Z_c(\boldsymbol{x}) + Z_d(\boldsymbol{x}) \tag{4.29}$$

Analogously to the covariance matrix constructed in kriging (Equation 4.9), there now is a covariance matrix (Equation 4.33) constructed through Equations 4.30 - 4.33, where $\boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_e)$ is a matrix of correlations between the data $\boldsymbol{X}_e$ and $\boldsymbol{X}_c$ of the form r_c (Equation 4.17), etc.

$$Cov\{\boldsymbol{Y}_c(\boldsymbol{X}_c), \boldsymbol{Y}_c(\boldsymbol{X}_c)\} = Cov\{Z_c(\boldsymbol{X}_c), Z_c(\boldsymbol{X}_c)\} = \sigma_c^2 \boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c) \tag{4.30}$$

$$Cov\{\boldsymbol{Y}_e(\boldsymbol{X}_e), \boldsymbol{Y}_c(\boldsymbol{X}_c)\} = Cov\{\rho Z_c(\boldsymbol{X}_c) + Z_d(\boldsymbol{X}_c), Z_c(\boldsymbol{X}_e)\} = \rho\sigma_c^2 \boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_e) \tag{4.31}$$

$$\begin{aligned} Cov\{\boldsymbol{Y}_e(\boldsymbol{X}_e), \boldsymbol{Y}_e(\boldsymbol{X}_e)\} &= Cov\{\rho Z_c(\boldsymbol{X}_e) + Z_d(\boldsymbol{X}_e), \rho Z_c(\boldsymbol{X}_e) + Z_d(\boldsymbol{X}_e)\} \\ &= \rho^2 Cov\{Z_c(\boldsymbol{X}_e), Z_c(\boldsymbol{X}_e)\} + Cov\{Z_d(\boldsymbol{X}_e), Z_d(\boldsymbol{X}_e)\} \\ &= \rho^2\sigma_c^2 \boldsymbol{R}_c(\boldsymbol{X}_e, \boldsymbol{X}_e) + \sigma_d^2 \boldsymbol{R}_d(\boldsymbol{X}_e, \boldsymbol{X}_e) \end{aligned} \tag{4.32}$$

$$\boldsymbol{C} = \begin{pmatrix} \sigma_c^2 \boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c) & \rho\sigma_c^2 \boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_e) \\ \rho\sigma_c^2 \boldsymbol{R}_c(\boldsymbol{X}_e, \boldsymbol{X}_c) & \rho^2\sigma_c^2 \boldsymbol{R}_c(\boldsymbol{X}_e, \boldsymbol{X}_e) + \sigma_d^2 \boldsymbol{R}_d(\boldsymbol{X}_e, \boldsymbol{X}_e) \end{pmatrix} \tag{4.33}$$

These correlations take the same form as Equation 4.7. However, in co-kriging there are now two correlations, r_c and r_d, which increases the amount of parameters to be estimated to $\boldsymbol{\theta}_c$, $\boldsymbol{\theta}_d$, $\boldsymbol{p}_c$, $\boldsymbol{p}_d$, and the scaling parameter ρ. The cheap data is independent from the expensive data, thus the MLEs for μ_c, σ_c^2, $\boldsymbol{\theta}_c$ and $\boldsymbol{p}_c$ can be found by maximization of the ln-likelihood (Equation 4.34, with constant terms ignored):

$$ln(L_c) = -\frac{n_c}{2} ln(\sigma_c^2) - \frac{1}{2} ln|\boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c)| - \frac{(\boldsymbol{y}_c - \mathbf{1}\mu_c)^T \boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c)^{-1} (\boldsymbol{y}_c - \mathbf{1}\mu_c)}{2\sigma_c^2} \tag{4.34}$$

Finally, the MLEs are found by solving Equations 4.35 and 4.36, which come from equaling to zero the derivatives of Equation 4.34 with respect to μ_c and σ_c^2.

$$\widehat{\mu}_c = \frac{\mathbf{1}^T \boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c)^{-1} \boldsymbol{y}_c}{\mathbf{1}^T \boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c)^{-1} \mathbf{1}} \tag{4.35}$$

$$\widehat{\sigma}_c^2 = \frac{(\boldsymbol{y}_c - \mathbf{1}\widehat{\mu}_c)^T \boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c)^{-1} (\boldsymbol{y}_c - \mathbf{1}\widehat{\mu}_c)}{n_c} \tag{4.36}$$

Then, Equations 4.35 and 4.36 are substituted into Equation 4.34 to give Equation 4.37 - the concentrated ln-likelihood. Then, in the same way as in kriging, $\boldsymbol{\theta}_c$ and $\boldsymbol{p}_c$ are estimated by maximizing this equation.

$$ln(L_c) \approx -\frac{n_c}{2} ln\left(\widehat{\sigma}_c^2\right) - \frac{1}{2} ln|\boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c)| \tag{4.37}$$

To find μ_d, σ_d^2, $\boldsymbol{\theta}_d$, $\boldsymbol{p}_d$ and ρ, Equation 4.38 is used to define the difference between the expensive and cheap data at points $\boldsymbol{X}_e$. If $\boldsymbol{y}_c$ is unavailable at some points in $\boldsymbol{X}_e$, ρ can be estimated using the kriging predictor $\widehat{\boldsymbol{y}}_c(\boldsymbol{X}_e)$ (Equation 4.23), where the already determined parameters $\widehat{\boldsymbol{\theta}}_c$ and $\widehat{\boldsymbol{p}}_c$ are used. This operation gives Equation 4.39.

$$\boldsymbol{d} = \boldsymbol{y}_e - \rho \boldsymbol{y}_c(\boldsymbol{X}_e) \tag{4.38}$$

$$ln(L_d) = -\frac{n_e}{2} ln(\sigma_d^2) - \frac{1}{2} ln|\boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c)| - \frac{(\boldsymbol{d} - \mathbf{1}\mu_d)^T \boldsymbol{R}_d(\boldsymbol{X}_e, \boldsymbol{X}_e)^{-1} (\boldsymbol{d} - \mathbf{1}\mu_d)}{2\sigma_d^2} \tag{4.39}$$

Again, MLEs $\widehat{\mu}_d$ and $\widehat{\sigma}_d^2$ are found the same way as before (Equations 4.40 and 4.41), and $\widehat{\boldsymbol{\theta}}_d$, $\widehat{\boldsymbol{p}}_d$ and $\widehat{\rho}$ by maximizing the concentrated ln-likelihood in Equation 4.42.

$$\widehat{\mu}_d = \frac{\mathbf{1}^T \boldsymbol{R}_d(\boldsymbol{X}_e, \boldsymbol{X}_e)^{-1} \boldsymbol{d}}{\mathbf{1}^T \boldsymbol{R}_d(\boldsymbol{X}_e, \boldsymbol{X}_e)^{-1} \mathbf{1}} \tag{4.40}$$

$$\widehat{\sigma}_d^2 = \frac{(\boldsymbol{d} - \mathbf{1}\widehat{\mu}_d)^T \boldsymbol{R}_d(\boldsymbol{X}_e, \boldsymbol{X}_e)^{-1} (\boldsymbol{d} - \mathbf{1}\widehat{\mu}_d)}{n_e} \tag{4.41}$$

$$ln(L_d) \approx -\frac{n_e}{2} ln\left(\widehat{\sigma}_d^2\right) - \frac{1}{2} ln|\boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c)| \tag{4.42}$$

Similarly to what was commented in **Section 4.1**, to maximize Equations 4.37 and 4.42 numerical algorithms may be used. However, one must be aware of the cost of these evaluations since for high-dimensional problems the many matrix inversions required for the maximization can make co-kriging impractical. The size of the matrices is determined by the amount of data points, and the number of search steps involved in the MLE operation depends on the number of parameters being tuned. In some cases this number can be reduced by assuming some parameters constant, but then the precision of the surrogate model would be affected, and its accuracy potentially reduced. Therefore, such assumptions must be made with caution.

Noticeably, the derivation of the co-kriging model followed the same line of thought as was used for kriging in **Section 4.1**. It is no different in the case of the co-kriging predictor. To create the co-kriging predictor, the observed data is augmented with a predicted value, and then the likelihood of this new, augmented data set needs to be maximized by varying the prediction while the model parameters are kept fixed, which will give an MLE $\widehat{y}_e(\boldsymbol{x})$. Equation 4.43 shows the matrix of concatenated cheap and expensive data points augmented with the prediction point, Equation 4.44 shows the vector of observed data augmented with the prediction, and Equation 4.45 is the augmented covariance matrix.

$$\widetilde{\boldsymbol{X}} = \begin{Bmatrix} \boldsymbol{X}_c \\ \boldsymbol{X}_e \\ \boldsymbol{x} \end{Bmatrix} \tag{4.43}$$

$$\widetilde{\boldsymbol{y}} = \begin{Bmatrix} \boldsymbol{y}_c \\ \boldsymbol{y}_e \\ \widehat{y}_e(\boldsymbol{x}) \end{Bmatrix} \tag{4.44}$$

$$\widetilde{\boldsymbol{C}} = \begin{pmatrix} \widehat{\sigma}_c^2 \boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_c) & \rho\widehat{\sigma}_c^2 \boldsymbol{R}_c(\boldsymbol{X}_c, \boldsymbol{X}_e) & \rho\widehat{\sigma}_c^2 \boldsymbol{r}_c(\boldsymbol{X}_c, \boldsymbol{x}) \\ \rho\widehat{\sigma}_c^2 \boldsymbol{R}_c(\boldsymbol{X}_e, \boldsymbol{X}_c) & \rho^2\widehat{\sigma}_c^2 \boldsymbol{R}_c(\boldsymbol{X}_e, \boldsymbol{X}_e) + \widehat{\sigma}_d^2 \boldsymbol{R}_d(\boldsymbol{X}_e, \boldsymbol{X}_e) & \left(\rho^2\widehat{\sigma}_c^2 + \widehat{\sigma}_d^2\right) \boldsymbol{r}_d(\boldsymbol{X}_e, \boldsymbol{x}) \\ \rho\widehat{\sigma}_c^2 \boldsymbol{r}_c(\boldsymbol{X}_c, \boldsymbol{x})^T & \left(\rho^2\widehat{\sigma}_c^2 + \widehat{\sigma}_d^2\right) \boldsymbol{r}_d(\boldsymbol{X}_e, \boldsymbol{x})^T & \rho^2\widehat{\sigma}_c^2 + \widehat{\sigma}_d^2 \end{pmatrix} \tag{4.45}$$

If $\boldsymbol{c}$ is a column vector of the covariance of $\boldsymbol{X}$ and $\boldsymbol{x}$, and remembering Equation 4.33, Equation 4.45 can be rewritten in a much simpler form:

$$\widetilde{\boldsymbol{C}} = \begin{pmatrix} \boldsymbol{C} & \boldsymbol{c} \\ \boldsymbol{c}^T & \rho^2\widehat{\sigma}_c^2 + \widehat{\sigma}_d^2 \end{pmatrix} \tag{4.46}$$

Still by analogy to kriging, to find an MLE $\widehat{y}_e(\boldsymbol{x})$, the term in Equation 4.47 needs to be maximized, which can be expressed as Equation 4.48. The inverse of the augmented covariance matrix from Equation 4.46 is presented in Equation 4.49.

$$-\frac{1}{2}(\widetilde{\boldsymbol{y}} - \boldsymbol{1}\mu)^T \widetilde{\boldsymbol{C}}^{-1} (\widetilde{\boldsymbol{y}} - \boldsymbol{1}\mu) \tag{4.47}$$

$$-\frac{1}{2} \begin{pmatrix} \boldsymbol{y} - \boldsymbol{1}\widehat{\mu} \\ \widehat{y}_e(\boldsymbol{x}) - \widehat{\mu} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{C} & \boldsymbol{c} \\ \boldsymbol{c}^T & \rho^2\widehat{\sigma}_c^2 + \widehat{\sigma}_d^2 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{y} - \boldsymbol{1}\widehat{\mu} \\ \widehat{y}_e(\boldsymbol{x}) - \widehat{\mu} \end{pmatrix} \tag{4.48}$$

$$\widetilde{\boldsymbol{C}}^{-1} = \begin{pmatrix} \boldsymbol{C}^{-1} + \boldsymbol{C}^{-1}\boldsymbol{c}\left(\rho^2\widehat{\sigma}_c^2 + \widehat{\sigma}_d^2 - \boldsymbol{c}^T\boldsymbol{C}^{-1}\boldsymbol{c}\right)^{-1} \boldsymbol{c}^T\boldsymbol{C}^{-1} & -\boldsymbol{C}^{-1}\boldsymbol{c}\left(\rho^2\widehat{\sigma}_c^2 + \widehat{\sigma}_d^2 - \boldsymbol{c}^T\boldsymbol{C}^{-1}\boldsymbol{c}\right)^{-1} \\ -\left(\rho^2\widehat{\sigma}_c^2 + \widehat{\sigma}_d^2 - \boldsymbol{c}^T\boldsymbol{C}^{-1}\boldsymbol{c}\right)^{-1} \boldsymbol{c}^T\boldsymbol{C}^{-1} & \left(\rho^2\widehat{\sigma}_c^2 + \widehat{\sigma}_d^2 - \boldsymbol{c}^T\boldsymbol{C}^{-1}\boldsymbol{c}\right)^{-1} \end{pmatrix} \tag{4.49}$$

Now Equation 4.49 is substituted into Equation 4.48, which with operations analogous to these commented at the end of **Section 4.1**, ultimately gives Equation 4.50, which is the co-kriging predictor.

$$\widehat{y}_e(\boldsymbol{x}) = \widehat{\mu} + \boldsymbol{c}^T \boldsymbol{C}^{-1}(\boldsymbol{y} - \mathbf{1}\widehat{\mu}) \tag{4.50}$$

If a prediction is made at an expensive point, e.g., $\boldsymbol{x}_{n+1} = \boldsymbol{x}_{e_i}$ and $\boldsymbol{c}$ is the $(n_c + i)^{th}$ column of $\boldsymbol{C}$, then $\boldsymbol{c}^T \boldsymbol{C}^{-1}$ is the $(n_c + i)^{th}$ unit vector and $\widehat{y}_e(\boldsymbol{x}_{e_i}) = \widehat{\mu} + \boldsymbol{y}_{n_c+i} - \widehat{\mu} = y_{e_i}$. Therefore, Equation 4.50 interpolates the expensive data, but in a sense will regress the cheap data where it does not coincide with $\boldsymbol{y}_e$. Lastly, the mean-squared error by analogy to kriging is measured with Equation 4.51.

$$s^2(\boldsymbol{x}) = \rho^2 \widehat{\sigma}_c^2 + \widehat{\sigma}_d^2 - \boldsymbol{c}^T \boldsymbol{C}^{-1} \boldsymbol{c} + \frac{1 - \boldsymbol{c}^T \boldsymbol{C}^{-1} \boldsymbol{c}}{\mathbf{1}^T \boldsymbol{C}^{-1} \mathbf{1}} \tag{4.51}$$

5

Knowledge Based Engineering framework

The main goal of this thesis work was the creation of an integrated framework to assist in preliminary aerodynamic design of spaceplanes. This chapter presents in detail the KBE framework designed and implemented for this task. The first section, **Section 5.1**, shows the architecture of the framework - explains the workflow and gives an overview of how different modules work with each other. **Section 5.2** presents in more detail the geometry module, where the implemented parametrization is described. A few examples are used to demonstrate its flexibility. Then, **Section 5.3** focuses on the low-fidelity aerodynamic analysis module. This section gives the reasoning behind the choice of this particular software, and briefly explains its calculation methods. **Section 5.4** discusses how the high-fidelity aerodynamic analysis is performed in this project. Lastly, **Section 5.5** discusses the final step of the workflow - data fusion of the results coming from the low- and high-fidelity analyses.

5.1. Framework architecture

On the highest level, the framework can be divided into two main branches - the *Spaceplane Branch* and the *Fluid Domain Branch*. Figure 5.1 shows the workflow in both branches.

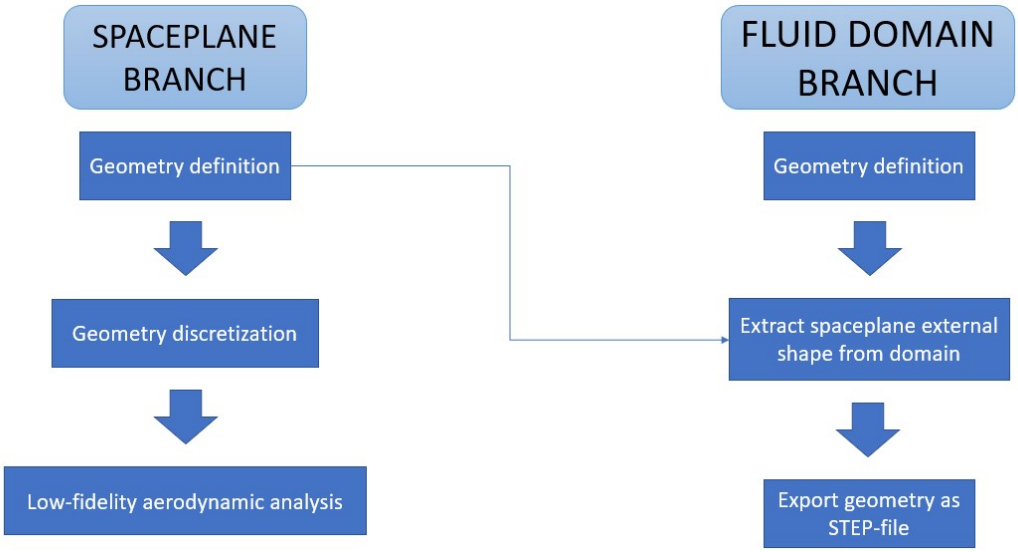


Figure 5.1: Basic high-level overview of the framework's structure.

The first step in the Spaceplane Branch is the definition of the spaceplane external shape (an in-depth description is offered in **Section 5.2**). This is where a potential designer can create a model, play with its parameters, and explore the design-space. To alleviate the workload traditionally put on the engineer, the following steps in this branch are done in large part automatically, with only minimum user's input.

Once the geometry is defined, the next step is to prepare it for the analysis. The low-fidelity aerodynamic analysis module employs ZONAIR, which is a software that utilizes a panel code for its calculations. This is the topic of **Section 5.3**, which opens with a detailed reasoning behind the selection of this particular program in **Section 5.3.1**. The geometry discretization for this panel code is limited to a surface mesh of the external geometry, and a few ZONAIR-specific models. A structured surface mesh is created automatically in the following steps:

1. Geometry is divided into blocks.
2. The number of divisions along the blocks' characteristic edges, as well as a *bias factor* can be easily changed according to the user's needs. This way a high level of mesh control is given to the user.
3. Each block is divided into panels using a *quad-dominant* mesh.

Once the model is discretized with a surface mesh, the ZONAIR-specific models are created. For proper functioning ZONAIR requires modeling of trailing edges of bodies and lifting surfaces, tips of lifting surfaces and wing-body junctions. The nodes positioned at the respective locations are automatically found, and the information is passed to the adequate functions within ZONAIR environment.

Then, the last step to be realized is the aerodynamic analysis. Here, the user needs to provide the flight conditions at which the analysis is to be conducted. The information required consists of a range of Mach numbers and altitudes, and a range of angles of attack. Additionally, control surfaces deflection angles can be defined in this step. Then, along with the panel model, all the information is translated into the form recognizable by ZONAIR, and input files are generated. The user can then choose to either run the analysis automatically for all the defined flight conditions, or manually, externally to the framework, through ZONAIR's console. Lastly, once the analysis is complete, the results can be visualised in a form of distributions of pressure coefficient or Mach number around the model (Figure 5.2), and graphs of the coefficients (Figure 5.3).

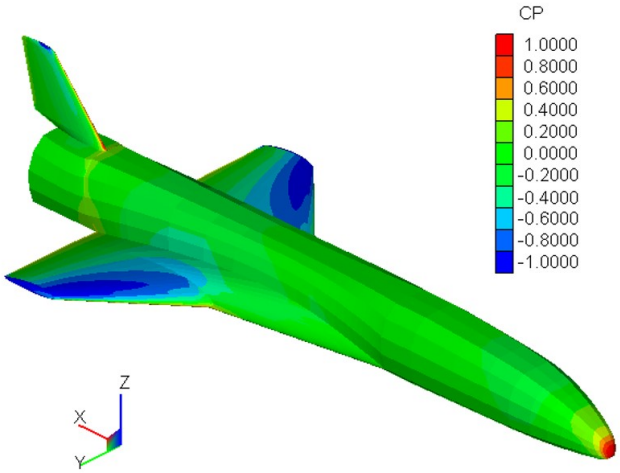
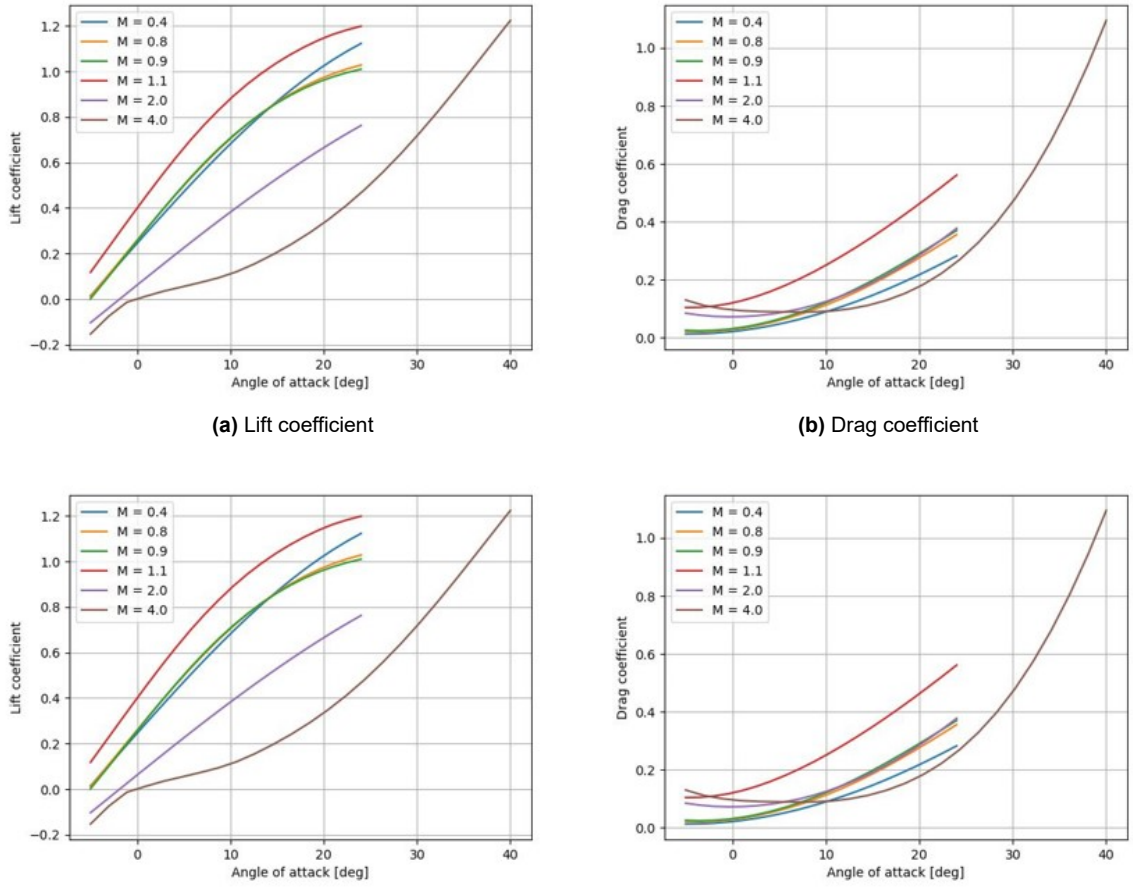


Figure 5.2: Example of pressure coefficient distribution around the body calculated by ZONAIR.



(a) Lift coefficient

(b) Drag coefficient

(c) Pitching moment coefficient

(d) Lift-over-drag ratio

Figure 5.3: Examples of visualisation of the aerodynamic coefficients calculated by ZONAIR available in the framework.

In the *Fluid Domain Branch*, the user can create the fluid domain around the analysed vehicle. The external shape of the domain is assumed as a box with rounded front and side faces (Figure 5.4). The user only defines the size of the domain through scaling parameters that refer to the size of the defined spaceplane geometry. The external shape of the spaceplane is automatically extracted from the fluid domain. Then, it can be exported as a STEP-file to be further pre-processed for the CFD calculations. If the user prefers to modify the geometry of the spaceplane using some other, external tool, it can also be exported alone as a separate STEP-file. Thus, the framework allows to export either the fluid volume around the spaceplane or the spaceplane itself.

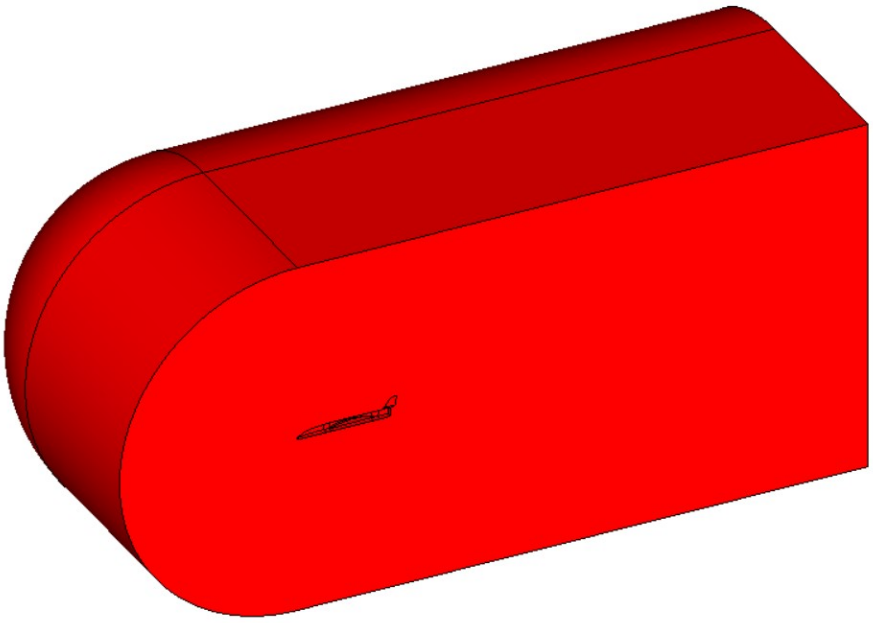


Figure 5.4: A sample shape of the fluid domain that can be exported from the framework.

5.2. Geometry parametrization

This section presents the parametrization implemented to generate a wide variety of different exterior shapes characteristic to spaceplanes. The goal was to create a parametrization providing high flexibility of possible geometries, while adhering to a few *loosely defined* guidelines:

- separate the geometry into components representative of external shapes observed in spaceplanes, i.e., fuselage (and other body-like components), wings (and other lifting surfaces)
- maintain high intuitiveness of selected variables, e.g., lengths, radii, angles, and not direct coordinates of points (e.g. defining cross-sections), or coefficients
- keep the number of parameters low
- implement functions that would prevent impossible geometries from being created

In order to achieve a robust parametrization that would satisfy the above-mentioned points, an extensive study of spaceplanes and similar aircraft and spacecraft was realized. The most common characteristics of the overall geometry, and the geometry of the components was identified and described with the use of simple and intuitive parameters. An example of a parametrization of an aircraft's external geometry proposed by De Marco et al. [100] was used as an inspiration for the parametrization ultimately implemented in the framework.

The geometry consists of two kinds of elements: body (e.g. fuselage) and lifting surfaces (e.g. wings). A body component can consist of three parts: the front, the middle and the rear part. The front and rear parts are described by three curves that define their sideview, topview and the cross-section. The middle part is just a straight connector between the two parts, and takes the cross-section and length as the only variables. The parameters used to define these curves are best explained visually. Figures 5.5-5.8 show the parameters that define the geometry of a body component. The framework also allows for import of point coordinates to define the fuselage curves, the same way as airfoils are traditionally defined. This way, the user does not need to follow this predefined parametrization if the fuselage curves of the baseline design were defined in a different way.

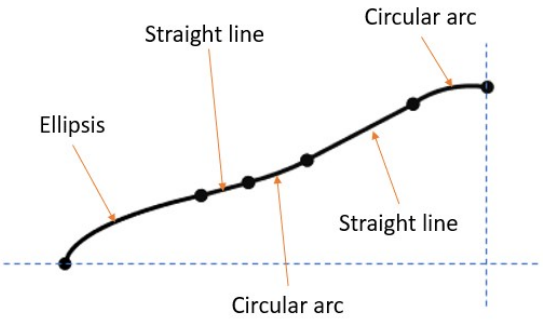


Figure 5.5: Definition of the sideview/topview curve used to generate front fuselage and rear fuselage parts.

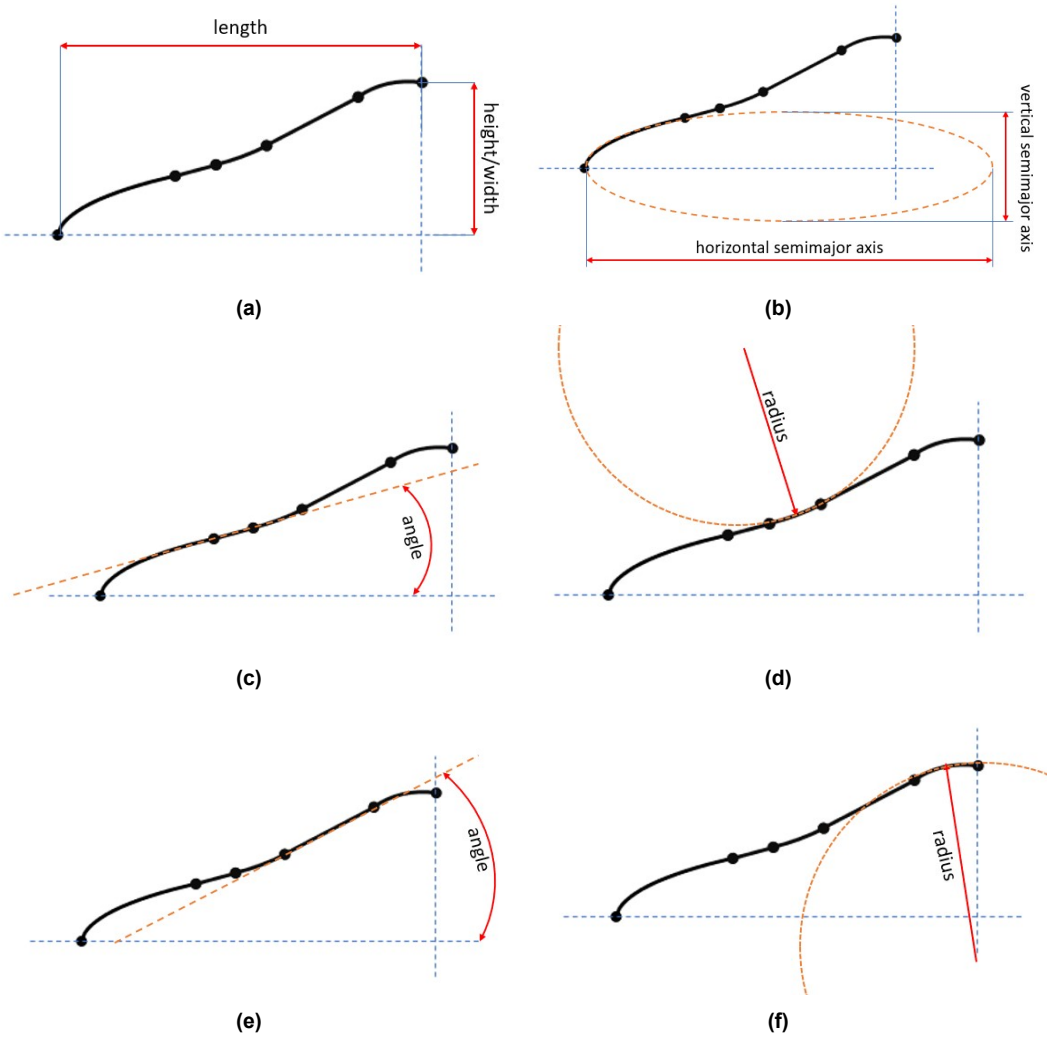


Figure 5.6: The parameters of the segments of the sideview/topview curve.

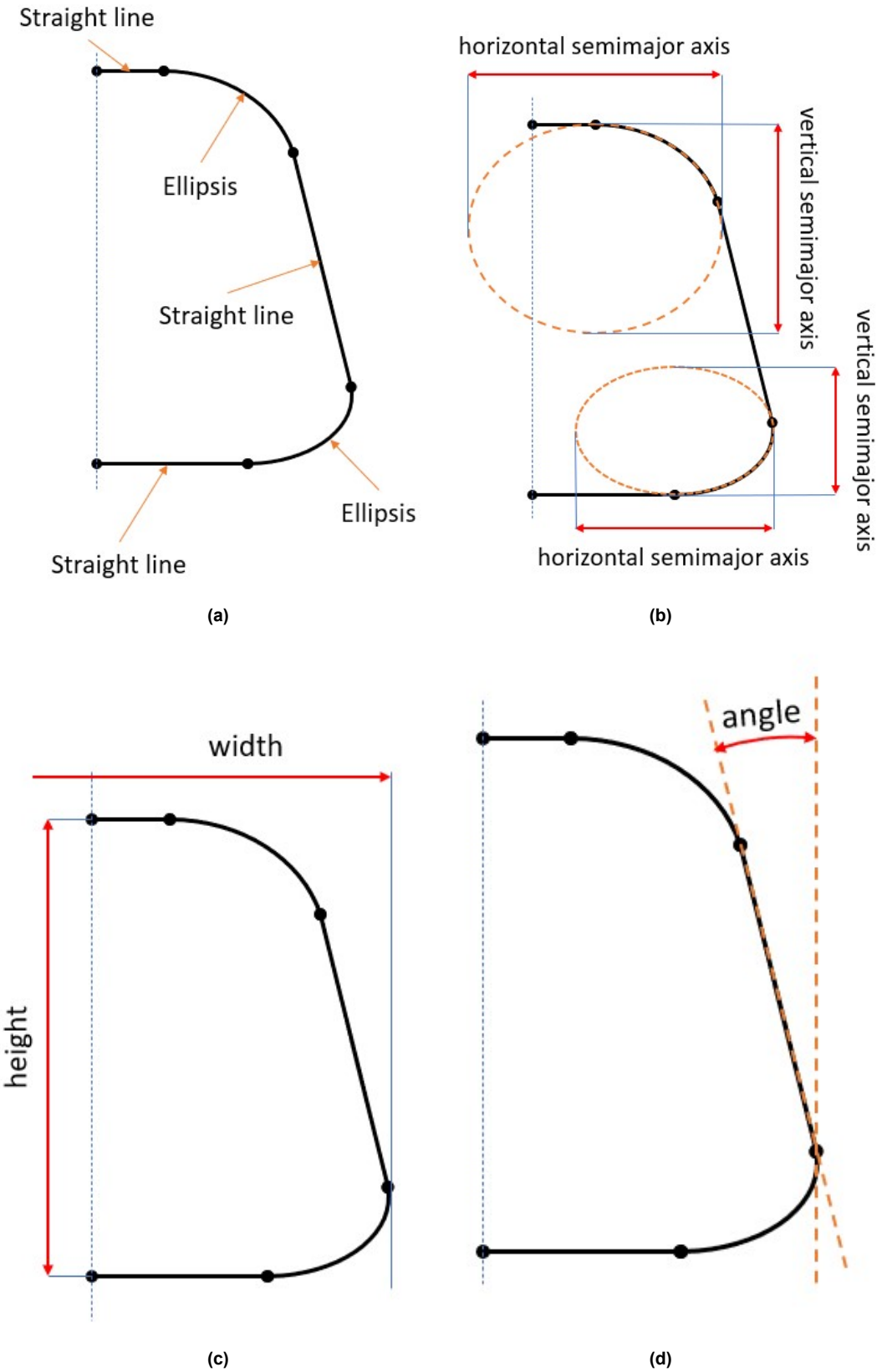
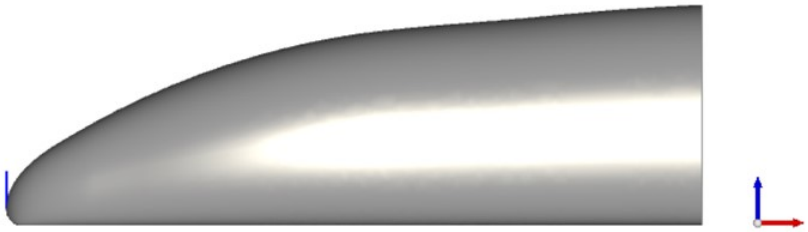
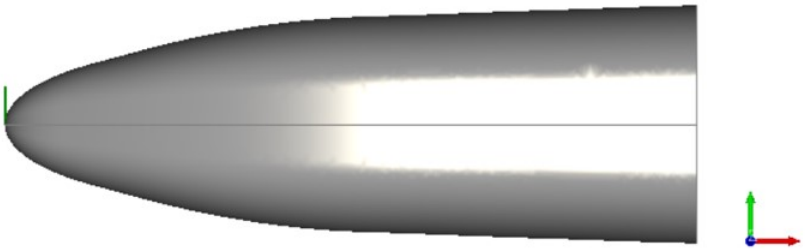


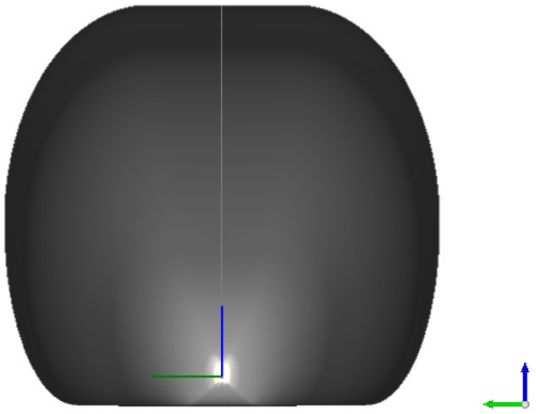
Figure 5.7: Definition and the parameters of the segments of the cross-section curve.



(a) Sideview



(b) Topview



(c) Cross-section

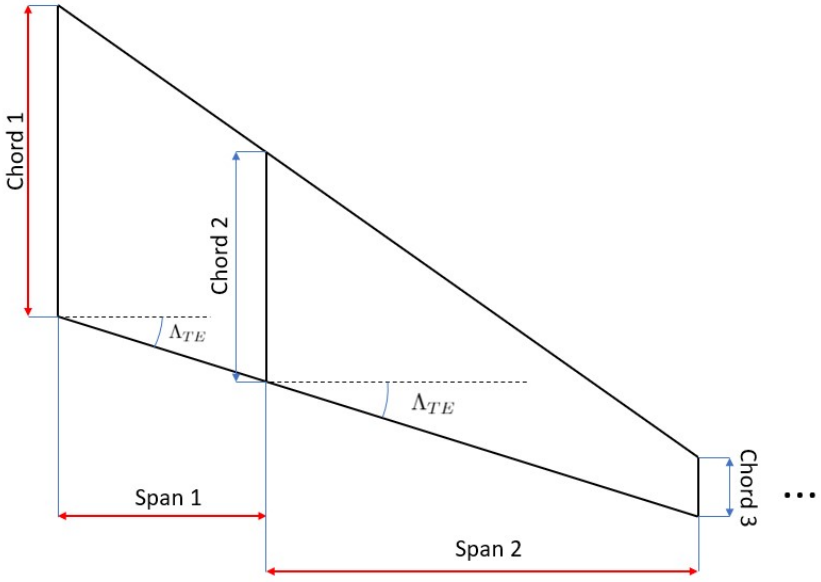
Figure 5.8: Example of a front fuselage part.

A lifting surface is defined by multiple segments that take the traditional parameters to describe their geometry. The number of segments is defined implicitly from the number of parameters provided. Apart from the planform, twist and dihedral, this shape is also characterized by airfoils at the ends of each segment. The framework allows for a baseline airfoil to be uploaded and then manipulated from within the framework. XFOIL, a 2D subsonic panel code for airfoil analysis, is used for this manipulation. Figure 5.9 shows the parameters that define the shape of a lifting surface.

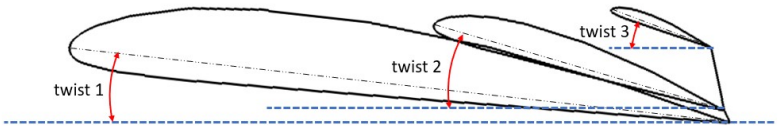
Each lifting surface can also be equipped with one of the three available wingtip devices: a rounded leading edge, a rounded tip, or a winglet (Figure 5.10). Rounded leading edge does not take any variable parameters. It is defined by a circular arc tangent to the leading edge, and a straight trailing edge with the same sweep angle as the last segment of the lifting surface. Rounded tip takes the *control span* as the only parameter. The control span is the distance between the last defined airfoil and the control points used to define the Brazier curve of the rounded tip planform (the actual span is therefore marginally smaller). The points are automatically defined in such a way so as to ensure tangency at the leading and trailing edges.

Even if the actual designed configuration is not equipped with this device, it is particularly useful for modeling for panel methods. Such codes might have problems with sharp angles between adjacent panels, which is prevented by employing a rounded tip. The last device, winglet, takes the usual winglet parameters for the shape definition. These are shown in Figure 5.11.

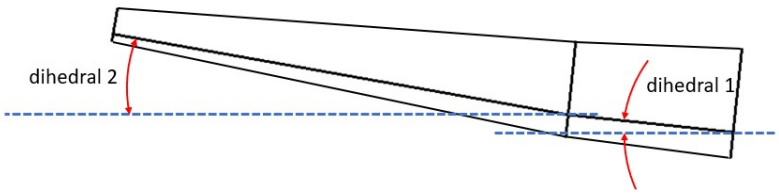
On top of that, each segment can be equipped with a control surface that can be defined by its span (as a percentage of the segment's total span), its chord-lengths (as percentages of chord-lengths at the end-sections), and its location on the segment (Figure 5.12a). Each such control surface can also be given a deflection angle (Figure 5.12b). It is frequently the case that the designers want the hinge-lines of the control surfaces to be perpendicular to the fuselage. Hence, an optional algorithm was implemented in the framework that takes only the chord-length of the inboard section, and automatically calculates the chord-length of the outboard section so that the hinge-line remains perpendicular to the fuselage.



(a) Planform

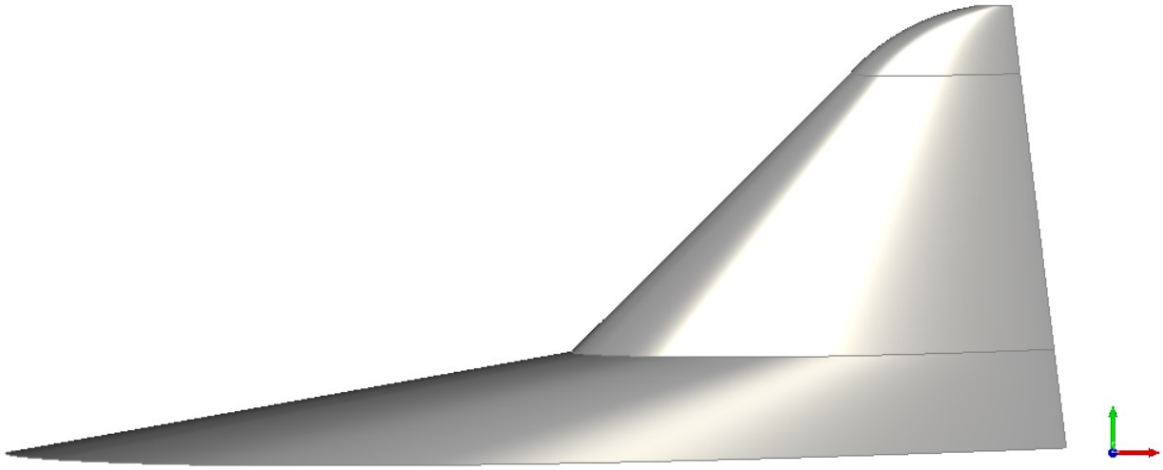


(b) Twist

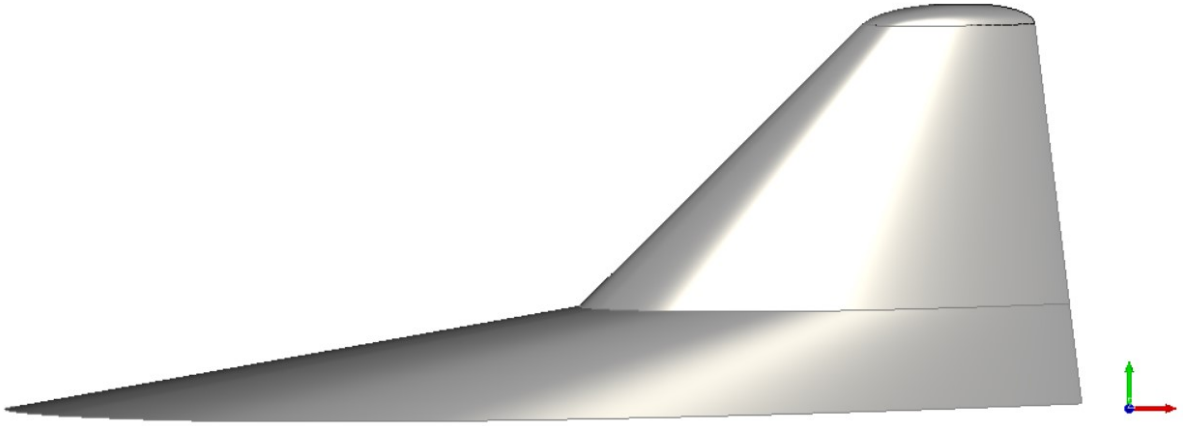


(c) Dihedral

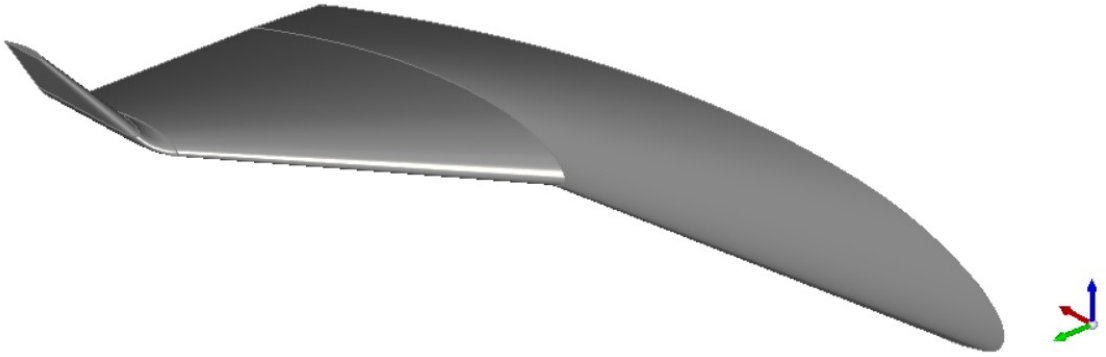
Figure 5.9: Parameters used to describe the geometry of a lifting surface component.



(a) Rounded leading edge



(b) Rounded tip



(c) Winglet

Figure 5.10: Wingtip devices available in the framework.

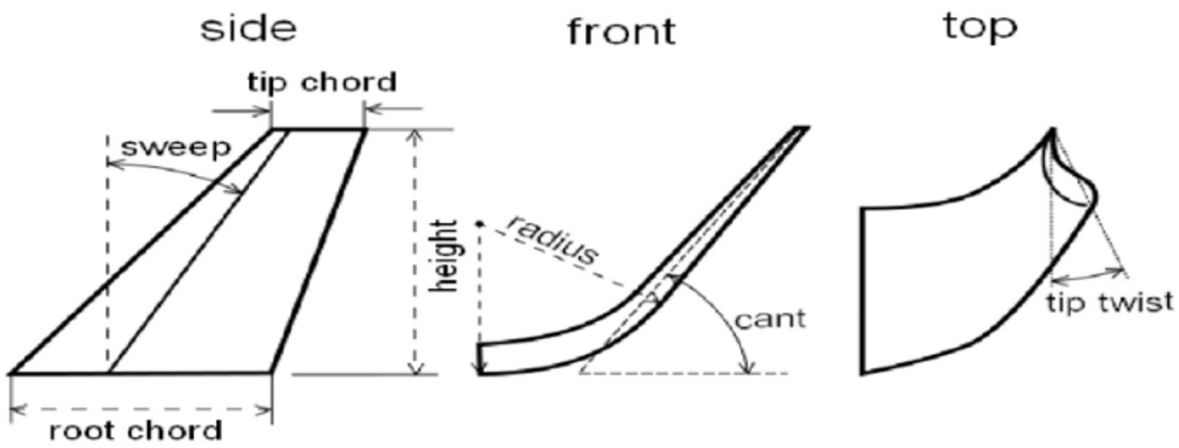
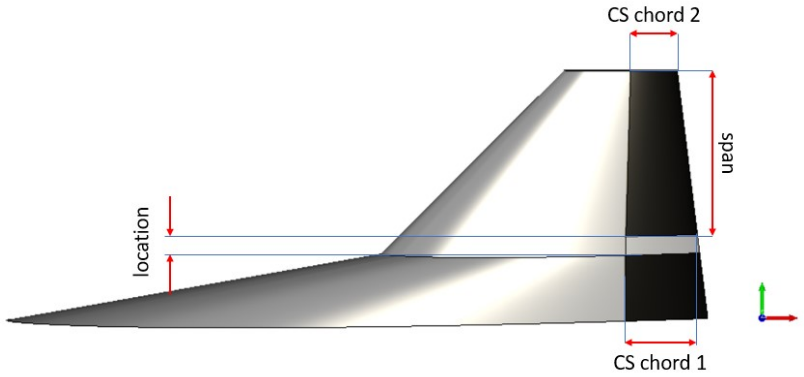
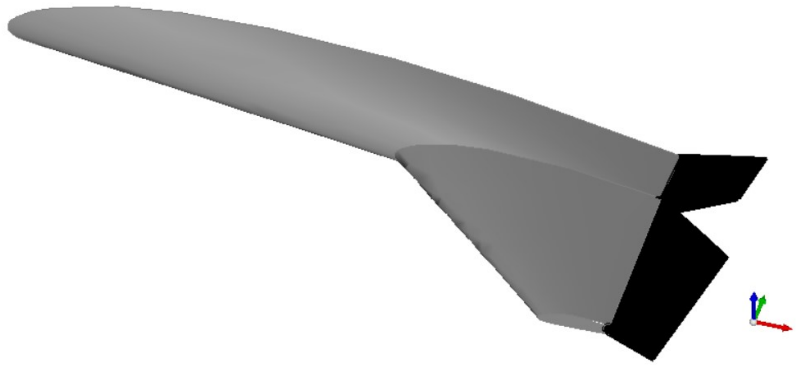


Figure 5.11: Parameters that define the shape of a winglet. [101]



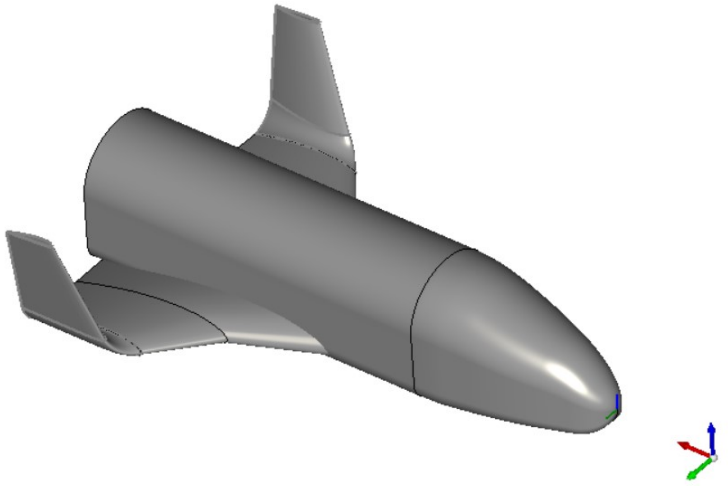
(a) Control surface parameters



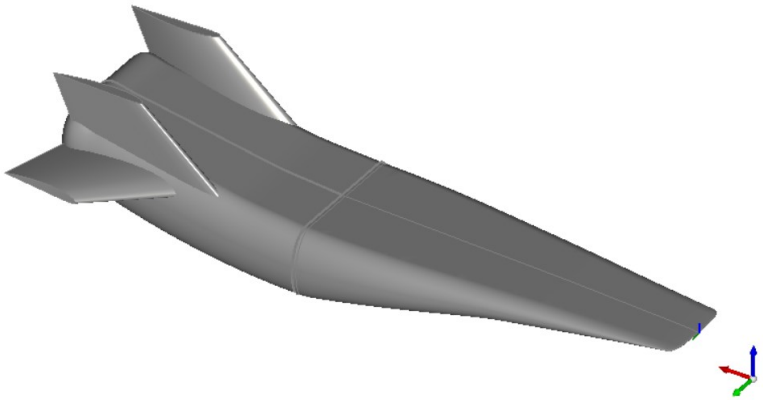
(b) Deflected control surfaces

Figure 5.12: An example of a lifting surface equipped with control surfaces.

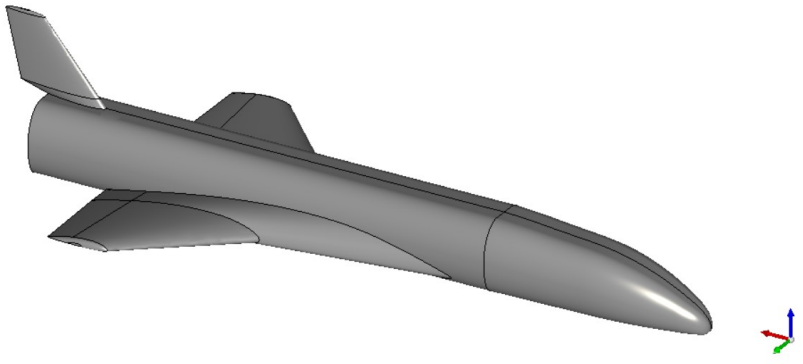
To demonstrate the flexibility of the proposed parametrization, a couple of selected geometries are presented in Figure 5.13.



(a) *HORUS* inspired geometry.



(b) *The X-43* inspired geometry.



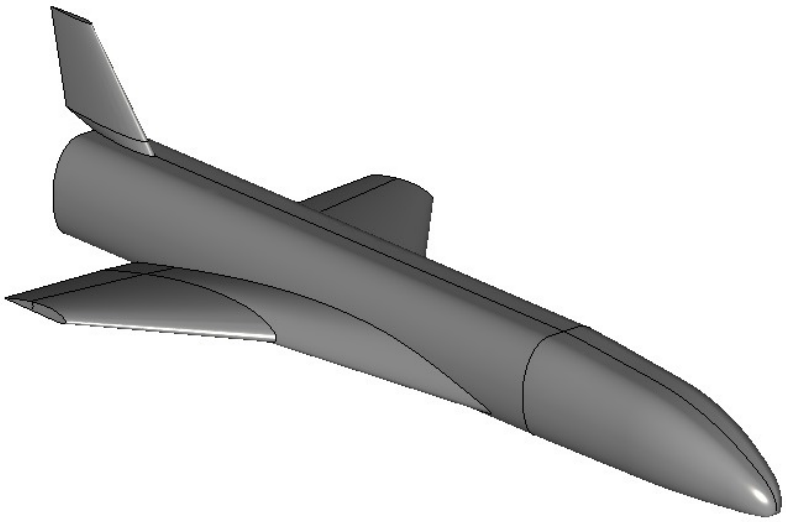
(c) *The X-34* inspired geometry.

Figure 5.13: Different configurations created using the proposed parametrization.

5.2.1. Modeling the X-34

The Orbital Sciences X-34 was already introduced in **Section 1.2**, the spaceplane chosen as the reference vehicle for this project. This section focuses on the geometry of *the X-34*, and discusses the differences between the real-life shape and the computer model created using the developed framework. Figure 5.14 presents the side-by-side comparison of the real-life geometry of the vehicle with its model.

(a) The Orbital Sciences X-34. [28]



(b) Computer model of the X-34.

Figure 5.14: Side-by-side comparison of the real-life X-34 and its computer model used in this project.

The biggest differences between the two shapes can be observed in the geometry of the fuselage. The real-life object has a varying cross-section along the plane's longitudinal axis, whereas for the model it was kept constant. The real plane's cross-section in the front is almost triangular with rounded corners, and the cross-section of the rear part is rectangular. Even though the framework makes it possible to vary the cross-section, this feature of the geometry was deemed unnecessary for the purpose of this project, which also simplified the modeling process.

Another big difference lies in the geometry of the wing. To the author's knowledge, there is no information publicly available about the twist angles, nor about the airfoils. The only airfoil-related characteristic found in the literature states that the bottom of the wing is kept flat, and the airfoils' relative thicknesses at the three characteristic sections are specified [31]. Therefore, the selected base-airfoil was the flat-bottomed Clark Y, with its thickness and camber manipulated to comply with the information found. Similar principles were followed when modeling the vertical tail. And, finally, no body-flap was modeled.

The general dimensions, such as the length, width, and height of the fuselage, as well as the wing and the vertical tail's planforms, were kept exactly as in the real-life model. The most important dimensions of *the X-34* are presented in Table 5.1.

Table 5.1: General dimensions of the X-34 (the reference dimensions for the aerodynamic coefficients are marked).

Dimension	Value	Unit
Wing area	**33.21**	$\boldsymbol{m^2}$
Wingspan	8.463	m
Mean aerodynamic chord	**4.432**	$\boldsymbol{m}$
Body length	16.43	m
Body width	1.85	m
Body height	1.7	m
Total length	17.6	m
Total height	3.61	m
CG x-coordinate *	**10.668**	$\boldsymbol{m}$
CG z-coordinate *	**0.85**	$\boldsymbol{m}$

* The framework's coordinate system is defined as originating at the nose of the body, the x-axis is pointed aft, the y-axis is pointed right (from the pilot's view), and the z-axis is pointed up.

5.3. Low-fidelity analysis module

This section is focused on the low-fidelity aerodynamic analysis module. The low-fidelity software chosen for this task is an advanced panel code - ZONAIR. In the **Section 5.3.1**, the decision-making and reasoning behind the choice of the selected software is explained. Followed by **Section 5.3.2**, which briefly presents the calculation methods applied in ZONAIR, and their limitations. **Section 5.3.3** discusses the geometry discretisation used to create the input files for ZONAIR.

5.3.1. Software selection

This section summarises the results of the software search. Some programs that were also considered for the task, but not included in this section due to their early dismissal, are *APAME - 3D panel method*, *AVL*, *Tornado*, *PANUKL*, *Tucan*, *Aeolus ASP*, *VSAERO*, and *OpenVSP*. All of them were discarded early on in the framework definition due to various reasons, such as, difficult to automate input file formats, very limited speed range, or licensing problems, etc. The programs presented in this section are the most promising ones considered for the task at hand. A short description, followed by a table presenting a comparison of these programs' capabilities (Table 5.2) can be found below.

The following programs were studied as potential candidates for the task:

- ***CBAero*** (Configuration-Based Aerodynamics) - a tool developed by NASA to predict the aerothermodynamic behaviour of conceptual aerospace configurations. The geometry needs to be defined by unstructured triangulated surface panels. The program uses a multi-pole panel code coupled with a streamline tracing scheme for subsonic Mach numbers, and for super- and hypersonic flows different panel methods, also coupled with the streamline tracing scheme, attachment line detection and stagnation-attachment line heating models are used. The software is under NASA Release Type: U.S. Government Purpose only (also available to academia under certain circumstances). [102]
- ***FlightStream*** - a surface vorticity flow solver for incompressible and compressible subsonic and transonic aerodynamics. It uses an unstructured surface mesh, removes the necessity of generation of the volume mesh, and claims to eliminate mesh-dependency on flow-field solutions and stability. Currently, research is being conducted to enable supersonic flow calculations. [103]
- ***DATCOM*** (Data Compendium) - a program using methods from USAF Stability and Control DATCOM to predict static stability, control and dynamic stability derivatives for relatively simple geometries. Simplified analytical and empirical methods are employed to calculate an aircraft's performance in all speed regimes. [104]
- ***PANAIR*** (Panel Aerodynamics) - a computer software developed for inviscid sub- and supersonic flow calculations around arbitrary aircraft geometries with a higher order panel scheme. The program solves linear PDEs of a configuration approximated by a set of surface panels. Only potential flows can be solved with this program. [105]
- ***S/HABP*** (Supersonic/Hypersonic Arbitrary Body Program) - a tool that serves to calculate super- and hypersonic aerodynamic characteristics of complex aircraft configurations using engineering level methods. Inviscid flows can be solved using one of many available surface inclination or small-perturbation methods, such as Modified Newtonian, Tangent-Wedge/Tangent-Cone or Shock Expansion. For viscous analysis, skin friction is incorporated into the calculations using Reynolds-number-based skin friction values, and skin friction data from boundary layer properties calculated from generated streamlines, which is then interpolated to the panels. [106]
- ***ZAERO*** - an engineering software that integrates all disciplines necessary for advanced aeroelastic calculations. For the purpose of this analysis only the module for aerodynamic calculations is considered. Its high-fidelity geometry module allows for the solution of flows around complex geometries. The unified Aerodynamic Influence Coefficient matrix scheme provides the capability to solve flows with all Mach numbers using a single aerodynamic model [107].
- ***ZONAIR*** - an application for calculations of flight loads that can include aeroelasticity and aerothermoelasticity. A unified high-order sub-/super-/hypersonic panel method is used for aerodynamic analysis of arbitrary aircraft shapes. Similarly to *ZAERO*, *ZONAIR* uses an unstructured panelling scheme that is compatible with various commercial FEA pre- and post-processors. Additionally, it employs a vortex roll-up scheme for high-angle-of-attack aerodynamics [108], [109].

Some aspects of the software included in Table 5.2 (aeroelasticity, aerothermodynamics) were not necessarily relevant for the main task of this thesis work, therefore did not play a significant role in the selection process. However, if the presented framework is ever developed further, these capabilities might be appreciated. This is further discussed in **Section 7.2**.

Inclusion of programs that do not cover the entire spectrum of velocities expected on a typical spaceplane trajectory is dictated by the fact that they could be used in tandem with other programs. For example, a *FlightStream* + *S/HABP* combination would allow for calculations at all the necessary Mach numbers. However, the decision was made that a single software used for all speed regimes would drastically simplify the design of the framework. Hence, this option was ultimately dismissed. In the final stage of the selection process, the options were narrowed down to three choices: *CBAero*, *ZAERO* and *ZONAIR*. Eventually, *CBAero* had to be eliminated from the consideration as it was only available for U.S. Government purposes or **U.S.** Academia. Between the other two programs *ZAERO* seemed the better option considering its capability of transonic aerodynamics calculations. However, ultimately it was decided to go with *ZONAIR* given that the author had previous experience with the software, what made its implementation into the framework much faster.

Table 5.2: A brief comparative study of characteristics of various programs considered. Marked with thicker lines is the ultimately selected software. [102], [103], [104], [105], [106], [107], [108], [109]

	CBAero	FlightStream	DATCOM	PANAIR	S/HABP	ZAERO	ZONAIR
Speed regime (subsonic/ transonic/ supersonic/ hypersonic)	All	Subsonic & transonic	Subsonic/ supersonic/ hypersonic	Subsonic & supersonic	Supersonic & hypersonic	All	Subsonic/ supersonic/ hypersonic
Method used	Panel methods + streamline tracing	Surface vorticity sheets and vorticity-based loads	Analytical/ empirical	Potential higher order panel method	**Inviscid**: engineering methods **Viscous**: Re-dependent skin friction or BL calculated from generated streamlines	6 high-order panel methods(1)	Unified high-order panel method
Geometry high fidelity	Yes	Yes	No	Yes	Yes	Yes	Yes
High AoA	Yes	Yes	Yes	No	Yes	No	Yes
Multi-body interference	Yes	Yes	No	Yes	Yes	Yes	Yes
Computational time(2)	Unknown(3)	Unknown(3)	«10min	~20min	Unknown(4)	< 10min(5)	~10min(5)
Aeroelasticity	No	No	No	No	No	Yes (static & dynamic)	Yes (static)
Aeroheating	Yes	No	No	No	Yes	Yes	Yes

(1) **ZONA6**:Subsonic unsteady aerodynamics, **ZSAP**: Sonic Acceleration Potential Method, **ZTAIC**:Transonic Unsteady Aerodynamic using a Transonic Equivalent Strip Method, **ZTRAN**: Transonic Unsteady Aerodynamics using an Overset Field-Panel Method, **ZONA7**: Supersonic Unsteady Aerodynamics, **ZONA7U**: Hypersonic Unsteady Aerodynamics

(2) Estimated for a typical personal laptop

(3) Expected relatively short (similar order of magnitude to ZONAIR)

(4) Expected relatively long (significantly longer than ZONAIR)

(5) Time estimated per Mach number for a model of ~2000 panels. The software takes around 5-6min to create the AIC for a specific Mach number, and then each AoA calculation is in the order of seconds.

5.3.2. ZONAIR calculation methods

This section briefly presents the calculation methods implemented in the software ultimately chosen as the low-fidelity tool. All the information presented in this section is based on *ZONAIR* User's manual [108], and *Panel methods: An introduction* by Erickson, 1991 [83].

The main equation solved by *ZONAIR* is the Prandtl-Glauert equation. For steady subsonic flows, this equation can be written as Equation 5.1, and for supersonic cases as Equation 5.2. In both equations M_∞ is the freestream Mach number and ϕ_{ii} is the perturbation velocity potential in a particular direction.

$$\nabla^2\phi = \left(1 - M_\infty^2\right)\phi_{xx} + \phi_{yy} + \phi_{zz} = 0 \tag{5.1}$$

$$-\nabla^2\phi = \left(M_\infty^2 - 1\right)\phi_{xx} - \phi_{yy} - \phi_{zz} = 0 \tag{5.2}$$

These are the simplest forms of the equations modeling compressible fluid flows. They are obtained by neglecting flow characteristics such as viscosity, heat transfer, and rotationality of the flow. All nonlinear terms are also neglected. These simplifications make the flow inviscid, irrotational and linear. However, it is possible to activate an option that includes these viscous effects in the following way. From the stagnation point, streamlines are computed to each panel in the model. Then, after the initial calculations, a two-dimensional boundary method is applied along each of these streamlines. This method introduces a viscous correction which slightly increases the fidelity of the model.

Other included phenomena are, for example, lift-induced- and wave drag. Wave drag is able to be predicted since Equations 5.1 and 5.2 do admit approximate weak-shock solutions with shock-expansion theory.

A panel model for *ZONAIR* needs to be built from discrete grid points that are connected to form triangular or quadrilateral panels. Both types of elements can be present in the same model. Continuity of the distribution of singularity over the panels of the whole model is ensured by unit singularity strength at each grid point. Then, this singularity is piecewisely linearly distributed over the panels around the grid point. Resulting from the superposition of the elementary distribution of singularities is a continuous singularity distribution over the entire model. Figure 5.15 shows the unit strength singularity distribution at a grid point. Moreover, since the four grid points defining a quadrilateral element are not necessarily co-planar, each such element is subdivided into six triangular elements as shown in Figure 5.16. This assures the continuity of the geometry of the panel model.

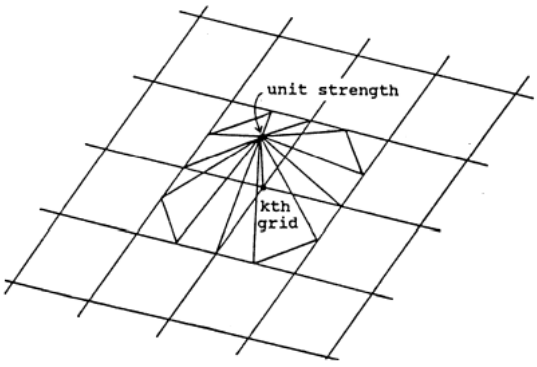


Figure 5.15: Elementary singularity distribution at a grid point. [108]

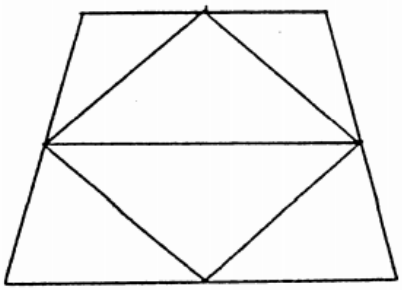


Figure 5.16: Subdivision of a quadrilateral element. [108]

Dirichlet and Neumann boundary conditions, respectively $\phi_L = 0$ and $\frac{\partial \phi}{\partial n} = -\vec{V} \cdot \vec{n}$ are imposed on each panel to solve the source and doublet strengths. Furthermore, no-force condition $\frac{\partial \phi}{\partial x} = 0$ is imposed on the wake for the wake-condition to be satisfied. This is illustrated in Figure 5.17. Inflow and outflow boundary conditions can be included to model inlet and nozzle effects. Transpiration boundary conditions are used for control surfaces deflection.

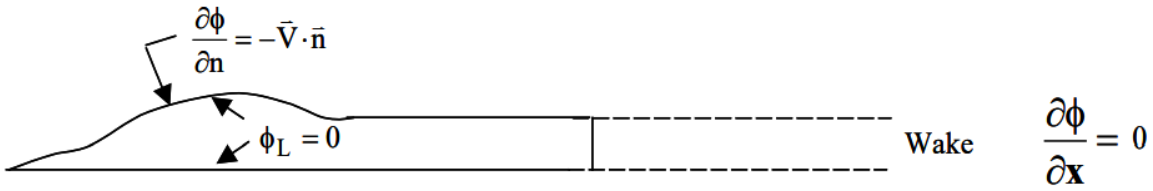


Figure 5.17: Boundary conditions imposed on body panels and the wake surface. [108]

Unlike most other panel-method-based solvers, *ZONAIR* does not require explicit wake modeling. It is sufficient to specify line segments along trailing edges of lifting surfaces or bodies where the wake surface is to begin. Such wake surfaces, as shown in Figure 5.18, sweep to infinity creating a flat wake surface. Integration from the specified line segment towards infinity allows for an exact solution. Thus, the influence of the wake can be included in the results by calculating the exact integral along each line segment.

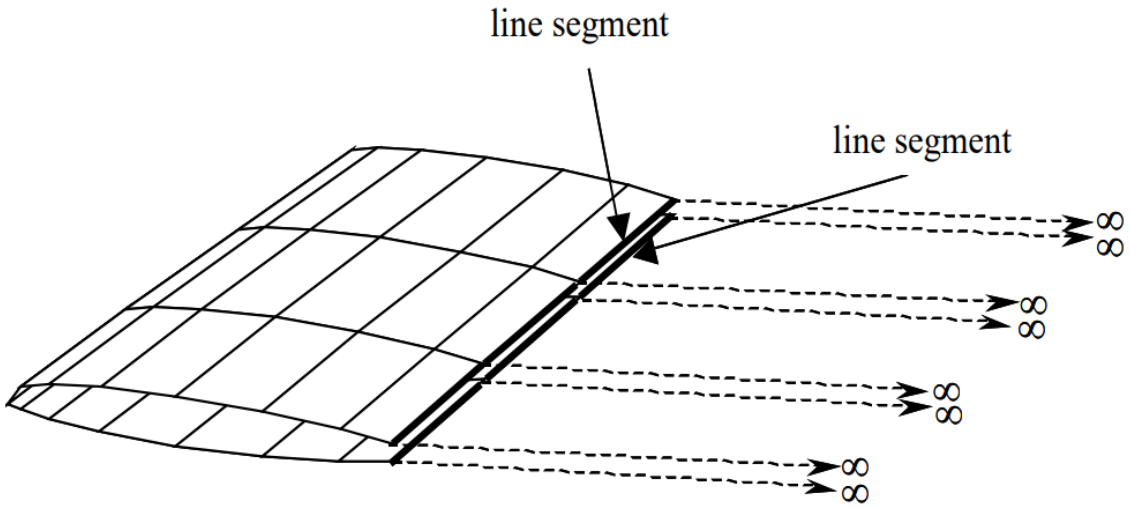


Figure 5.18: Line segments for wake modeling. [108]

When it comes to the calculations in a supersonic flow regime, a linearized supersonic flow formulation defines the so-called domain of influence determined by the freestream Mach number. Most supersonic panel methods generally only work if the entire body is submerged in this region, but if any part of the model is exposed outside of it (Figure 5.19), they no longer work properly. Such panels are called *super-inclined*, i.e., the angle of incidence on these elements is larger than the Mach-cone angle. To avoid this numerical singularity that would appear in most traditional panel codes, *ZONAIR* uses the corresponding oblique shock angle for a cone which is based on the Exact Euler Conical-Flow Solutions. The oblique shock angle is calculated from the angle between the freestream velocity vector and the inclination of the panel. It is then used to compute a modified Mach wave to position the new region of influence slightly ahead of the panels that would normally be outside of this zone.

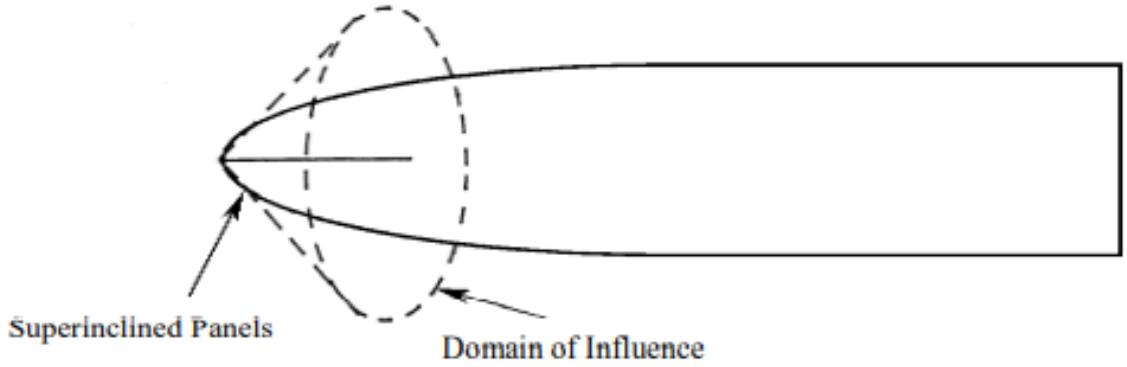


Figure 5.19: Domain of influence and super-inclined panels. [108]

5.3.2.1. ZONAIR limitations

Panel methods were already introduced and discussed in **Section 3.2.2.3**. And, even though *ZONAIR* is a relatively advanced panel code, it shares very similar limitations. This section presents some of the most common limitations of the majority of panel codes, in particular *ZONAIR*.

As already mentioned in **Section 5.3.2**, the Prandtl-Glauert equation (Equations 5.1 and 5.2) neglects some important flow characteristics and hence restricts physics of the flow, namely [83], [110], [111], [112]:

1. Skin-friction drag is not correctly calculated. Even though a correction is applied to account for the viscous effects and yields better results, the accuracy of this method is still not particularly high.
2. Modeling separation is problematic.
3. Due to their highly nonlinear nature, transonic shocks might not be correctly predicted.
4. Phenomena such as junction flows (**Section 1.3.1**), shock-boundary-layer interactions (**Section 1.3.2**) and shock-shock interactions (**Section 1.3.3**) cannot be modeled with this type of solver.

ZONAIR is expected to abide these same limitations, which can be expected to have major influence on the results of the simulations. At this moment it is also important to expand on what was already mentioned in **Section 5.3.2**. As visualised in Figure 5.20, in subsonic flows skin-friction drag typically constitutes around 50% of the total drag, while in supersonic cases this contribution falls to around 20%. Since this is the only major drag component that is not properly modeled in *ZONAIR*, it can be expected that results of the drag coefficients will be relatively more accurate for high-speed simulations.

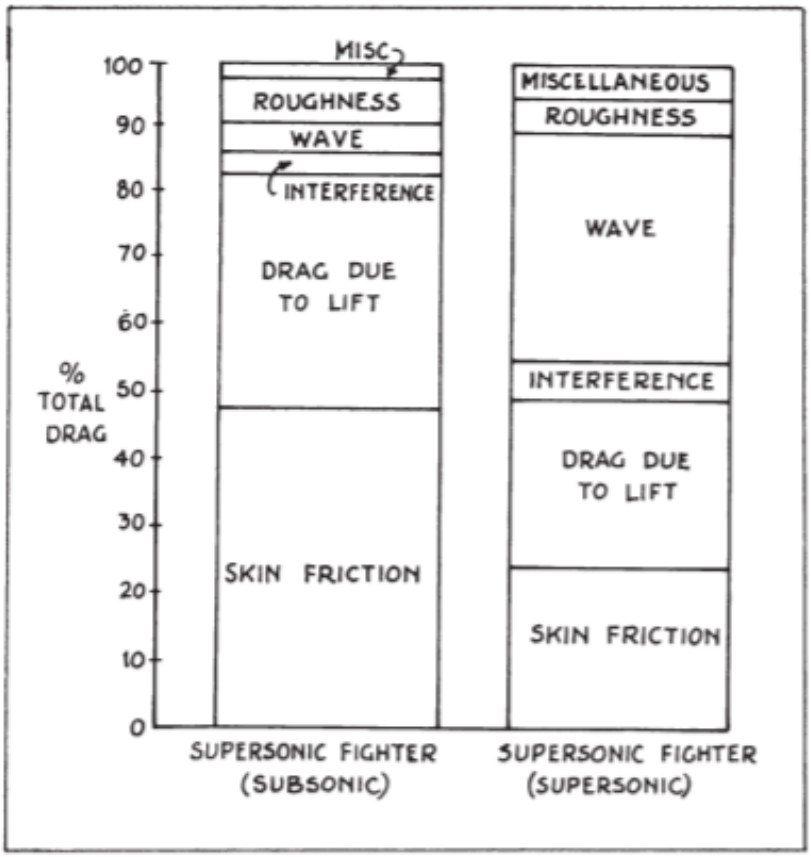


Figure 5.20: Approximate drag breakdown of a typical fighter aircraft. [113]

5.3.3. ZONAIR geometry

This section presents the general guidelines for geometry definition for *ZONAIR*. As is typical for panel methods, one geometry can be used for all the calculations independently of the analysed flight conditions (Mach number, and incidence angles). The density of panels in particular regions is mostly dependent on the geometry of the actual aircraft, but expected flow characteristics should also be taken into consideration.

A well-constructed panel model should well reflect the actual geometry of the aircraft, i.e., higher density of the panels is required where the curvature is high, e.g., at the leading edges of wings and the vertical tailplane, the frontal section of the fuselage. On top of that, more panels should also be placed in the regions where higher velocity gradients are expected, e.g., at the trailing edges, near the wing-tips. Furthermore, high angles between adjacent panels should be avoided. For example, it is better to avoid flat wingtips. It is recommended to apply so-called rounded wingtips even if the analysed aircraft has actually flat wingtips. This is to avoid high velocity gradients, that could produce infinite velocity. This guideline was followed during the development of the framework, which is one of the reasons why the rounded tip device was created (**Section 5.2**).

Once the model discretization is complete, an algorithm is automatically run to identify correct nodes and panels at the characteristic locations of the model. This algorithm finds the co-ordinates of the elements of the panel model that represent the regions such as wing trailing edges, fuselage trailing edge, wing-fuselage junctions, wingtips, etc. The framework automatically translates all this information into a *ZONAIR*-compatible format. Then, suitable algorithms are run internally by *ZONAIR* to create adequate wake-models.

5.4. High-fidelity analysis

As discussed in the very beginning of the definition of the framework, the high-fidelity analysis is for the most part done externally, and only the eventual results are fed back into the framework. This section is entirely focused on the approach taken towards the high-fidelity analysis conducted during this project. **Section 5.4.1** briefly presents the software used in this process - *Ansys Fluent*. Then, **Section 5.4.2** is dedicated to the discretization of the flow domain around the vehicle. Lastly, **Section 5.4.3** discusses the set-up of the numerical simulations.

5.4.1. Ansys Fluent

In the words of the developer "*Ansys Fluent is the industry-leading fluid simulation software known for its advanced physics modeling capabilities and industry leading accuracy*"[114]. It is one of the most popular fluid flow simulation programs, widely used in the aerospace industry. It can be used for a wide variety of applications, i.a., battery modeling, electric motor cooling, multiphase flows, combustion modeling, fluid-structure interactions, and most importantly in the light of this project - turbulence modeling. A broad collection of turbulence models is available in this environment, including the most popular models, such as the ones introduced in **Section 3.2.3**.

On top of that, Ansys Fluent is constructed with a *Streamlined Workflow* capability, which helps to speed up pre- and post-processing of CFD calculations. Thanks to this option, generation of high-quality computational models is drastically simplified. Furthermore, this approach allows for a high degree of automation of the simulation process. A capability that was not explored further due to the time constraints of this project, but could be implemented in the future versions of the framework.

5.4.2. Computational domain

Unlike for panel methods, the computational fluid domain around the vehicle needs to be defined on top of the surface mesh, in order to apply high-fidelity CFD methods. The developed framework allows for exporting of either the 3D geometry of the spaceplane itself, or the fluid domain around it, depending on the user's needs. In the presented project, some further manipulation of the spaceplane model was deemed useful to prepare it for the simulations. Thus, in this instance, the vehicle's shape was exported for the manipulation in *SpaceClaim* environment, a part of *Ansys* package. In the case of the intended simplifications, *SpaceClaim* copes better with positive shapes (i.e., as seen from the outside), rather than negative ones (i.e., as seen from the inside). The changes applied to the model constituted rounding of the trailing edges of the wing, the vertical tail, and the fuselage. Furthermore, the project focuses on the longitudinal motion only, therefore it is sufficient to use the so-called *half-model*. It means, that the geometry can be cut in half along the vehicle's symmetry plane, and symmetrical flow conditions on both sides are assumed.

Then, the fluid domain was created using a simple box-shape. The dimensions of this domain can be seen in Figure 5.21. Remembering the dimensions of the vehicle itself (Table 5.1), the size of the domain is well within standard guidelines, which usually recommend the domain to be ten times the size in each of the three main dimensions of the analysed object. The fluid domain in the presented analysis stands at 23 times the total length in the longitudinal direction, 17 times the wingspan in the spanwise direction and 27 times the total height in the vertical direction. It was decided to increase the size of the domain over the suggested minimum to ensure that the domain boundaries would have no effect on the results even for the most extreme flight conditions. It was further verified during the simulations, where special attention was paid to whether the influence of the vehicle on the freestream properties dissipated before reaching the external boundaries of the fluid domain. As an example, Figure 5.22 shows the velocity field in one of the most *extreme* cases simulated during the project (i.e., with the most

far-reaching influence on the domain). As can be seen, the changes introduced into the flow field dissipate before any of the boundaries is reached.

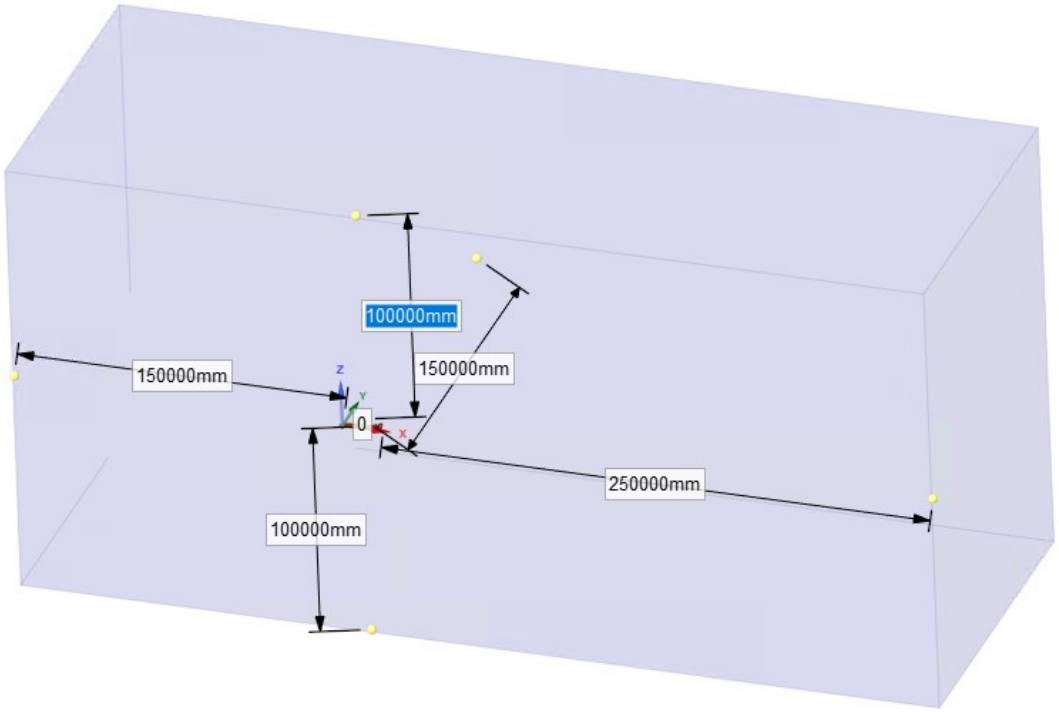


Figure 5.21: Dimensions of the computational fluid domain around the vehicle.

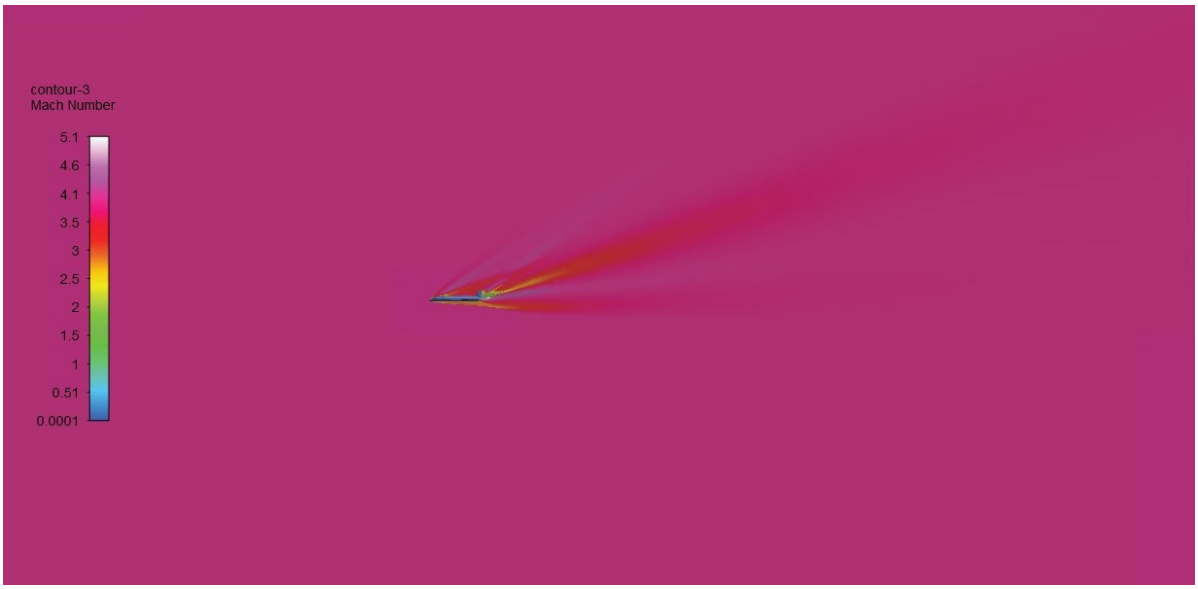


Figure 5.22: Distribution of Mach number in the flow field at Mach number of 4.0, and AoA of 20°- view of the entire domain.

Once the domain is fully defined, it needs to be correctly prepared for the simulations. For this purpose, it needs to be discretized. For the introduction to the idea of domain discretization refer back to **Section 3.2.4**. The meshing process was conducted using *Ansys Fluent Meshing* tool adhering to the guidelines regarding grid-quality as discussed in that section. The

tool has various in-built types of grids available for the user, one of which is a so-called Mosaic poly-hexcore mesh, which was used for the discretization in this project. It is a relatively new technology that promises to accelerate the process, reduce total face count and generate higher quality cells than comparable methods. It combines a polyhedral- and a hexahedral-type meshes, with the polyhedral elements serving as connectors between disparate faces. This allows to overcome one of the major simulation challenges, which is the transition between different types of elements in complex geometries. The more *flexible* polyhedral elements are found at the domain boundaries (inlet, outlet, walls, etc.), and then smoothly transition to form very regular hexahedral elements in the rest of the domain.

Most of the grid-generation is automated within this environment, what translates to a very steep learning curve and relatively few parameters to control by the user. The parameters mostly played with were the element sizes at the domain boundaries, as well as curvature and proximity conditions to control the size of the elements in the characteristic locations. On top of that, there were two so-called *bodies of influence* placed in the domain, to help refine the region around the vehicle and in the wake behind it. Lastly, 15 layers of cells were placed near the spaceplane surfaces to properly resolve the boundary layer. In the process, various meshes were created in order to reach grid-independent results (this is the subject of **Section 6.2**). This type of mesh does not use the standard mesh parameters such as skewness or orthogonality for quality control, but a single, general quality parameter. This parameter takes values between 0 and 1, and according to the guidelines of the developer, it should be no lower than 0.05. For all the meshes ultimately used in the simulations, it was kept above this suggested minimum value. Furthermore, to some cases additional refinement was applied using the automatic refinement option. This option allows to search for cells with certain flow characteristics, e.g., cells across which there is a pressure gradient larger than a certain given value. Then, an algorithm of cell divisions is run, to create a denser mesh in the region. Figure 5.23 shows cells with large pressure jump ready for refinement for the simulation at Mach number 2.0 and angle of attack 0°.

An example of one of the generated meshes can be seen in Figures 5.24-5.27, which show a grid made up of 36,140,408 elements. Figure 5.24 is the general view of the domain, where the polyhedral elements can be seen at the boundaries - large on the outside (these elements are far away from the area of interest), and very small near the analysed body. Figures 5.25 and 5.26 nicely demonstrate the transition between the two types of elements. Lastly, Figure 5.27 is a close-up on the trailing edge of the main wing of the spaceplane, where the boundary layer cells can be distinguished.

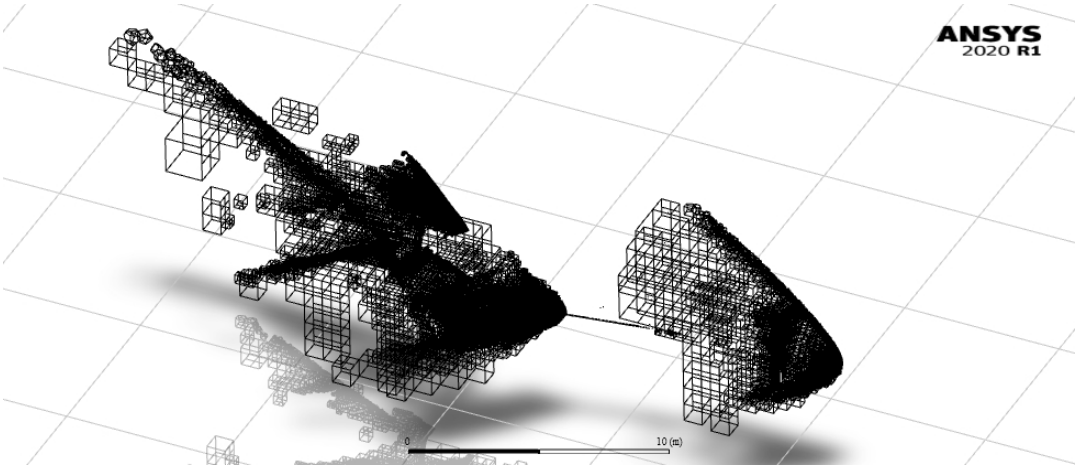


Figure 5.23: Cells with large pressure gradient at Mach number 2.0 and angle of attack 0° found using the in-built Ansys Fluent cell-refinement function.

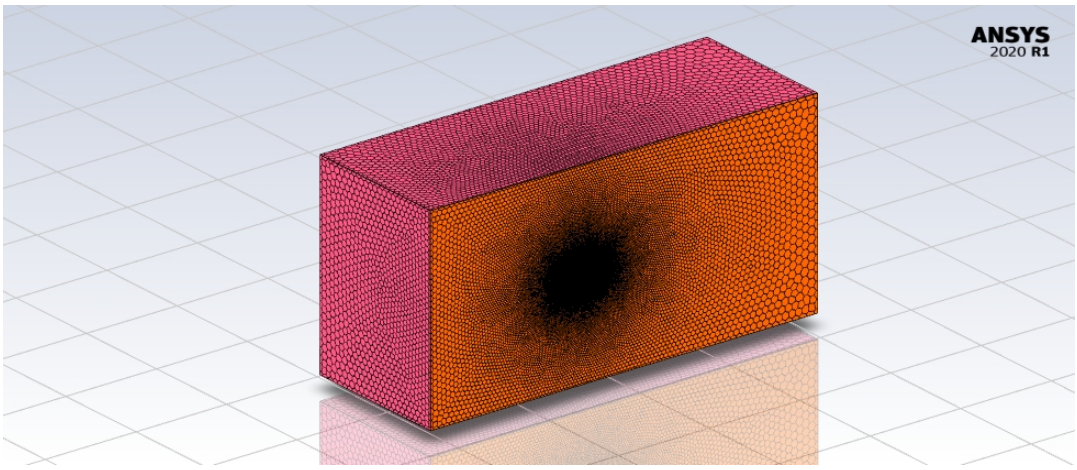


Figure 5.24: Full view of the fluid domain discretized with a poly-hexcore mesh (36,140,408 elements).

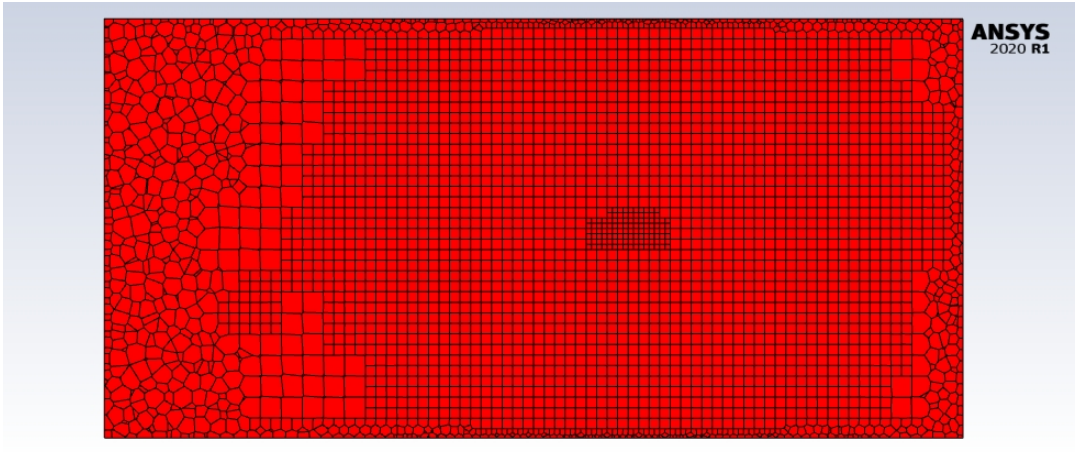


Figure 5.25: Cross-section with an XZ-plane at 10m away from the symmetry plane.

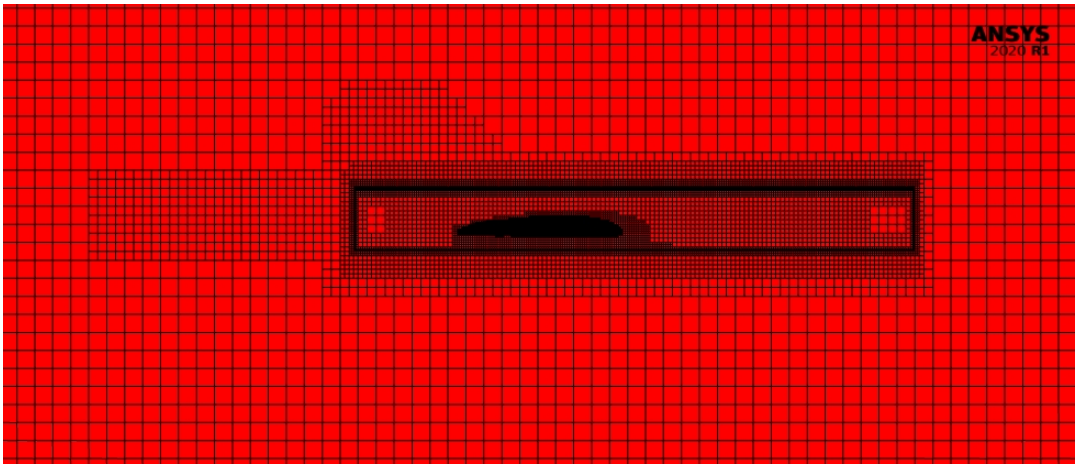


Figure 5.26: Cross-section with an XZ-plane at 1.5m away from the symmetry plane.

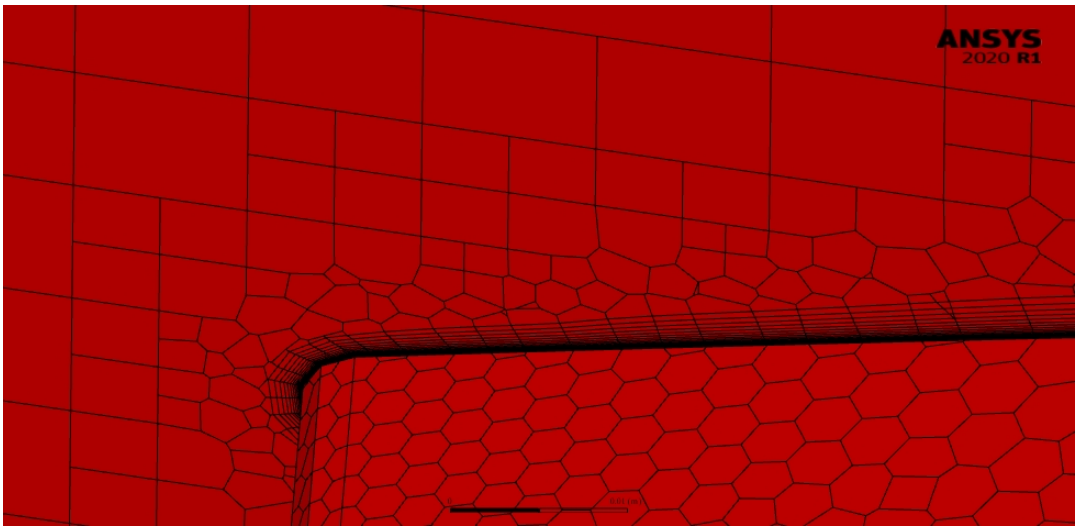


Figure 5.27: Cross-section with an XZ-plane at 3.5m away from the symmetry plane. Zoom on the trailing edge.

5.4.3. Solver setup

The last step in the preparation of the simulations is the setup of the CFD solver. This step consists of, but is not limited to, the definition of the fluid flow model (some of the most popular ones were presented in **Section 3.2.3**), discretization scheme, boundary conditions, fluid properties, etc. There is an innumerable amount of ways the settings can be chosen. These settings can have a major influence on the final results, which for different calculation methods can be very different even if a converged solution is reached. Therefore, it is paramount to choose the setup carefully and accordingly, bearing in mind the flow conditions to be simulated. The settings used in the simulations performed during the realization of this project are listed below:

- **Solver type:**
  There are two types of solvers available in *Ansys Fluent* - pressure-based and density-based. Both methods use the momentum equations to obtain the velocity field. The main difference between the two is that in the density-based solver the density field is calculated through the continuity equation, whereas the equation of state is used to compute the pressure field. The pressure-based approach manipulates the continuity and momentum equations to calculate the pressure (or pressure correction) equation and from this equation the pressure field is extracted. Traditionally, the pressure-based solver was used for subsonic, mostly incompressible simulations, and the density-based solver was developed for high-speed (and thus compressible) cases. However, recent advancements in CFD allowed both types of solvers to be used for both compressible and incompressible flows. Nevertheless, it is still usually recommended to use the density-based solver for flows where compressibility effects might play a big role (e.g., shock wave formation). Another big difference is that the density-based method solves the continuity, momentum and energy equations in a coupled manner, while the pressure-based solver uses a more segregated approach. It means that the density-based solver solves all these equations in one iteration, thus is slower and more memory intensive. This is shown with a diagram in Figure 5.28. In this project, a wide range of Mach numbers was simulated, from low-subsonic ($M = 0.2$) to high-supersonic ($M = 4.0$). To speed up some simulations, the low-speed cases (Mach numbers up to and including $M = 0.6$) were simulated with the pressure-based (coupled) solver, and all the higher Mach number cases were run with the density-based approach. Later, a selected few low-Mach cases were also computed with the density-based solver to verify the consistency of the results.

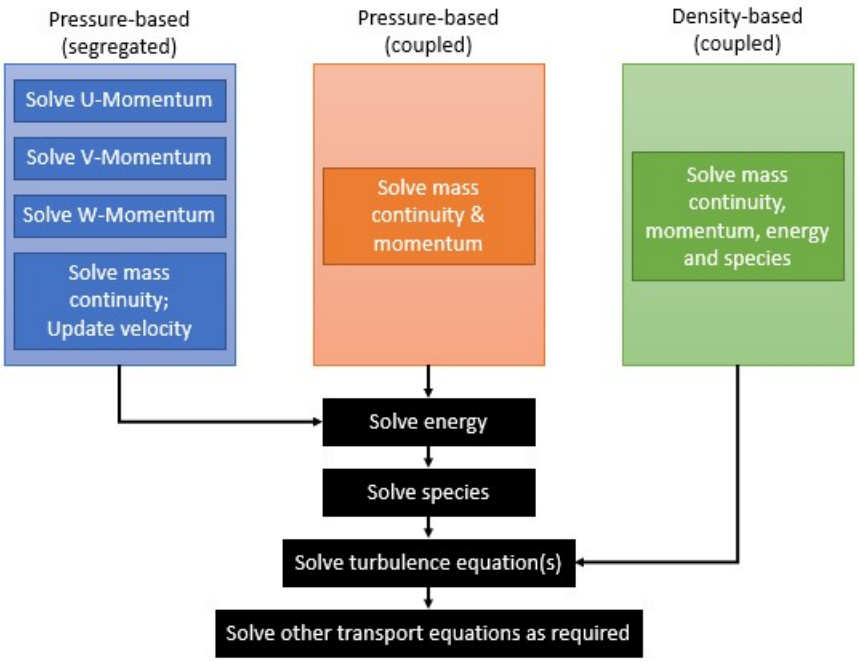


Figure 5.28: Different types of solvers available in Ansys Fluent and their solving sequences.

- **Solver *time*:**
  All the cases can be run as either steady-state or transient simulations. The steady-state simulation computes the mean values of the fully developed solution (time invariant). The transient simulation calculates the instantaneous parameters in every time-step for every quantity. In this project, each case was run with the steady-state simulation.
- **Fluid properties:**
  The fluid used for all the simulations was air with the following properties:
  - Density: according to the ideal gas law
  - Specific heat: constant
  - Viscosity: according to the Sutherland law
  - Molecular weight: constant
- **Energy equation:**
  Since the simulations were run using the ideal gas law, the energy equation was added to the governing equations for all the cases.
- **Turbulence model:**
  The selection of the turbulence model was limited to the ones already introduced in **Paragraph 3.2.3.3.1**. In the end, the one-equation Spalart-Allmaras model was used for various reasons. It uses only one equation for turbulence modeling, thus the computational overhead is lower in comparison with the other considered models. The model is regarded as highly robust and relatively accurate. It was extensively validated and is usually able to predict lift and drag with reasonable accuracy for small and moderate angles of attack. The angles of attack to be simulated in this project range from 0° to 25°, which can be considered *small to moderate*, thus the choice of this closure model is reasonable. Its main drawbacks come from its one-equation formulation. Time and length scales of turbulence are not as well defined as they are in other models (e.g. SST), what might lead to problems with solving shear flows, under-prediction of separation and decaying turbulence [115], [116], [117]. Ultimately, the highest level of accuracy is not the main goal of this project, thus bearing in mind that this model is still an industry standard, potential increase in accuracy was traded off for faster and more robust simulations.

- **Boundary conditions:**
  The following boundary conditions were used in all the simulated cases:
  - Since the half-model was used, the symmetry boundary condition was applied at the symmetry XZ-plane
  - The front, top, bottom, and side faces were given the pressure far-field boundary conditions. These are used to model freestream conditions at *infinity*, and allow the user to directly specify the Mach number, ambient pressure, and ambient temperature. Furthermore, the direction of the flow can be explicitly specified by the definition of the flow vector, which makes possible simulations at different angles of attack without changing the mesh for each case. The direction of the flow is directly related to the vehicle's angle of attack.
  - The rear face was given the pressure outlet boundary condition.
  - The surface forming the spaceplane's external shape was given the wall boundary conditions, with no slip, standard roughness model, no heat flux, and no heat generation rate.
- **Solution methods:**
  For the density-based cases, the *implicit* formulation, with Flux Type: *Advection Upstream Splitting Method* was used. The AUSM method is a numerical function that solves a general system of conservation equations. A detailed description of the method is outside the scope of this project, but the reasoning behind the choice of this particular method is that it offers a uniform accuracy and convergence across all Mach numbers (from low-subsonic to hypersonic). The spatial discretization for the gradient was selected as *Least Squares Cell Based* method, and *Second Order Upwind* for both flow and modified turbulent viscosity. Lastly, high order term relaxation was enabled with the relaxation factor of 0.25.

  For the pressure-based solver, the following solution methods were applied. Spatial discretization was selected as *Least Squares Cell Based* for gradient, *Second Order* for pressure, and *Second Order Upwind* for density, momentum, energy and modified turbulent viscosity. The same high order term relaxation setting were set, and a pseudo transient method was enabled (it might help in reaching faster convergence in steady-state simulations).
- **Solution controls:**
  The main solution control parameter in the density-based solver is the Courant number, which directly influences the time step (Δt) of the simulation. It is calculated through Equation 5.3, and depending on the formulation, CFL_{max} gets less or more strict. Δx in this equation is the size of the smallest cell. For explicit formulation, CFL_{max} should not exceed 1, while for implicit formulation it can be more relaxed. During the entire campaign this number was adjusted to the simulated flight conditions accordingly.

$$CFL = \frac{u\Delta t}{\Delta x} \leq CFL_{max} \tag{5.3}$$

  For the pressure-based solver the parameters of solution controls were left at their default values.
- **Convergence criteria:**
  Convergence of the simulations was controlled using standard residuals. Additionally aerodynamic coefficients, and mass fluxes were tracked throughout the simulations to ensure well-converged results. In the density-based cases, the general condition put on all residuals was 10^{-3}. Similar convergence criteria were applied for cases simulated with the pressure-based solver. The only difference was in the energy residual, for which the condition was increased to 10^{-6}. These are the suggested values for the respective solver types as described in the User's Guide [118].

5.5. Building the aerodynamic database

Chapter 4 introduced two interpolation methods - kriging and co-kriging. This short section describes how the aerodynamic database is built in the framework employing these two methods.

The code employing the kriging and co-kriging algorithms was created in MATLAB by Forrester, Sóbester, and Keane as a part of their book titled *Engineering Design via Surrogate Modelling* [95]. It is publicly available under the *GNU Lesser General Public License* [119]. Some changes and minor adjustments were introduced into the code to slightly extend its capabilities and make it work better with the created framework. In the *out-of-the-box* version of the provided code the p parameters were invariable, and kept constant, at $p = 2$ for all the cases. One of the bigger changes that allowed for a notable improvement of the algorithm's accuracy was making the p parameters tunable by the optimisation algorithm. The MATLAB workflow employs a genetic algorithm to find the optimised values of all the θ_l, p_l, and ρ parameters to build the surrogate Mach-alpha surfaces of the coefficients of lift, drag and pitching moment. All the calculations are done automatically when called by the user, as long as the files containing the results of the low-fidelity and high-fidelity simulations are made available to the framework.

6

Results

The results presented and discussed in this chapter serve as the validation case of the created framework. **Section 6.1** introduces the strategy and rationale of the validation process. Then, **Section 6.2**, shows the mesh independence study for both low-fidelity panel method calculations, and high-fidelity CFD. **Section 6.3** compares the high-fidelity CFD results to the aerodynamic coefficients of *the X-34* found in the literature, coming from wind-tunnel testing campaigns. This can be considered a *sanity check* validation of these results. Following is the comparison of *ZONAIR* with *Ansys Fluent* results in **Section 6.4**. Lastly, **Section 6.5** discusses the results of the building of the aerodynamic databases using kriging and co-kriging algorithms.

6.1. Methodology

Since the discrete grid, be it a surface mesh or a volume mesh, only approximates the actual geometry, it is important to only use simulation results that are not dependent on this discretization. Therefore, before any validation can be done, the mesh independence study of both the panel method and the CFD analysis, needs to be conducted. Only then, the validation of the framework can be carried out.

In order to analyse the created framework it was necessary to establish a validation case. For this, *the X-34*'s trajectories available in the literature were studied ([30], [31]) to select the flight conditions expected during a spaceplane's mission. Three such trajectories are presented in Figure 6.2. The data points generated with the low- and high-fidelity tools were selected based on the information found in these figures. Then, the range of Mach numbers and angles of attack was chosen. The atmospheric flight conditions were selected by approximating the altitude at which a given Mach number could be expected. The data points do not match the exact points along the trajectory but rather reflect the combinations of velocities, angles of attack and altitudes that a spaceplane could experience. These points were chosen arbitrarily and no sampling algorithm was employed for this task. The idea was to spread the points evenly in the selected ranges of Mach numbers and angles of attack, with a more densely populated transonic region due to high expected gradients of the coefficients of interest. Because of a limited time that could be dedicated to this step of the project, the range of Mach numbers was set to between $M = 0.2$ and $M = 4$, and the angles of attack between $\alpha = 0^o$ and $\alpha = 25^o$. For *ZONAIR*, the population of simulated points is very dense, with the step for Mach number of 0.05, and angle of attack of 1°, excluding Mach numbers of $M = 0.95$ and $M = 1$ provided that the results produced by *ZONAIR* at these Mach numbers were extremely inaccurate, and were detrimental to the interpolation. The simulated data points are summarised in Table 6.1 and Figure 6.1. The altitude given in Table 6.1 was used to calculate all the necessary prop-

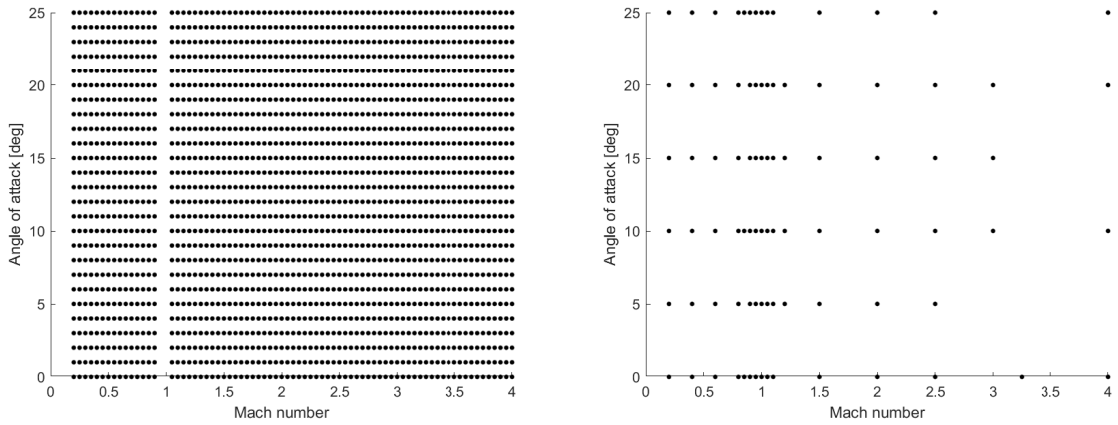
erties based on the International Standard Atmosphere (ISA). In the present study, basically only half of the *speed trajectory* is analysed. This approach is fully justified having in mind the following reasons:

- the main goal of this study is not to create a surrogate model, but to validate the proposed approach by applying these interpolation algorithms to building an aerodynamic database; this task is still fully completed even on a limited Mach range
- above the Mach number of $M = 4$, when the spaceplane starts to enter the hypersonic speed regime, the changes in the aerodynamic coefficients are very small (relatively to the slower speed regimes), thus if the approach can be validated in this selected region, it will also work if the Mach range is extended

Table 6.1: Overview of the 87 data points simulated with Ansys Fluent.

Mach number	Angles of attack	Altitude	Reynolds number*
0.20	0°, 5°, 10°, 15°, 20°, 25°	*SL*	20.3E6
0.40	0°, 5°, 10°, 15°, 20°, 25°	*SL*	40.7E6
0.60	0°, 5°, 10°, 15°, 20°, 25°	*SL*	61.1E6
0.80	0°, 5°, 10°, 15°, 20°, 25°	*SL*	81.5E6
0.85	0°, 10°, 25°	10km	31.6E6
0.90	0°, 5°, 10°, 15°, 20°, 25°	10km	33.5E6
0.95	0°, 5°, 10°, 15°, 20°, 25°	10km	35.4E6
1.00	0°, 5°, 10°, 15°, 20°, 25°	10km	37.2E6
1.05	0°, 5°, 10°, 15°, 20°, 25°	10km	39.1E6
1.10	0°, 5°, 10°, 15°, 20°, 25°	10km	41.0E6
1.20	5°, 10°, 15°, 20°	10km	44.7E6
1.50	0°, 5°, 10°, 15°, 20°, 25°	10km	55.9E6
2.00	0°, 5°, 10°, 15°, 20°, 25°	12km	56.7E6
2.50	0°, 5°, 10°, 15°, 20°, 25°	15km	44.2E6
3.00	10°, 15°, 20°	15km	53.0E6
3.25	0°	25km	11.6E6
4.00	0°, 10°, 20°, 25°	35km	29.1E5

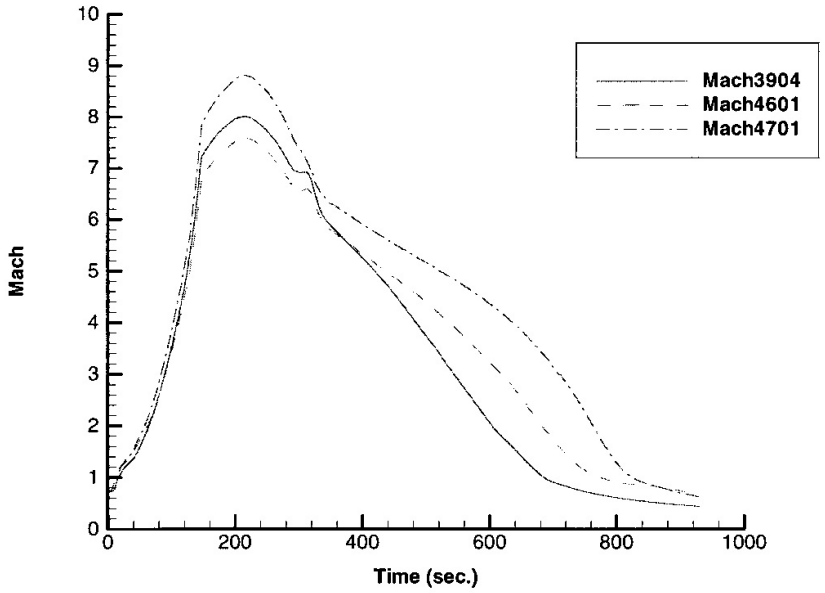
* Reynolds number referred to the mean aerodynamic chord $MAC = 4.432m$



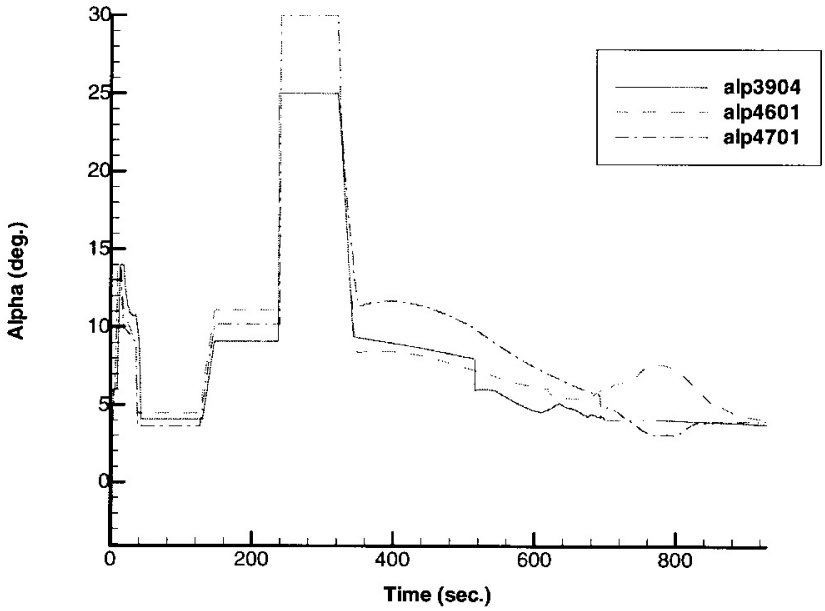
(a) ZONAIR data points.

(b) Ansys Fluent data points.

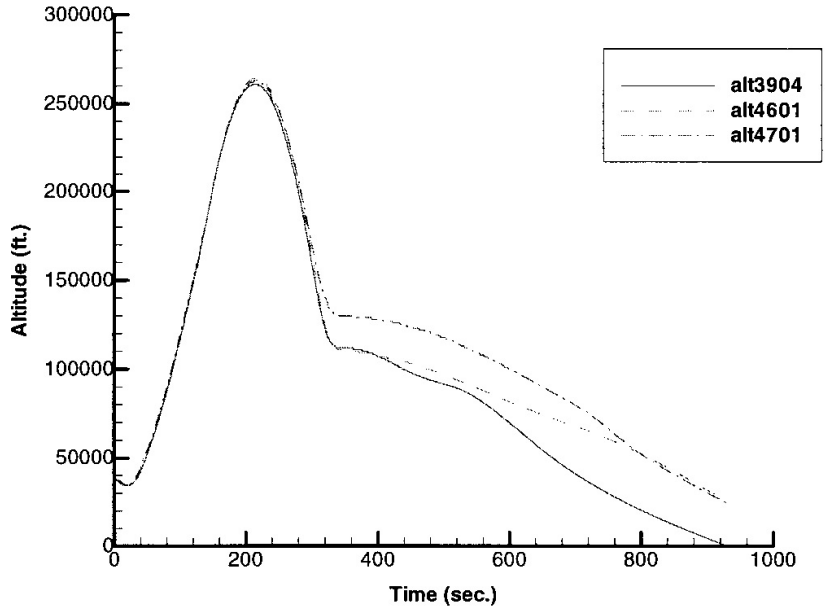
Figure 6.1: Discrete data points simulated in ZONAIR and Ansys Fluent.



(a) Mach time history.



(b) Alpha time history.



(c) Altitude time history.

Figure 6.2: Mach number, angle of attack and altitude time histories of three trajectories of the X-34. [30]

Once all the simulations are finished, the study of the algorithm is conducted. This study is performed in a relatively straightforward way. The aim of this analysis is to simultaneously validate the accuracy of the interpolation, and find out how the accuracy changes as the number of expensive data points and their distribution change. For this purpose, the analysis is done outside of the created framework. A simple program was created to automate the analysis process, which allows to choose the number of expensive data points used for co-kriging. The four points delimiting the analysed Mach-alpha range ($M = 0.2$ at $\alpha = 0°$ and $\alpha = 25°$, $M = 4$ at $\alpha = 0°$ and $\alpha = 25°$) are always used for the process. It ensures that the method is used for interpolation only, and there is no extrapolation. The other points are selected fully randomly to prevent any potential bias. The remaining points are used to compute the accuracy of the created models. There are multiple such runs realized for each number of points, and each time the points are selected randomly. For each run, the relative error at each validation point, the mean absolute error (MAE) and the error's standard deviation (STD) are calculated in the three speed ranges - subsonic (below $M = 0.8$), transonic ($M = 0.8$ to $M = 1.2$) and supersonic (above $M = 1.2$). Then, the mean absolute error and the standard deviation are averaged over all the realized runs and compared to the MAEs and STDs of kriging models. Since the number of expensive data points is limited, the division of thereof into the *interpolation group* and *verification group* can have high influence on the resulting errors. Especially, since a higher number of points used for interpolation also means a higher uniformity of distribution of these points across the entire domain. The reason for this interrelation is that the expensive data points had already been generated. The random selection algorithm chooses the points from the already existing, discrete database. Importantly, if fewer points are selected, there is a higher chance that they would all be located in close proximity of each other. For example, if only six points are selected, the chance that all will be in the subsonic region, or that they would all have the same angle of attack, etc., is higher than for the same to happen with a case with twenty six points. For this reason, in the presented study, it is impossible to fully separate the uniformity of the distribution from the number of points used for interpolation. Hence, the number of performed runs should be high enough to thoroughly study the resulting surrogate models. In this document, for each number of points, thirty runs are presented. However, during the realization of the project many more such runs were completed to ensure high quality of the presented results. The ones included in this document are the most representative of the observed trends.

6.2. Mesh independence study

In order to accept the simulation results of the employed methods, the mesh independence study needs to be carried out. For this purpose, various meshes were created with an increasing number of elements. Each mesh with smaller cells in the most critical areas of the geometry, where the biggest gradients of the flow characteristics are expected, e.g., the boundary layer region, regions of high curvature, wake behind the vehicle, etc. For each tool, two cases of the mesh independence study were conducted - one in the subsonic speed regime, and one in the supersonic regime. The study consisted of the analysis of the change in the longitudinal aerodynamic coefficients with the subsequent mesh refinement steps.

6.2.1. ZONAIR

In this part of the study, five meshes were analysed for both the subsonic (M=0.45), and the supersonic (M=1.55) case. Simulations at both speed regimes were conducted at the angle of attack of 0°. The refinement of the surface mesh was concentrated on:

- better reflecting the geometry in regions of high curvature, such as the wing's and the vertical tail's leading edges, nose of the fuselage, and fuselage's rounded corners
- refining the regions such as trailing edges, or wingtips, where relatively higher gradients of the flow properties should be expected

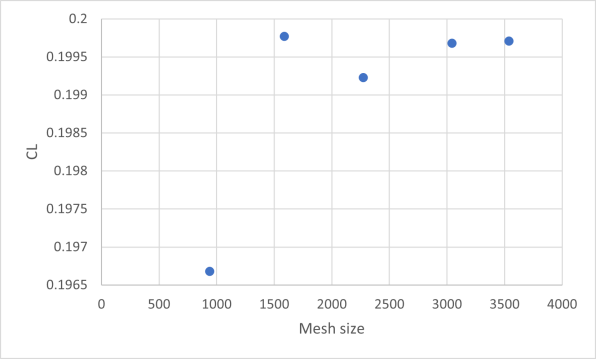
The results of this study can be seen in Tables 6.2 and 6.3, and Figures 6.3 and 6.4. As can be observed in the presented results, for both the subsonic and the supersonic cases, the mesh-convergence is achieved with the third mesh consisting of 2,273 elements. But even the second, not as dense mesh of 1,589 elements, does provide results that are very close. Only for the coarsest mesh, the values of the coefficients differ significantly. Since the increase in the computational time required for the simulations when going from 1,589 to 2,273 elements is not too high, it was decided to perform all the low-fidelity calculations with Mesh#3.

Table 6.2: Longitudinal aerodynamic coefficients at M=0.45 - subsonic grid independence study of ZONAIR.

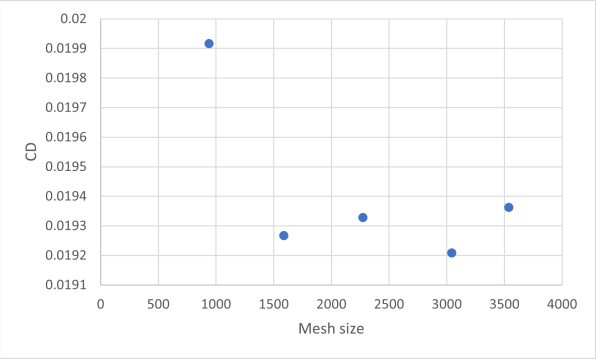
Number of elements	C_L	C_D	C_M
Mesh#1: 941	0.19668	0.019916	-0.13768
Mesh#2: 1,589	0.19977	0.019267	-0.13750
Mesh#3: 2,273	0.19923	0.019328	-0.13751
Mesh#4: 3,043	0.19968	0.019208	-0.13746
Mesh#5: 3,537	0.19971	0.019363	-0.13747

Table 6.3: Longitudinal aerodynamic coefficients at M=1.55 - supersonic grid independence study of ZONAIR.

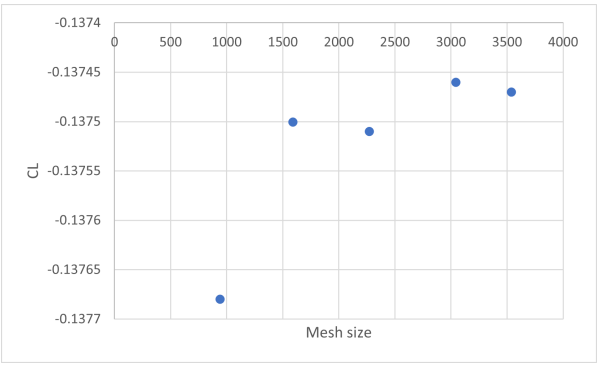
Number of elements	C_L	C_D	C_M
Mesh#1: 941	0.13273	0.090324	-0.15907
Mesh#2: 1,589	0.11913	0.075345	-0.16121
Mesh#3: 2,273	0.11460	0.074062	-0.16135
Mesh#4: 3,043	0.11505	0.072837	-0.16161
Mesh#5: 3,537	0.11393	0.073534	-0.16121



(a) Lift coefficient

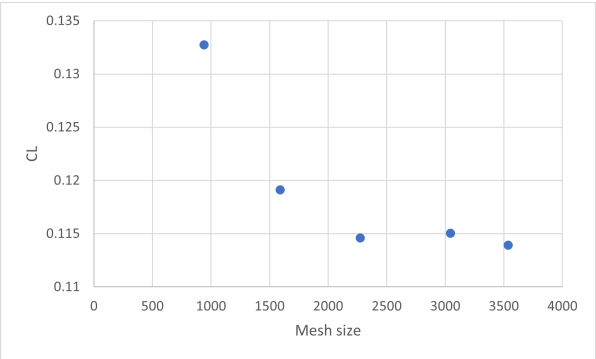


(b) Drag coefficient

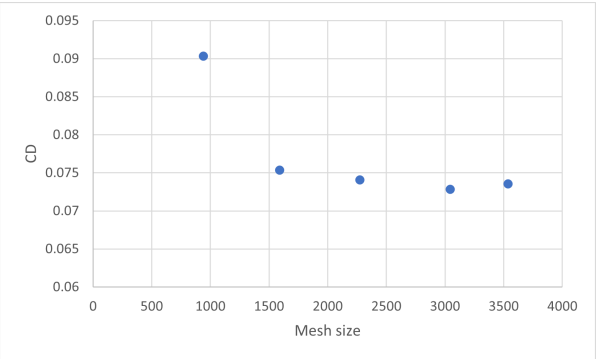


(c) Pitching moment coefficient

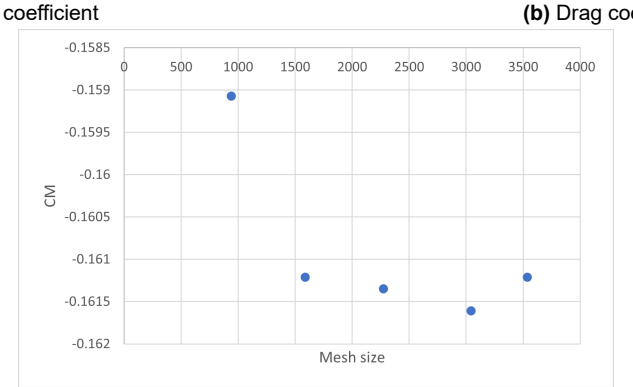
Figure 6.3: Longitudinal aerodynamic coefficients at M=0.45 - subsonic grid independence study of ZONAIR.



(a) Lift coefficient



(b) Drag coefficient



(c) Pitching moment coefficient

Figure 6.4: Longitudinal aerodynamic coefficients at M=1.55 - supersonic grid independence study of ZONAIR.

6.2.2. Ansys Fluent

Similarly to the case of *ZONAIR*, for *Ansys Fluent* again five different meshes were created for the purpose of the mesh independence study. The same principle was applied, and each subsequent mesh had an increased number of cells through refinement of the regions with the highest gradients of the flow characteristics. At the end of the study, one baseline mesh is selected, that would later be refined locally depending on the simulated cases. An example of a phenomenon that requires separate local refinement is a shockwave. Once the first simulation arrives at a solution, the results of this simulation are used to refine the regions where the shocks are predicted. This is to improve convergence, and the quality of the final results. This *baseline* mesh independence study is carried out for one subsonic case (M=0.4, AoA=0°), and one supersonic case (M=1.5, AoA=0°).

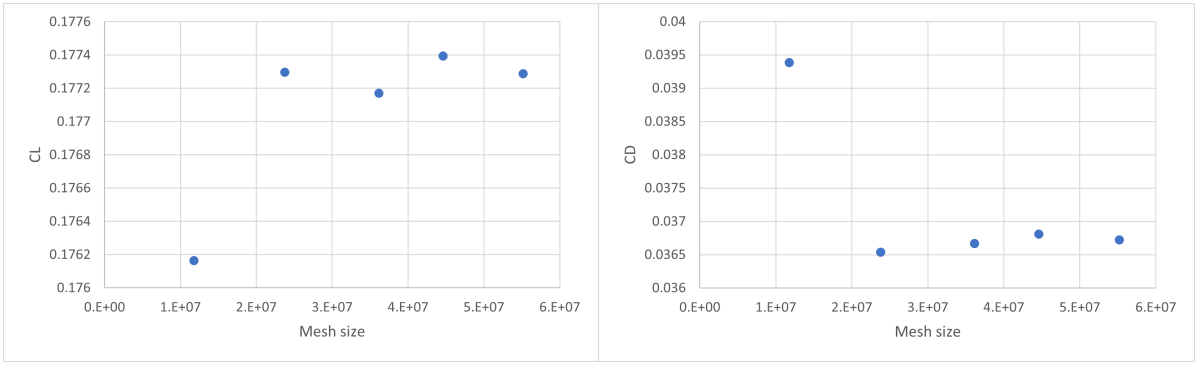
The outcome of the analysis is presented in Tables 6.4 and 6.5, and Figures 6.5 and 6.6. The obtained values of the longitudinal aerodynamic coefficients show that the convergence is already reached with Mesh#2. Therefore, for the rest of the simulations planned for this project, this one was chosen as the baseline mesh.

Table 6.4: Longitudinal aerodynamic coefficients at M=0.4 - subsonic grid independence study of Ansys Fluent.

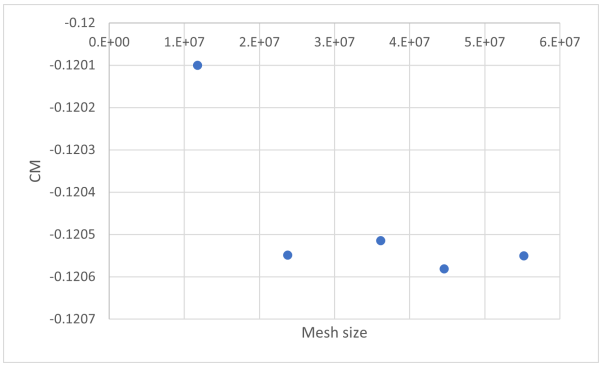
Number of elements	C_L	C_D	C_M
Mesh#1: 11,772,892	0.17617	0.039385	-0.12010
Mesh#2: 23,770,388	0.17729	0.036539	-0.12054
Mesh#3: 36,140,408	0.17717	0.036670	-0.12051
Mesh#4: 44,594,017	0.17739	0.036808	-0.12058
Mesh#5: 55,182,903	0.17728	0.036720	-0.12055

Table 6.5: Longitudinal aerodynamic coefficients at M=1.5 - supersonic grid independence study of Ansys Fluent.

Number of elements	C_L	C_D	C_M
Mesh#1: 11,772,892	0.070074	0.085217	-0.10228
Mesh#2: 23,770,388	0.071554	0.087241	-0.10965
Mesh#3: 36,140,408	0.071654	0.087154	-0.11071
Mesh#4: 44,594,017	0.072015	0.087532	-0.10962
Mesh#5: 55,182,903	0.071761	0.087621	-0.11018

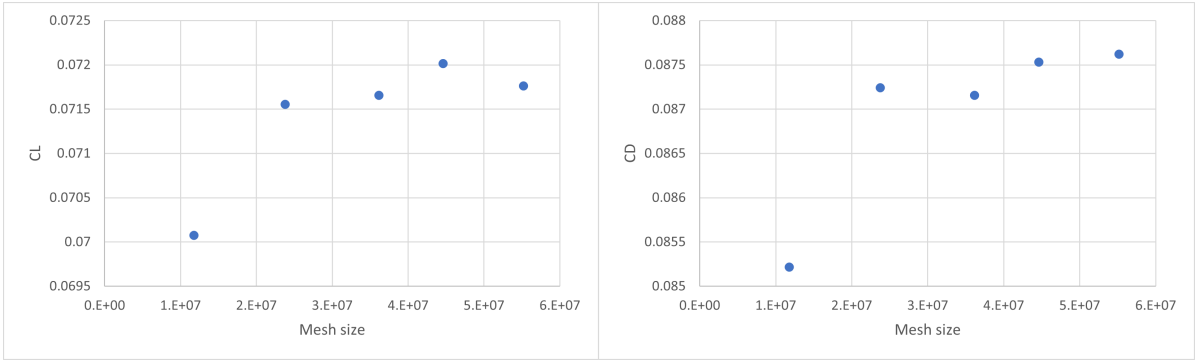


(a) Lift coefficient **(b)** Drag coefficient

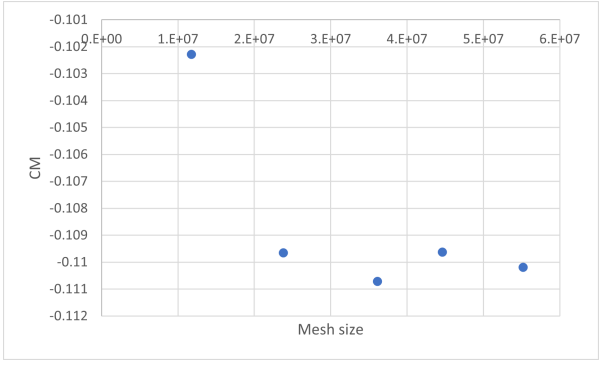


(c) Pitching moment coefficient

Figure 6.5: Longitudinal aerodynamic coefficients at M=0.4 - subsonic grid independence study of Ansys Fluent.



(a) Lift coefficient **(b)** Drag coefficient



(c) Pitching moment coefficient

Figure 6.6: Longitudinal aerodynamic coefficients at M=1.5 - supersonic grid independence study of ZONAIR.

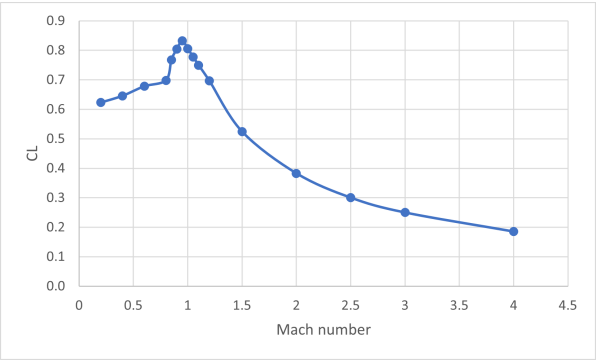
6.3. Ansys Fluent results

Although, given the layout of this report up to this point, it would be more logical to start with the analysis of low-fidelity results, in this case, it was decided to switch the order. The reason is that first the high-fidelity simulations are validated against wind-tunnel data, and only then *ZONAIR* results are compared to the CFD results in **Section 6.4**. Thus, in this section, the results of the simulations realized with the high-fidelity tool are presented. The goal is to verify the correctness of the obtained data, and identify potential problems. The main steps to be taken in this part of the study are:

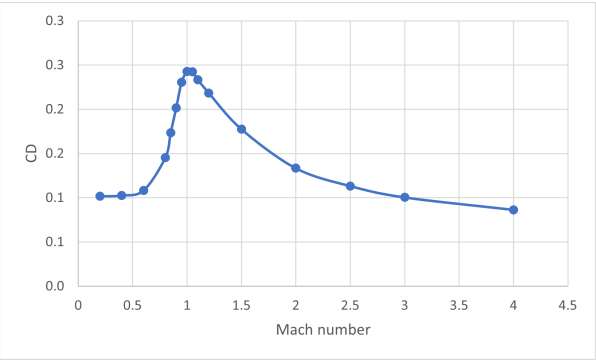
- analyse variation of the aerodynamic coefficients with increasing Mach number, and check if they agree with theory
- compare obtained CFD results with the wind-tunnel data of the reference vehicle, *the X-34* [30], [120]

The exact numerical values of the coefficients of interest at each simulated data point can be found in Appendix A.

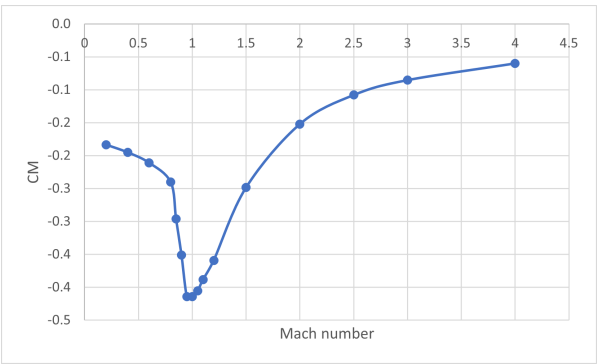
Figure 6.7 shows the variation of the longitudinal aerodynamic coefficients with Mach number as calculated with *Ansys Fluent*. The theoretical variation of lift and drag coefficients is shown in Figure 6.8. As can be observed, the trends of both force coefficients agree very well with theory. They both grow slowly in the subsonic region until they *hit* a more steep rise in the transonic regime. They reach their respective maxima around M=1, and the values of the coefficients begin to drop again as the Mach number increases into the supersonic speeds. The pitching moment coefficient also follows what should be expected from theory. As the Mach number increases, the center of pressure moves rearward, increasing the nose-down pitching moment, and as the speed is increased beyond the sonic speed, the pitching moment should decrease again. Figure 6.9 shows how the neutral point of a supersonic aircraft, YF-12A, travels with increasing Mach number. The variation in the location of the neutral point is highly correlated with the change in pitching moment. Therefore, the trends obtained from the CFD simulations of all the analysed coefficients do agree qualitatively with theory.



(a) Lift coefficient

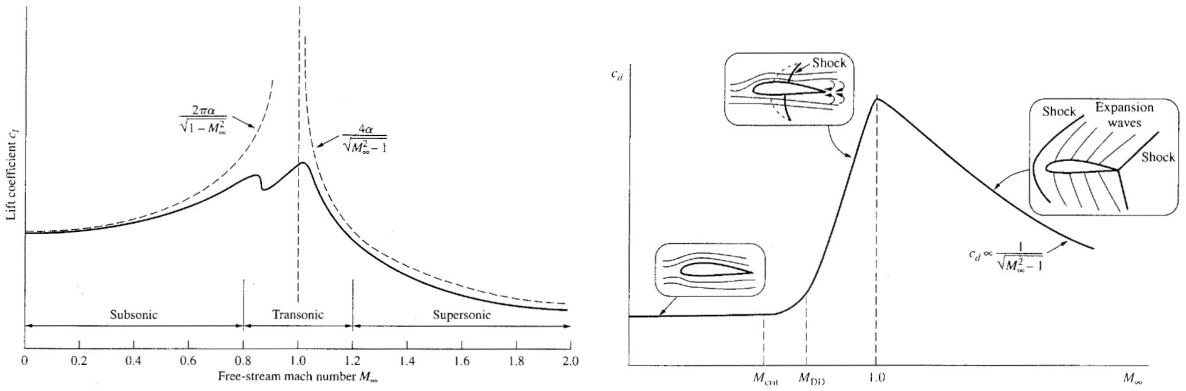


(b) Drag coefficient



(c) Pitching moment coefficient

Figure 6.7: Aerodynamic coefficients from Ansys Fluent simulations as a function of Mach number, at AoA=10°.



(a) Lift coefficient **(b)** Drag coefficient

Figure 6.8: Generic variation of lift and drag coefficients with Mach number. [41]

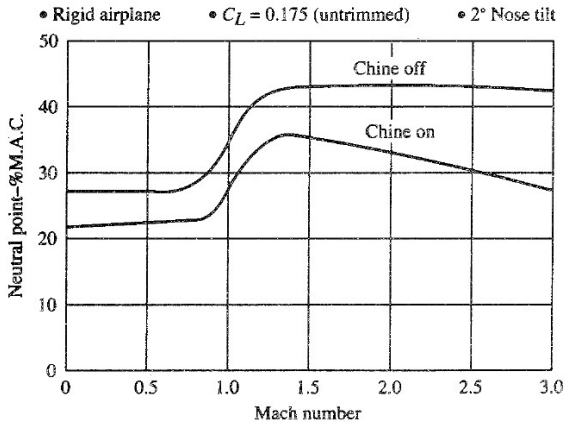
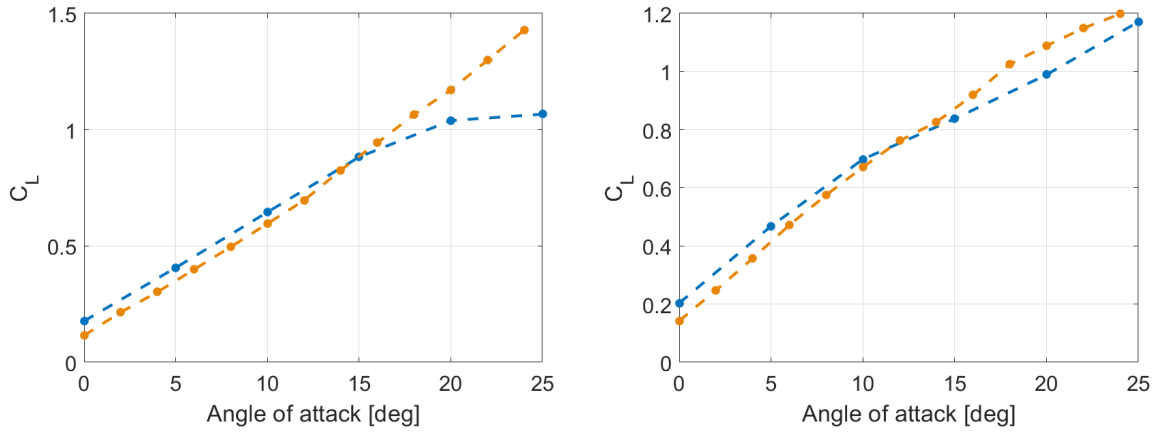


Figure 6.9: Variation of the location of the neutral point with Mach number of YF-12A. [41]

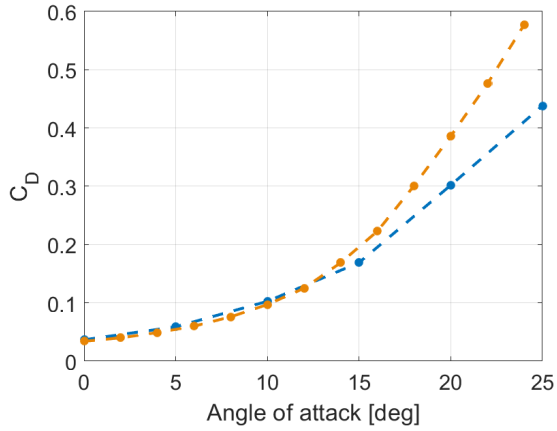
Next, the CFD data is compared to the wind-tunnel results of *the X-34*. Even though there are some differences between the two geometries, and some uncertainties that cannot be quantified, overall there should be a relatively good agreement. However, because of the above-mentioned reasons, and because of the unquantifiable error during the digitization of the wind-tunnel data, there is no point in quantifying the difference between the two data sets. Thus, only a visual, qualitative comparison is offered.

Figure 6.10 presents the comparison of the two data sets in the subsonic speed regime. As can be seen, in the low-angle-of-attack region, the agreement between the two sets is excellent. Only above the angle of attack of 15° the difference starts to be more notable. Apart from the above-mentioned geometrical differences, the possible source of this discrepancy can be explained by the nature of the employed turbulence model. Possibly, it is unable to accurately resolve the wake behind the vehicle as the angle of attack increases. Furthermore, there is a big disagreement between the CFD and the WT data for pitching moment coefficient. While there is very little change in thereof for the WT data, the CFD predicts a stable increase in the nose-down pitching moment. The most probable reasons for this discrepancy would be the different airfoil shape that could not be accurately modeled, different cross-section of the rear part of the fuselage, and the lack of the body-flap. Pitching moment is the most sensitive to this kind of changes in geometry since it depends on the exact pressure distribution on the vehicle, which might explain a good agreement of the force coefficients, with a big difference in pitching moment coefficient.

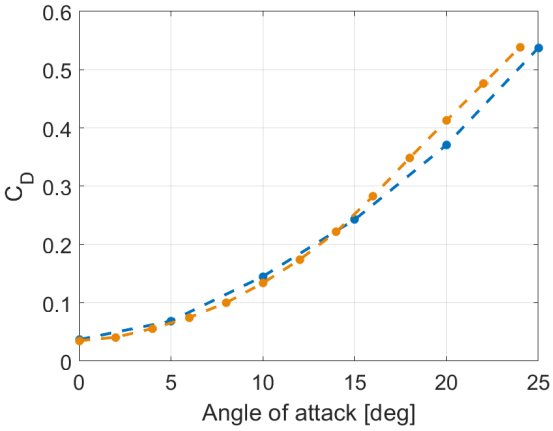


(a) Lift coefficient, M=0.4

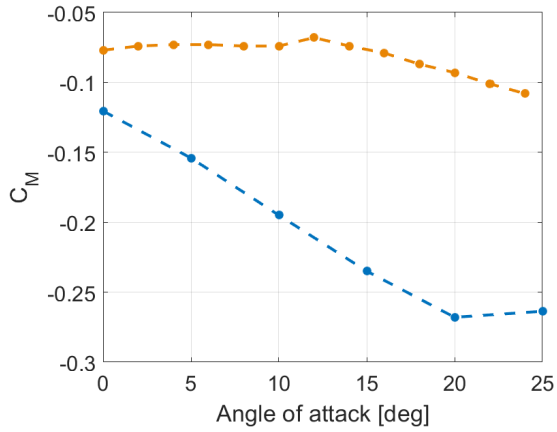
(b) Lift coefficient, M=0.8



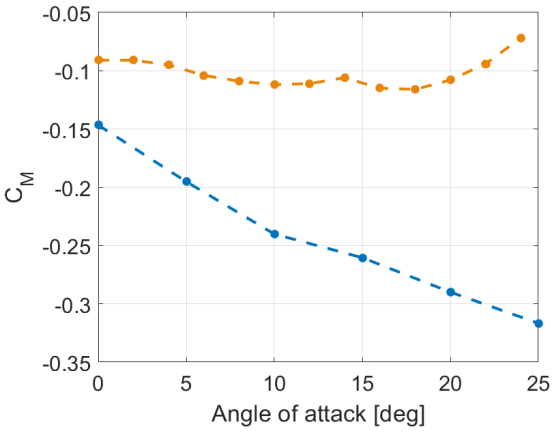
(c) Drag coefficient, M=0.4



(d) Drag coefficient, M=0.8



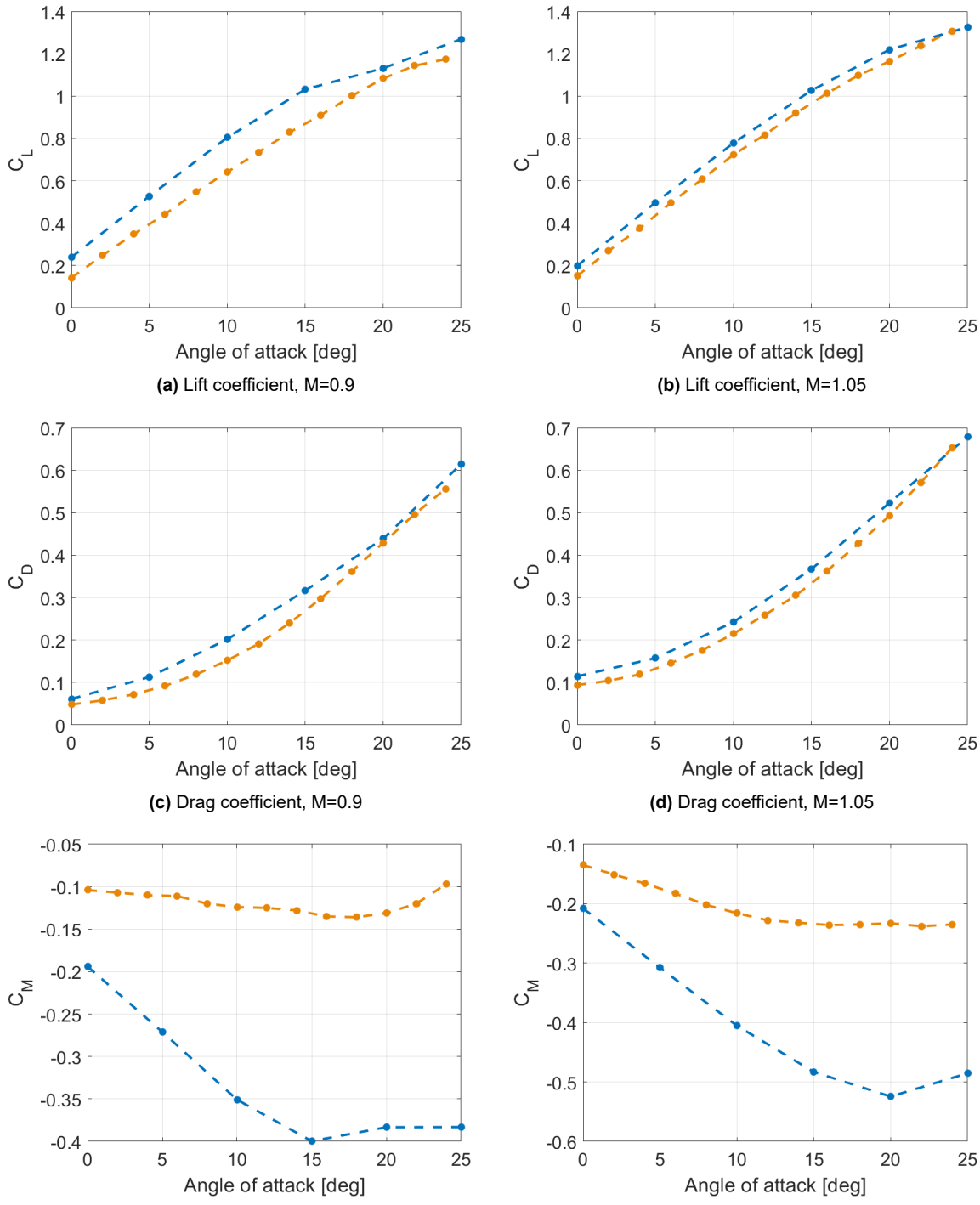
(e) Pitching moment coefficient, M=0.4



(f) Pitching moment coefficient, M=0.8

Figure 6.10: CFD (**blue**) vs the X-34 wind-tunnel data (**orange**) at two subsonic Mach numbers (M=0.4, M=0.8).

Figure 6.11 follows with a similar comparison, this time in the transonic speed regime - one slightly below the sonic speed at M=0.9, and one just above the sonic speed at M=1.05. Similarly to the subsonic case, the agreement for the two force coefficients is satisfactory, especially for M=1.05. This time, even in the moderate-angle-of-attack range, the difference between the CFD and the WT data stays relatively close. When it comes to pitching moment coefficient, basically the same conclusions can be drawn as for the two discussed subsonic Mach numbers.



(a) Lift coefficient, M=0.9

(b) Lift coefficient, M=1.05

(c) Drag coefficient, M=0.9

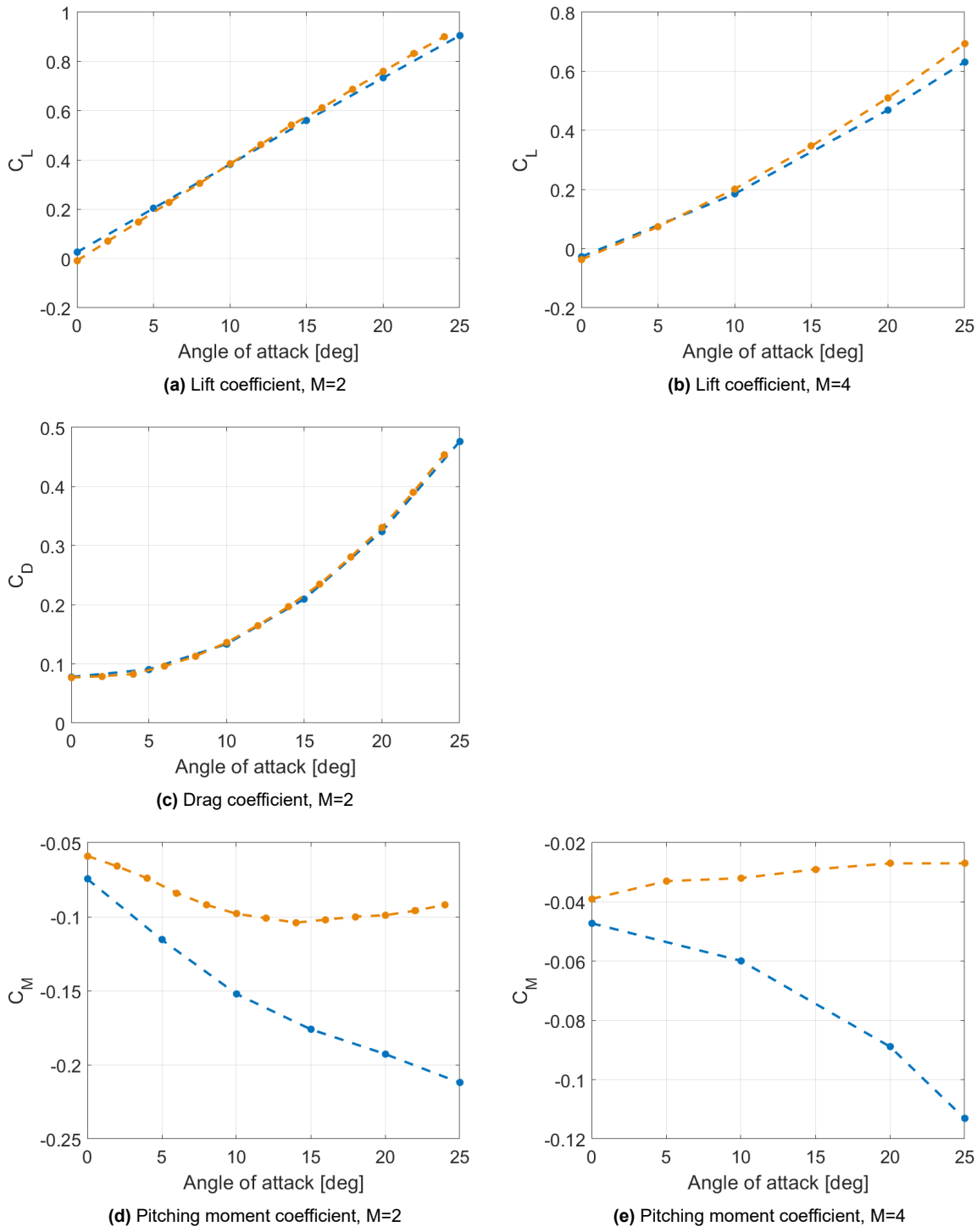
(d) Drag coefficient, M=1.05

(e) Pitching moment coefficient, M=0.9

(f) Pitching moment coefficient, M=1.05

Figure 6.11: CFD (**blue**) vs the X-34 wind-tunnel data (**orange**) at two transonic Mach numbers (M=0.9, M=1.05).

Lastly, Figure 6.12 compares the same coefficients in the supersonic speed range, at M=2 and M=4. Again, the force coefficients agree very well. And in this speed regime, this agreement is visibly the best. Unfortunately, there was no data available for drag coefficient at M=4, thus a comparison could not be made. However, it can still be concluded that the agreement between the two data sets in this regime is acceptable. Nonetheless, yet again, there is a big discrepancy between the pitching moment coefficients.



(a) Lift coefficient, M=2

(b) Lift coefficient, M=4

(c) Drag coefficient, M=2

(d) Pitching moment coefficient, M=2

(e) Pitching moment coefficient, M=4

Figure 6.12: CFD (**blue**) vs the X-34 wind-tunnel data (**orange**) at two supersonic Mach numbers (M=2, M=4).

The big discrepancy in the pitching moment coefficient casts a shadow on the obtained results. Nevertheless, the main goal of this project is not to create the perfect surrogate model of this one specific geometry, but rather to build a workflow capable of creating accurate surrogate models of an arbitrary spaceplane configuration. If some unidentified errors do appear in the CFD simulations, it still does not influence the main goal of the study. Secondly, pitching moment coefficient can be very sensitive to the geometry differences such as the ones between the wind-tunnel-tested *X-34* model and the configuration created using the discussed framework. Lastly, there is nothing *nonphysical* in the data obtained through the CFD simulations. Hence, the data obtained with *Ansys Fluent* is satisfactory for the project's goals.

A CFD analysis is about much more than just obtaining raw coefficient values, it allows for the simulation of the entire flow field around the vehicle. Thus, during the study of the CFD results, the flow fields produced in all the simulations were analysed to make sure there were no unexpected occurrences that would invalidate the final results, and that the obtained values match the generated flow fields. Some of the resulting fields, in terms of pressure coefficient or Mach number, can be seen in Figures 6.13-6.15.

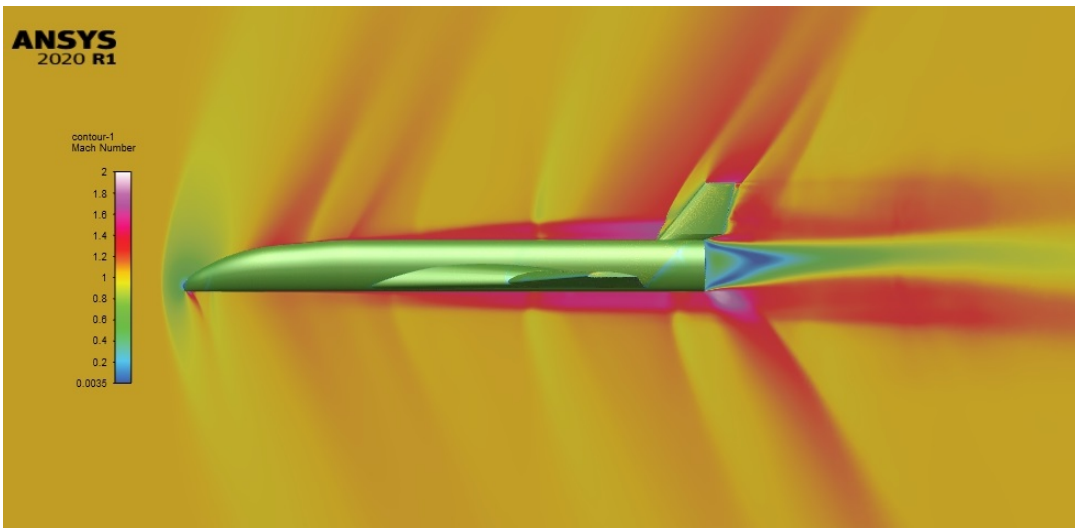


Figure 6.13: Mach number in the flow field around the vehicle at M=1.1, AoA=0°

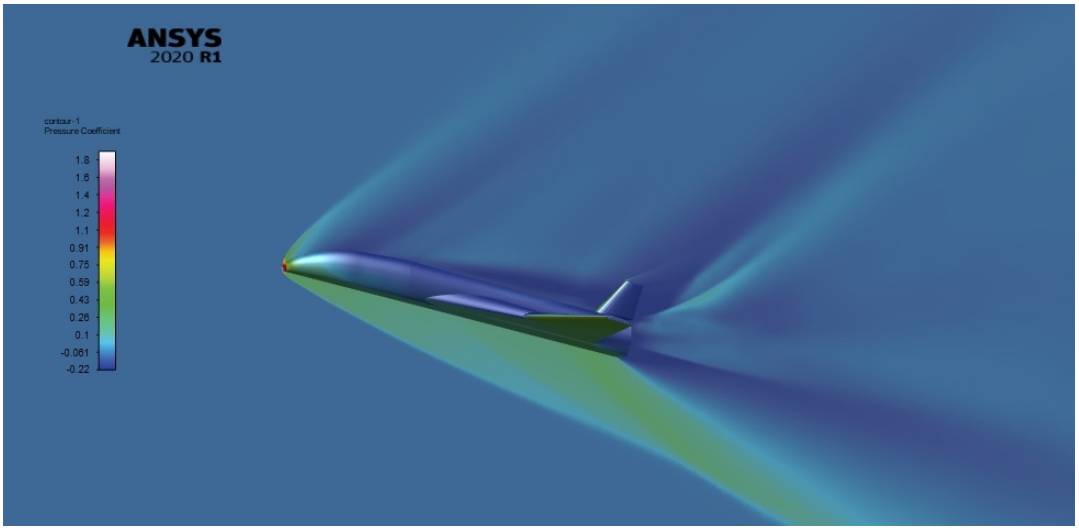


Figure 6.14: Pressure coefficient in the flow field around the vehicle at M=2.5, AoA=20°

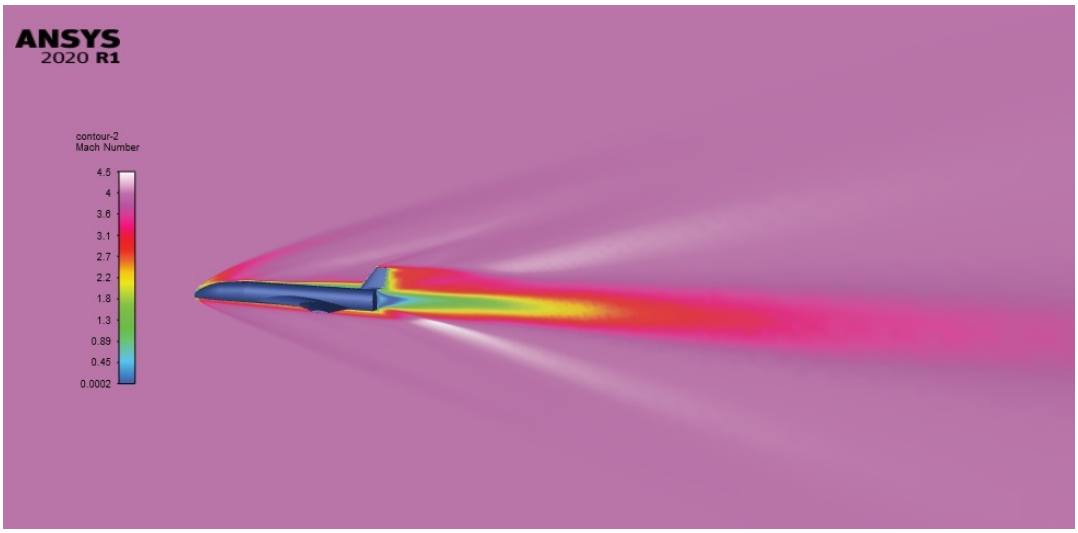
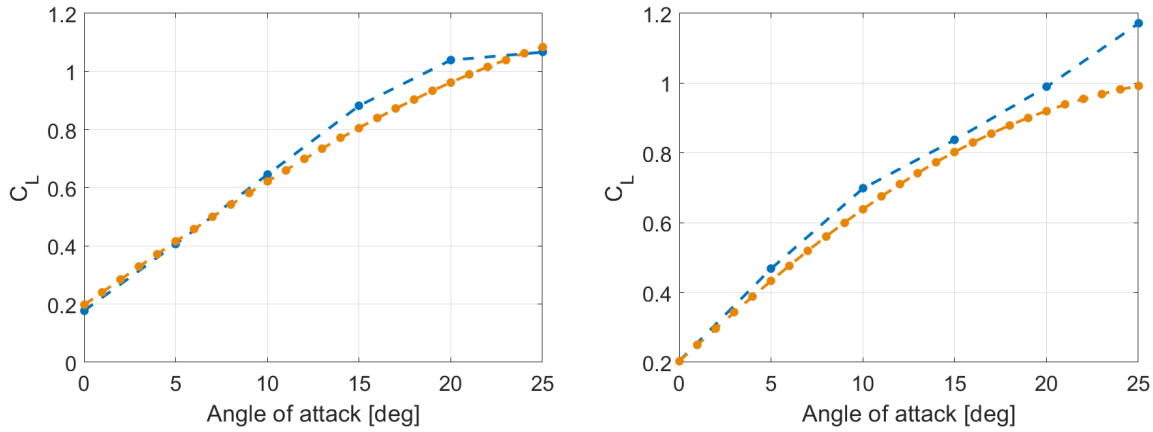


Figure 6.15: Mach number in the flow field around the vehicle at M=4, AoA=0°

6.4. ZONAIR results

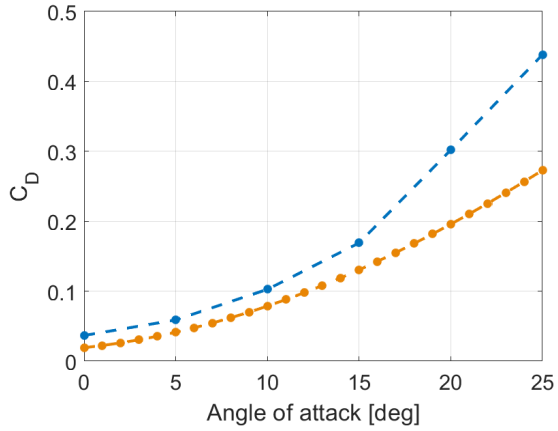
This section is dedicated to the analysis of the results of the low-fidelity calculation tool - *ZONAIR*. A brief comparison with the high-fidelity data is offered at selected Mach numbers in the subsonic, transonic, and supersonic speed regimes, similarly to how it was done in **Section 6.3**. Again, no quantitative study of the presented results is done. The aim of this section is to get an overview of how the data coming from the two tools chosen for this project compares in specific speed regimes. The quantitative analysis of the results is the subject of **Section 6.5**, where the interpolation results are discussed in detail.

Figures 6.16, 6.17 and 6.18 show the variations in longitudinal aerodynamic coefficients with increasing angle of attack at respectively subsonic (M=0.4 and M=0.8), transonic (M=0.9 and M=1.05) and supersonic (M=2 and M=4) speeds. The analysed Mach numbers were chosen specifically so they coincide with the speeds analysed in the wind-tunnel experiments on *the X-34*. Thus, a comparison can be made between *ZONAIR* and WT results just as well. In most cases the agreement between the two applied simulation tools is relatively good. Particularly in the case of the supersonic speed regime. The biggest discrepancies can be observed in pitching moment coefficient. It indicates that while the forces exerted over the vehicle are computed fairly accurately, the actual distribution over the full configuration does not match. Furthermore, it is important to note that drag coefficient is mostly underestimated. It should not be surprising given the nature of the two solvers. As discussed in **Section 5.3.2.1**, the skin friction drag is not computed directly, but only a correction is applied on the results to account for the viscous effects. As expected, a better agreement can be observed in the cases above sonic speed, where wave drag starts to be the major drag component. Thus, the results reflect what should be expected given the limitations of the lower-fidelity software.

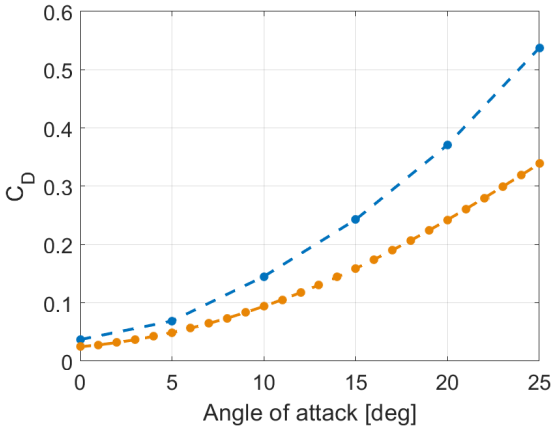


(a) Lift coefficient, M=0.4

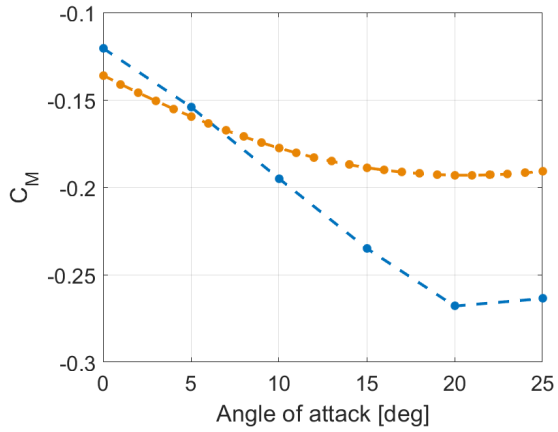
(b) Lift coefficient, M=0.8



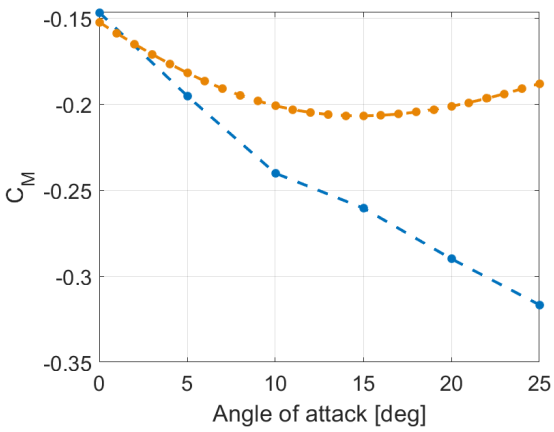
(c) Drag coefficient, M=0.4



(d) Drag coefficient, M=0.8

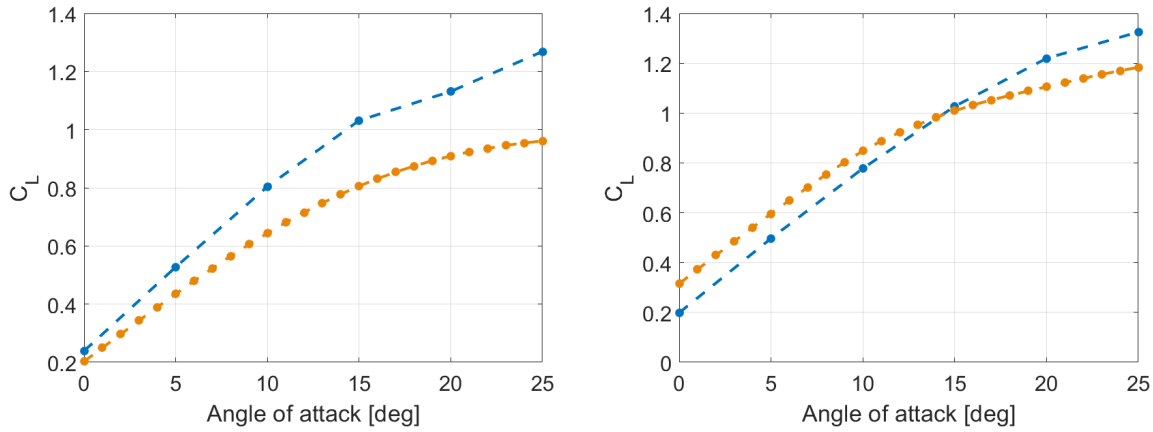


(e) Pitching moment coefficient, M=0.4



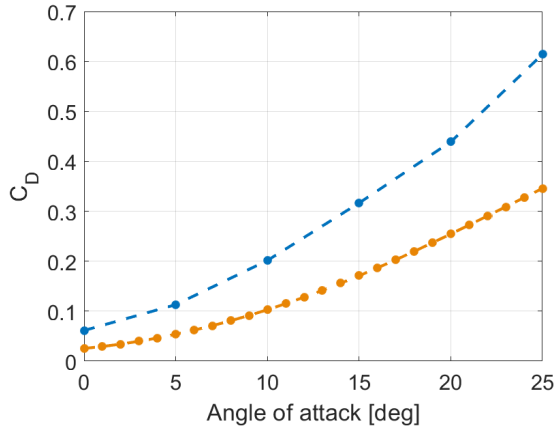
(f) Pitching moment coefficient, M=0.8

Figure 6.16: *Ansys Fluent* (**blue**) vs *ZONAIR* data (**orange**) at two subsonic Mach numbers (M=0.4, M=0.8).

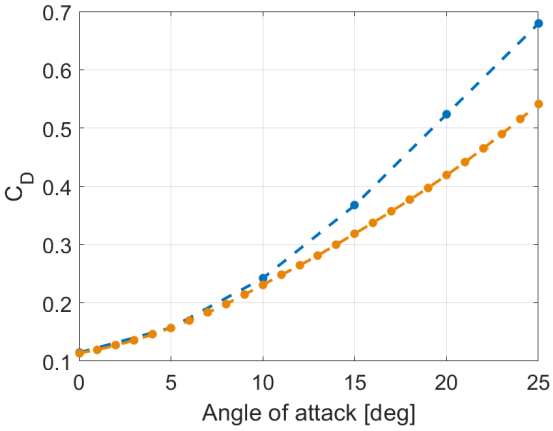


(a) Lift coefficient, M=0.9

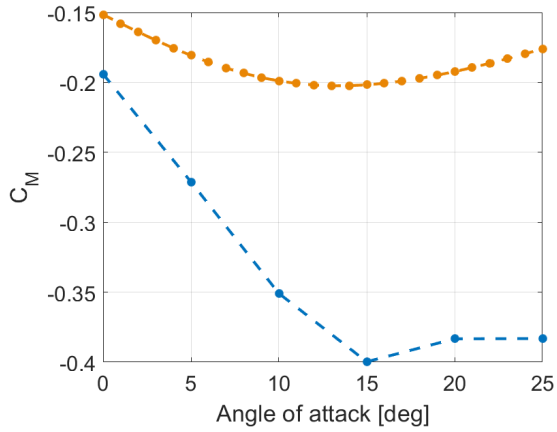
(b) Lift coefficient, M=1.05



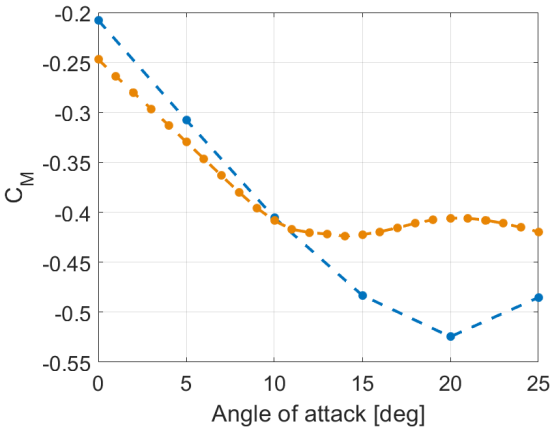
(c) Drag coefficient, M=0.9



(d) Drag coefficient, M=1.05

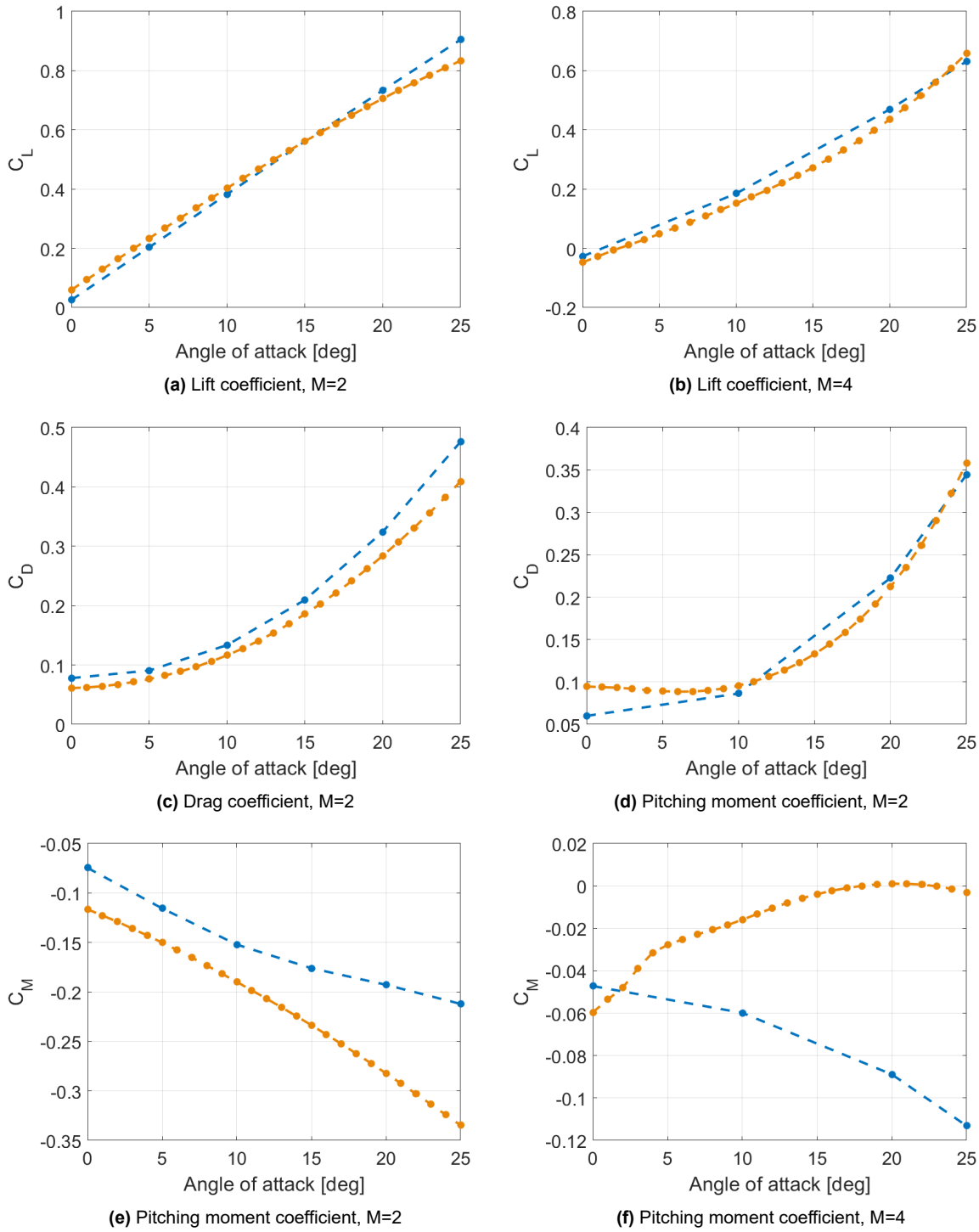


(e) Pitching moment coefficient, M=0.9



(f) Pitching moment coefficient, M=1.05

Figure 6.17: *Ansys Fluent* (**blue**) vs *ZONAIR* data (**orange**) at two transonic Mach numbers (M=0.9, M=1.05).



(a) Lift coefficient, M=2

(b) Lift coefficient, M=4

(c) Drag coefficient, M=2

(d) Pitching moment coefficient, M=2

(e) Pitching moment coefficient, M=2

(f) Pitching moment coefficient, M=4

Figure 6.18: *Ansys Fluent* (**blue**) vs *ZONAIR* data (**orange**) at two supersonic Mach numbers (M=2, M=4).

6.5. Interpolation results

This section is the centerpiece of this report. Here, the results of the two interpolation schemes are presented and thoroughly analysed. **Section 6.5.1** focuses on the results of pure kriging interpolation scheme, using only the low-fidelity data. Then, **Section 6.5.2** examines the effect of adding the expensive data points on the obtained solution.

6.5.1. Kriging results

To better understand the advantage that comes with multi-fidelity interpolation, it is important to analyse the surrogate models created exclusively with cheap data. This analysis is carried out with the following steps:

1. cheap data is interpolated using kriging algorithm to create the surrogate models for each coefficient of interest;
2. all the expensive data points generated with Ansys Fluent are used as verification cases, to study the accuracy of these models;
3. the mean absolute error is calculated at each point, and then the error is averaged in each of the speed regimes (subsonic, transonic, and supersonic) separately, and its standard deviation is computed.

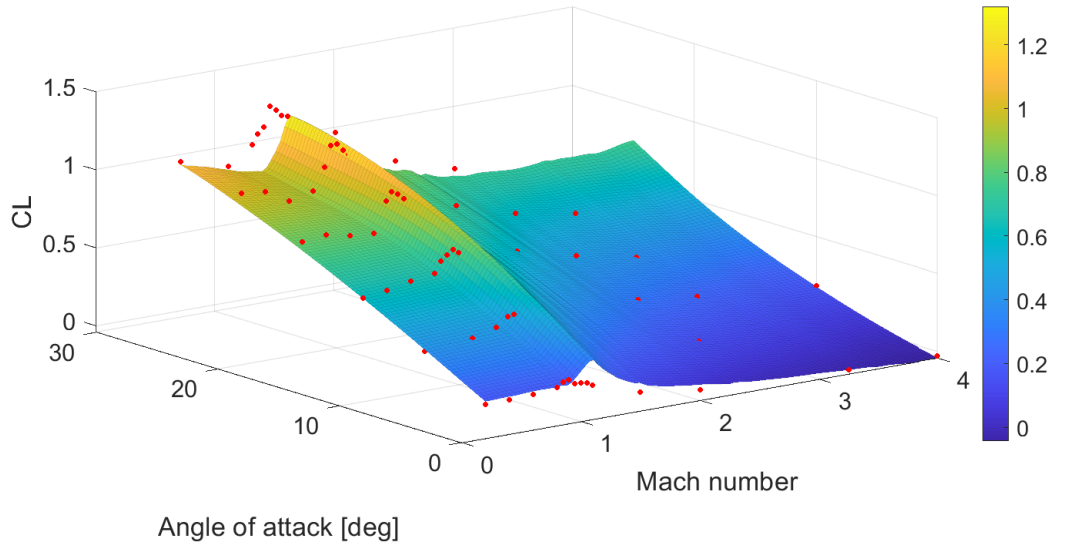
Figure 6.19a shows the surrogate model of lift coefficient resulting from kriging interpolation. The red dots visible in the figure represent the expensive data points. Additionally, Figure B.1, attached to this report in Appendix B shows bar plots of the relative errors at each data point. From there, it can be more clearly seen that generally, the surrogate model of the lift coefficient under-predicts the values for lower angles of attack, and over-predicts for higher angles. Particularly in the subsonic and transonic ranges. In the supersonic range, it mostly under-predicts *independently* of whether the angle is low or high. Additionally, it should be noted that the biggest outliers are generally the data points at $\alpha = 0°$. The reason for this is that at very low angles of attack, the values of lift coefficient are generally low, which translates to a high relative error, even if the absolute difference between the true and predicted values of the coefficient is not large. Table 6.6 shows the values of the MAE of the surrogate models in each of the speed ranges, and the standard deviation of the errors. There, the first column shows that the error of lift coefficient is very good in the subsonic regime, averaging around 5% over all the analysed points, with the highest error of 14.7% at M=0.8, AoA=25°. The standard deviation in this range is also relatively low. Then, in the higher speed ranges, the error increases notably, averaging around 19% and 15% respectively, and so does its standard deviation. If the outlying cases are ignored, the error oscillates around ±10%. For preliminary design phase, this level of expected error could still be found acceptable.

The analysis of the surrogate model of drag coefficient (Figure 6.19b and Figure B.2) shows that in the great majority of cases, the predicted values are lower than the true values. Furthermore, as expected, the lowest accuracy of the surrogate model is in the subsonic range. It comes from the limitations of the low-fidelity solver employed (vide **Section 5.3.2.1**). Restating the point made in that section: panel methods, particularly *ZONAIR*, cannot accurately predict viscous drag, which is a great drag contributor in subsonic range, but this contribution decreases as the speed of flight increases. This trend is well reflected in the results of this study. Above M=1, bar a few outliers, the predicted values of drag coefficient are within ±15% of the true values.

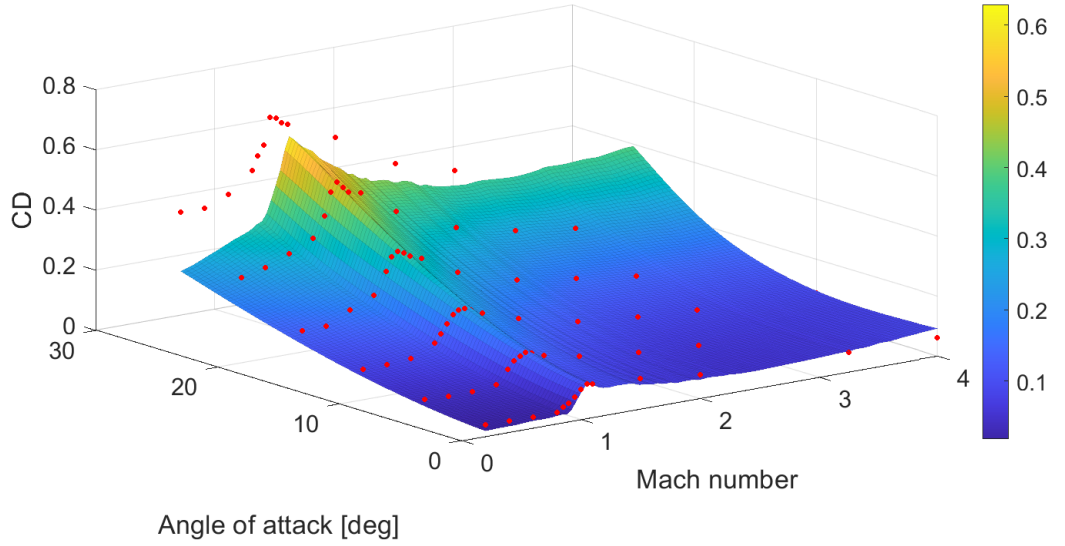
Lastly, the comparison of kriging model of pitching moment coefficient with the expensive data points (Figure 6.19c and Figure B.3) shows that the model is generally inaccurate in all ranges with no significant outliers. While MAEs of drag and pitching moment coefficients are comparable, the relatively high values of MAE of the former are skewed by a couple of outliers. The MAEs of pitching moment coefficient stay at a similarly inaccurate level at the majority of the points. Moreover, the standard deviation of the errors are fairly high, indicating low consistency and low confidence in the predicted results.

Table 6.6: MAEs of the kriging surrogate models in percent relative to the CFD data, and their STDs.

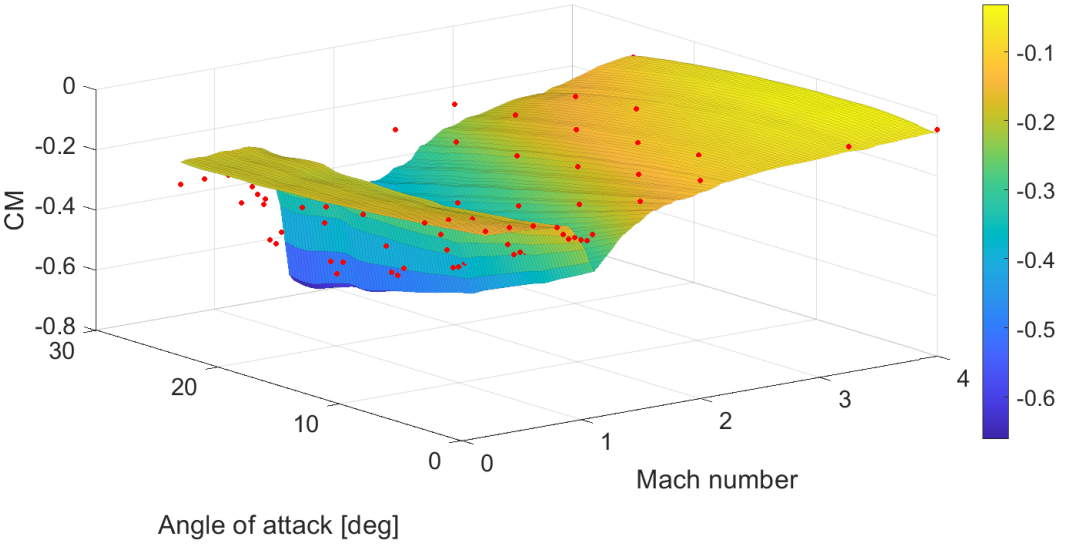
Speed regime	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}
Subsonic	5.622	33.192	16.982	6.66	7.34	15.59
Transonic	18.936	26.149	29.678	25.87	18.60	31.01
Supersonic	15.386	12.902	30.201	29.21	14.28	22.20



(a) Lift coefficient surrogate model



(b) Drag coefficient surrogate model



(c) Pitching moment coefficient surrogate model

Figure 6.19: Surrogate models of longitudinal aerodynamic coefficients created with kriging. Red dots are the expensive data.

6.5.2. Co-kriging results

Now the expensive data points are added for the multi-fidelity interpolation. This section is dedicated to the analysis of the surrogate models created with the data fusion method - co-kriging. Here, the accuracy of the surrogate models is studied in three major aspects:

- the effect of adding higher-fidelity results for the interpolation;
- the influence of the number of the expensive data points;
- the influence of the distribution of the expensive data points.

Before this study is started, it is fundamental to stress one critical observation already made in **Section 6.1**. Namely, that the data set of expensive points is predetermined with a limited amount points - there are 87 of them. The population of expensive data points is divided into two groups: the points used for interpolation and the points used for validation. Logically, if more points are used for interpolation, fewer points are left for validation, and vice-versa. Moreover, the accuracy of the interpolation is dependent not only on the amount of points, but also on the actual *coordinates* of thereof. This interrelation between the two groups might skew the results if not properly accounted for. Thus, to thoroughly study the influence of the number of points as well as their distribution, multiple runs need to be realized. Each run with both groups fully randomized. For a clear and consistent picture, various criteria are used for the analyses. The general fit of the surfaces through the expensive interpolation points is judged qualitatively. Then, using more precise metrics, the errors at each validation point is calculated, and the mean absolute error of each coefficient in each of the distinct speed ranges is computed. Standard deviation of the errors is used to analyse the consistency of the predictions.

There are four separate cases analysed in more detail with an increasing number of expensive interpolation points:

- Case 1: 10 interpolation points (4 delimiting points + 6 random points) - **Section 6.5.2.1**
- Case 2: 20 interpolation points (4 delimiting points + 16 random points) - **Section 6.5.2.2**
- Case 3: 30 interpolation points (4 delimiting points + 26 random points) - **Section 6.5.2.3**
- Case 4: 40 interpolation points (4 delimiting points + 36 random points) - **Section 6.5.2.4**

These four cases have been found to be sufficient to establish the driving factors of the performance of the models. Then, a more general analysis of the trends is carried out in **Section 6.5.2.5**, where more cases are included in the study (with 50, 60, and 70 interpolation points). This time, however, without going into their detailed analyses.

Important notes regarding figures displayed in this section:

- Multiple runs were performed for each case, and presenting all the surrogate models would clutter the document and serve no valuable purpose in explaining the results. Thus, for each case two representative examples of the surrogate models are presented. One representing the best, or *nearly-best*, fit out of the performed runs, and a contrasting one, whose resulting models are on the opposite side of the spectrum. The adjectives *good* and *bad* used in this section to describe the highlighted models are used loosely, and should be treated purely as labels used to differentiate between the two highlighted runs.
- All the figures adhere to the same pattern/legend:
  - colored surface - surrogate model created with co-kriging
  - *shadow* surface - surrogate model created with kriging
  - black dots - expensive data points used for interpolation
  - red dots - expensive data points used for validation

6.5.2.1. Case 1: 10 points

In the first analysed case, ten expensive data points are used for the interpolation - the standard four delimiting, or corner points, and the other six - selected at random from the remaining points. Figure 6.20 shows an example of *relatively well* constructed surrogate models (Run #27 in Table 6.7). In contrast, Figure 6.21 presents an example of *bad* surrogate models created with the same amount of points (Run #22 in Table 6.7). The comparison of the two highlighted runs calls attention to how the created surrogate models can be very different with so few data points. Even with comparable mean absolute errors.

First important observation that can be made based on the results of the thirty performed runs is that adding so few expensive data points does not improve the accuracy of the models compared to pure kriging. On the contrary, there is not a single model that would result in better overall predictions than just single-fidelity, kriging predictions. The MAEs presented in Table 6.7 is just one criterion used in this research. A closer look at the results reveals an inconsistency, leading to high uncertainty of the results. The visualisation of the surfaces (e.g., Figures 6.20 and 6.21), the analysis of each validation data point separately (e.g., Figures C.1-C.3 and C.4-C.6) and the analysis of the standard deviation of the errors (Table C.1) greatly compliment this analysis. All these criteria indicate a very poor performance of models produced with so few expensive data points.

Furthermore, with just ten points, the resulting surrogate models can vary significantly between one another. The worst observed runs generally had one of the following characteristics:

- points located mostly very closely to each other (little spread across the domain);
- points having mostly the same angle of attack or Mach number;
- points mostly located at the edges of the domain, or no points located at the edges of the domain, indicating that a balance should be kept.

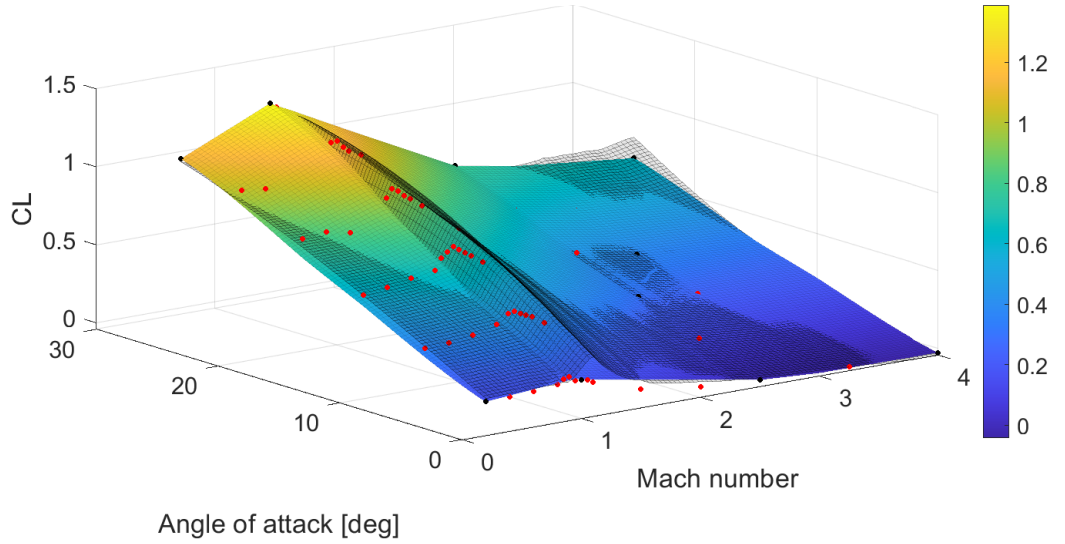
The runs that did not show these characteristics performed much better. The models with the points more uniformly distributed, and with varying values of Mach number and angle of attack perform better than their *concentrated* counterparts. Such runs have lower errors, lower standard deviations, and better reflect the shape of the target surfaces.

Another important fact to note is that the employed optimisation algorithm is non-deterministic. It means that the *optimised* values are not always the actual best fit. While, generally, the calculated values of θ_i and p_i parameters are consistent, during the study it was observed that similarly selected points can lead to very different surrogate models. A possible cause for such discrepancies is the fact that so few expensive data points leave more room for error. This supposition is later confirmed when more expensive points are added for interpolation, and thus the results of the interpolation are much more consistent across all realized runs.

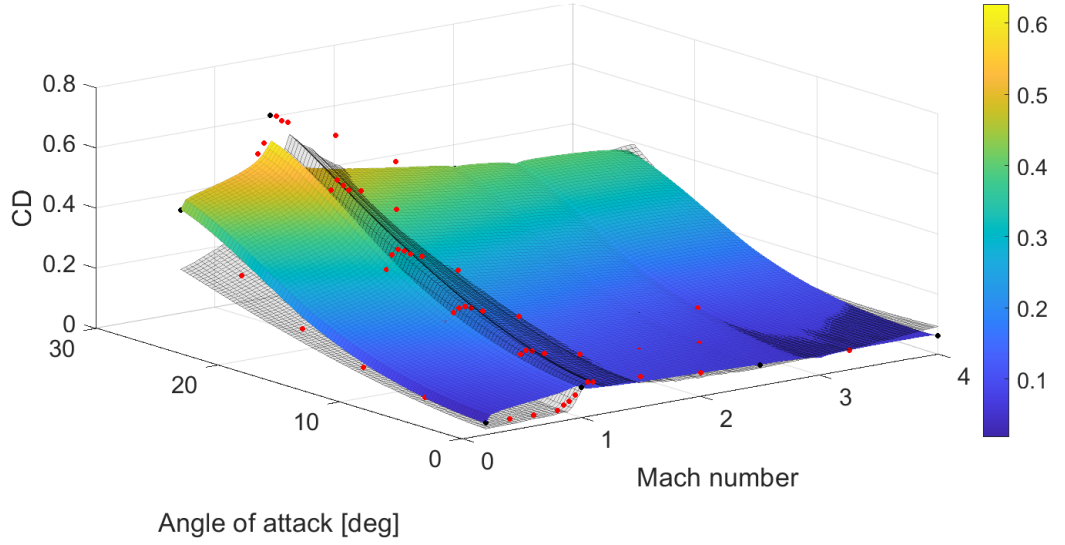
The observations made above show that ten expensive data points are too few to accurately correlate low- and high-fidelity tools. Even though more spread out points do increase the accuracy of the model (compared to concentrated points), the overall accuracy is still lower than that of just single-fidelity, kriging interpolation. The results presented in this section highlight that more points need to be used if the accuracy of the model is to be increased.

Table 6.7: MAEs of the ten runs with 4 delimiting and 6 random points used for interpolation.

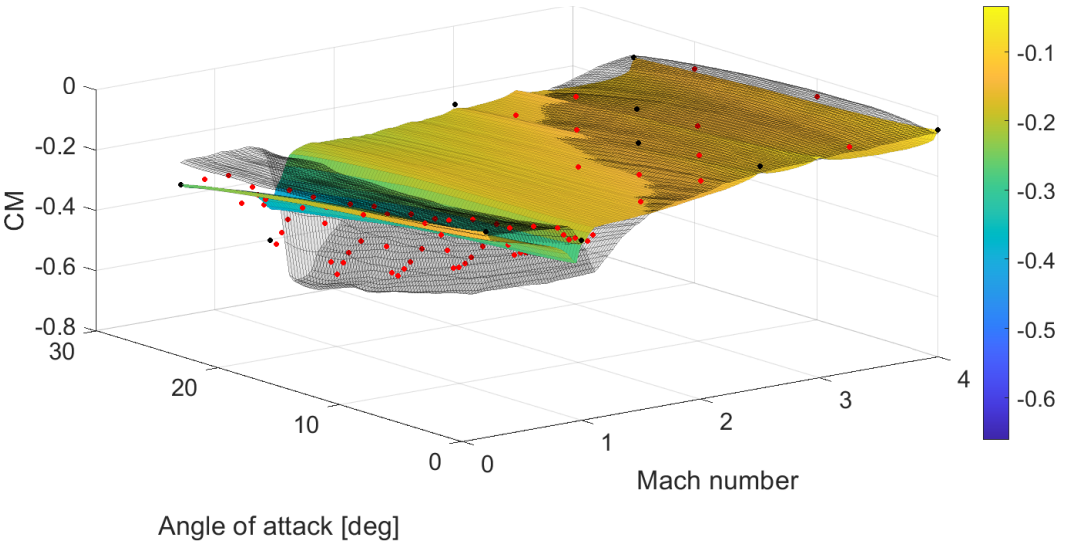
	Subsonic			Transonic			Supersonic		
Run	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}
1	9.98	39.84	14.41	19.28	41.99	32.29	38.18	43.81	26.35
2	5.67	30.26	12.51	17.37	26.05	32.17	28.31	12.66	44.75
3	16.73	43.40	21.76	41.68	38.38	46.69	55.36	50.35	48.54
4	13.32	22.71	48.80	16.07	20.02	56.84	28.04	15.30	100.27
5	17.77	29.31	23.64	4.82	8.44	14.03	62.15	56.83	56.38
6	7.69	38.17	38.20	6.41	12.51	15.03	64.25	21.39	52.41
7	10.10	40.92	33.98	12.37	17.30	38.76	35.61	14.92	75.54
8	8.65	59.09	11.82	11.32	19.10	20.86	22.22	12.75	27.08
9	16.48	28.29	89.54	13.31	19.43	65.75	33.21	9.12	29.68
10	18.21	19.99	26.71	25.03	11.72	41.24	35.53	77.87	78.76
11	7.92	33.15	28.61	17.24	19.19	41.52	21.64	11.66	31.40
12	9.54	41.97	64.21	8.30	25.84	98.79	49.66	43.84	317.11
13	8.75	81.83	10.88	11.06	36.51	32.35	44.36	46.59	29.90
14	19.13	54.93	23.24	15.46	13.89	25.90	54.52	51.55	100.88
15	37.48	95.17	21.18	27.89	26.29	26.39	92.49	24.76	28.66
16	27.56	83.14	49.74	7.40	24.94	11.40	71.83	48.92	33.65
17	8.46	46.45	19.04	8.01	22.45	29.31	45.54	18.30	22.38
18	12.16	37.22	46.92	13.36	22.12	43.43	42.60	14.01	123.14
19	15.51	97.93	80.71	20.77	42.76	73.72	18.90	60.78	198.19
20	8.89	28.99	6.19	22.11	29.86	39.00	23.09	14.63	57.79
21	10.15	28.13	6.18	22.47	37.15	24.50	22.99	19.32	14.11
22	12.59	42.24	56.10	11.46	13.96	27.06	28.72	24.40	103.25
23	20.60	25.36	50.11	21.56	24.11	96.14	85.02	47.43	213.29
24	9.78	39.83	10.61	9.30	17.23	22.79	42.76	29.90	103.86
25	5.93	35.07	40.84	9.20	10.56	19.85	11.57	8.37	24.33
26	5.69	18.49	21.11	11.45	22.03	34.51	44.74	27.93	86.59
27	9.86	45.41	15.17	11.28	20.83	31.10	17.85	10.73	27.79
28	13.83	16.75	98.24	19.25	17.76	39.18	57.96	105.54	316.59
29	14.45	47.10	118.35	12.02	18.17	26.39	20.48	32.98	72.94
30	11.25	43.97	77.46	8.27	13.15	53.95	36.44	15.76	79.59
Mean	13.14	43.17	38.88	15.18	22.46	38.70	41.20	32.41	84.17
Kriging	5.62	33.19	16.98	18.94	26.15	29.68	15.39	12.90	30.20



(a) Lift coefficient surrogate model

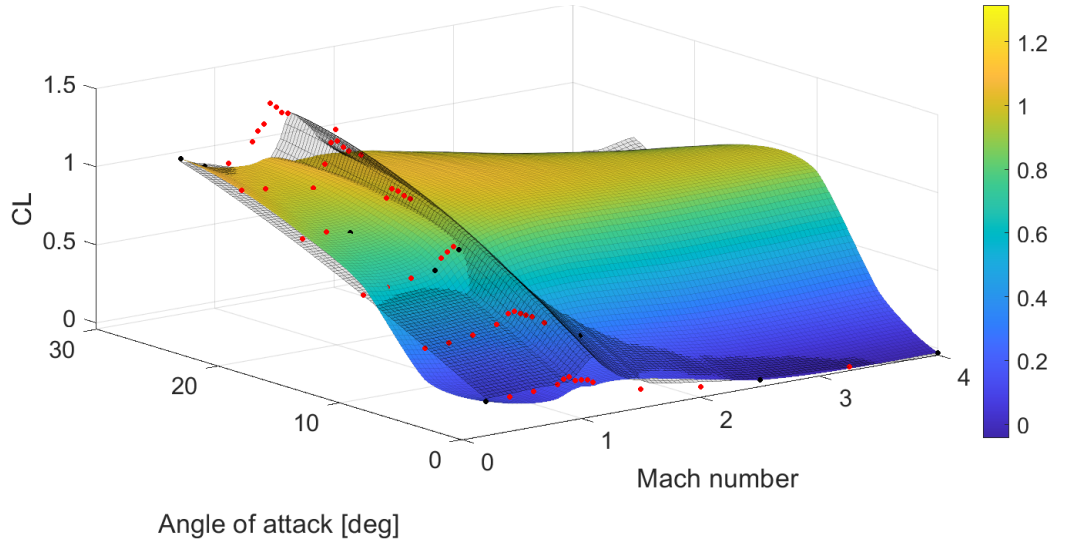


(b) Drag coefficient surrogate model

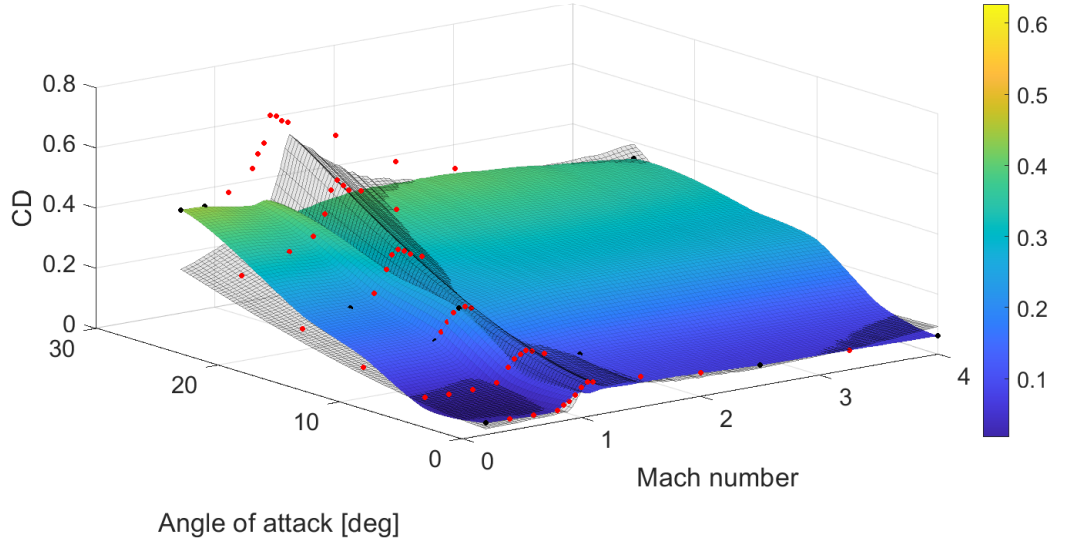


(c) Pitching moment coefficient surrogate model

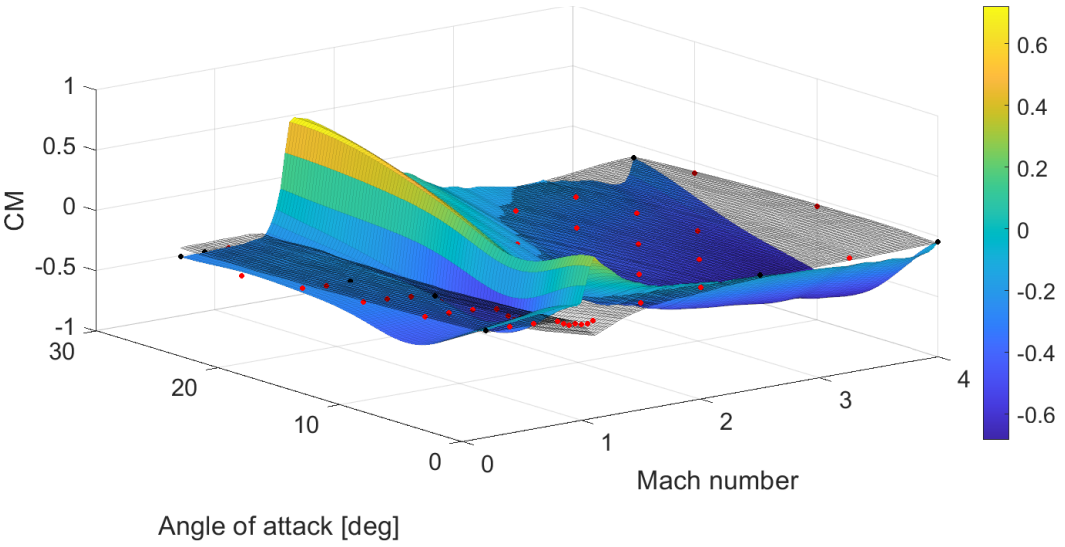
Figure 6.20: Example of relatively good surrogate models of longitudinal aerodynamic coefficients created with co-kriging using 10 expensive data points. Run#27 from Table 6.7.



(a) Lift coefficient surrogate model



(b) Drag coefficient surrogate model



(c) Pitching moment coefficient surrogate model

Figure 6.21: Example of bad surrogate models of longitudinal aerodynamic coefficients created with co-kriging using 10 expensive data points. Run#23 from Table 6.7.

6.5.2.2. Case 2: 20 points

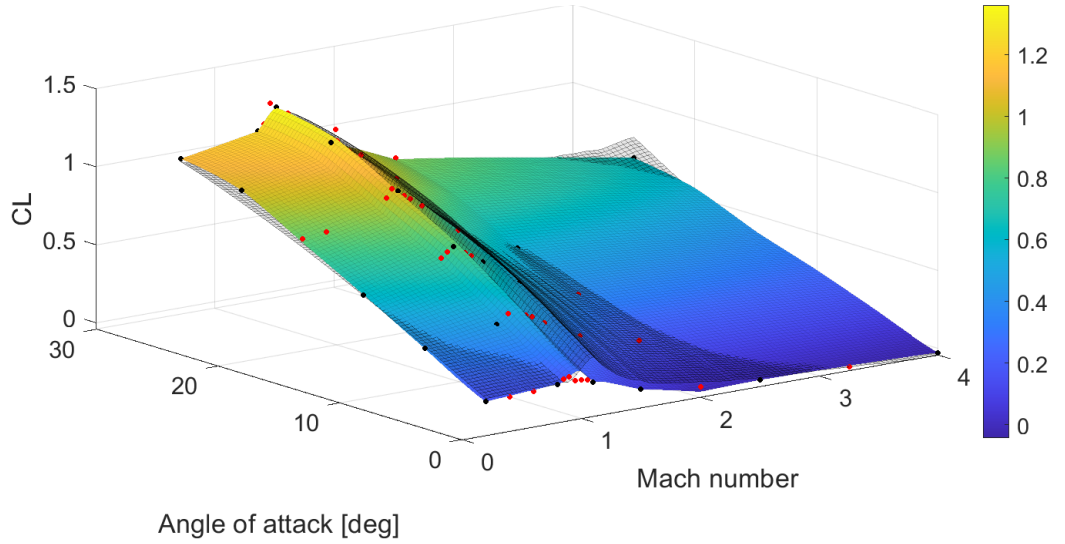
The previous case showed that, at least with the available data, ten points are too few to create accurate surrogate models. Hence, as the next step of this study, ten more points are added to the interpolation group. The exact same approach of fully randomized interpolation/validation groups is adopted for this case. The only difference is the size of the two groups. From the first look at some examples of the new surrogate surfaces, a big improvement can be observed. Figure 6.22 shows example surrogate models generated with twenty points that is on the *more accurate side of the spectrum*, while Figure 6.23 presents an example of the opposite. The former is a visible improvement over the previously analysed case, while the latter still shows poor accuracy.

The two presented examples illustrate how big a role the distribution of the points across the domain plays in the accuracy of the models. The *bad* models in Figure 6.23 were produced with points mostly located in the low-Mach-number range, and almost exclusively at angles of attack of 5° and 10°. It causes problems for the algorithm to accurately predict the coefficients in the higher Mach region. It is especially noticeable in the transonic region, where the true values (red dots) can be seen far above the force coefficient surfaces, and far below the pitching moment coefficient surface. The supersonic region is also not too well predicted. In contrast, the *good* models in Figure 6.22 have their interpolation points more spread out, more of them located in the transonic region where the most rapid changes in the coefficients should be expected, and at varying angles of attack and Mach numbers. Because of this, the model produces better fitted surfaces with a higher confidence in its predictions.

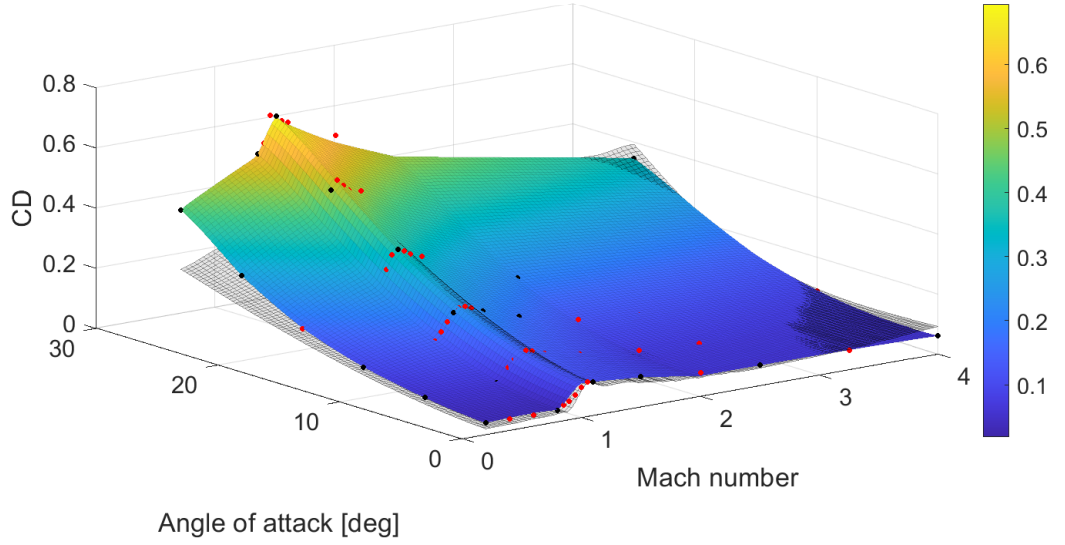
Table 6.8 shows the MAEs of each of the thirty runs realized for the twenty-point case. Straight away it can be noted that the results are drastically improved over the previous case. The majority of the runs show the same or higher accuracy than pure kriging. With a closer look at the results, through the already discussed visualisation of the full surfaces (Figures 6.22 and 6.23), and the errors at each data point separately (e.g., Figures C.7-C.9 and C.10-C.12), this improvement of the models is further emphasized. Not only are the MAEs lower, but the overall predictions are also more consistent leading to higher confidence in the results (Table C.2). It should be noted that already with twenty high-fidelity data points, the overall accuracy can be increased over single-fidelity-interpolated models.

Table 6.8: MAEs of the thirty runs with 4 delimiting and 16 random points used for interpolation.

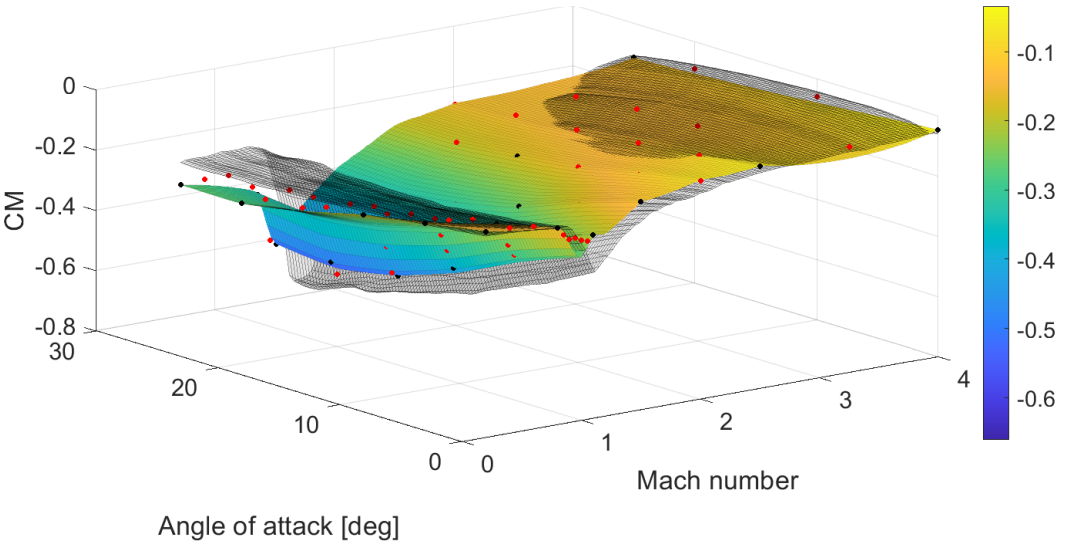
	Subsonic			Transonic			Supersonic		
Run	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}
1	4.72	15.92	20.07	3.63	6.78	5.27	14.21	9.78	24.88
2	6.71	14.48	13.32	7.43	9.00	13.87	7.57	8.20	12.21
3	7.14	19.64	14.35	3.98	6.56	7.29	20.74	16.02	24.72
4	9.95	29.83	19.61	14.33	15.80	18.43	22.97	10.35	20.69
5	4.51	20.68	17.32	7.88	10.97	10.11	8.03	6.29	14.17
6	4.23	10.83	17.48	2.50	7.93	7.33	24.31	13.32	41.56
7	10.97	19.79	9.73	23.13	13.45	24.41	16.93	9.96	23.15
8	5.84	18.21	12.97	11.36	17.92	16.99	12.08	8.48	10.91
9	9.16	17.40	15.61	9.07	23.45	41.81	15.01	59.12	33.65
10	6.58	12.18	17.02	9.53	6.61	10.54	17.88	6.23	43.03
11	9.13	18.40	14.37	4.60	7.13	11.19	31.08	23.83	39.12
12	9.91	30.18	17.81	10.31	14.71	14.07	15.45	10.98	15.61
13	8.56	13.42	24.21	3.13	7.90	12.09	14.07	8.93	33.24
14	5.61	8.71	5.51	9.22	8.08	14.09	22.80	6.13	9.78
15	6.88	18.82	9.55	6.03	11.87	20.12	7.82	4.37	21.37
16	3.42	7.77	6.03	10.35	4.82	11.73	13.89	7.93	10.97
17	6.67	12.09	11.98	4.94	4.82	7.02	19.59	7.75	14.92
18	11.63	39.72	27.18	14.47	24.09	27.23	14.30	5.34	24.58
19	7.75	27.81	7.77	8.06	12.14	14.48	14.67	11.86	10.30
20	7.04	18.20	6.41	3.67	9.85	7.80	28.28	19.22	34.41
21	6.41	23.03	10.27	3.77	7.80	4.42	14.72	7.37	11.87
22	8.49	17.19	13.11	5.92	13.63	11.00	12.62	8.14	9.10
23	8.87	18.97	10.40	6.04	10.15	13.45	27.60	7.92	26.74
24	10.75	19.63	7.67	11.83	9.79	11.36	18.66	11.12	18.51
25	7.39	11.14	11.08	6.79	5.72	10.78	24.03	22.28	32.54
26	6.77	13.88	14.49	5.32	5.99	9.22	26.23	11.79	23.31
27	8.10	32.56	19.77	7.64	20.69	14.12	21.51	21.98	51.22
28	6.84	13.59	5.74	7.12	12.02	10.79	49.79	46.33	24.39
29	10.49	24.51	24.40	7.97	11.76	11.21	11.29	8.01	16.20
30	4.40	8.89	8.70	13.38	18.10	25.02	14.03	6.67	9.68
Mean	7.50	18.58	13.80	8.11	11.32	13.91	18.74	13.52	22.89
Kriging	5.62	33.19	16.98	18.94	26.15	29.68	15.39	12.90	30.20



(a) Lift coefficient surrogate model

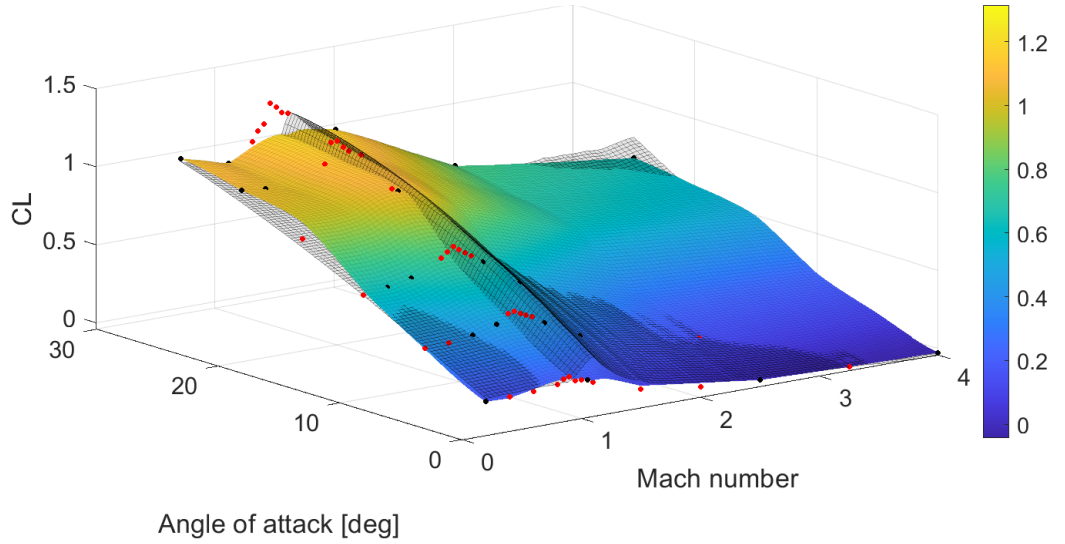


(b) Drag coefficient surrogate model

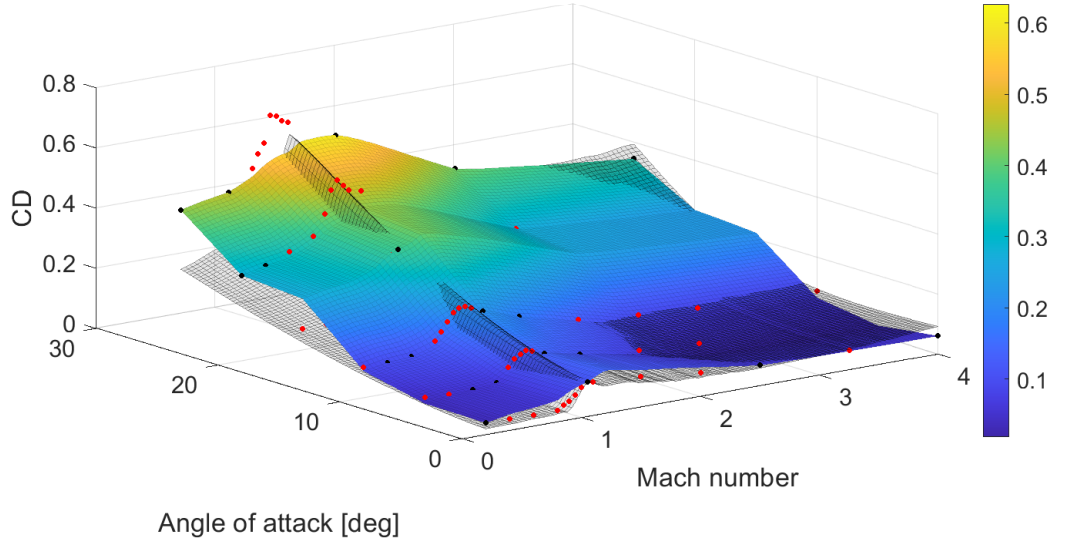


(c) Pitching moment coefficient surrogate model

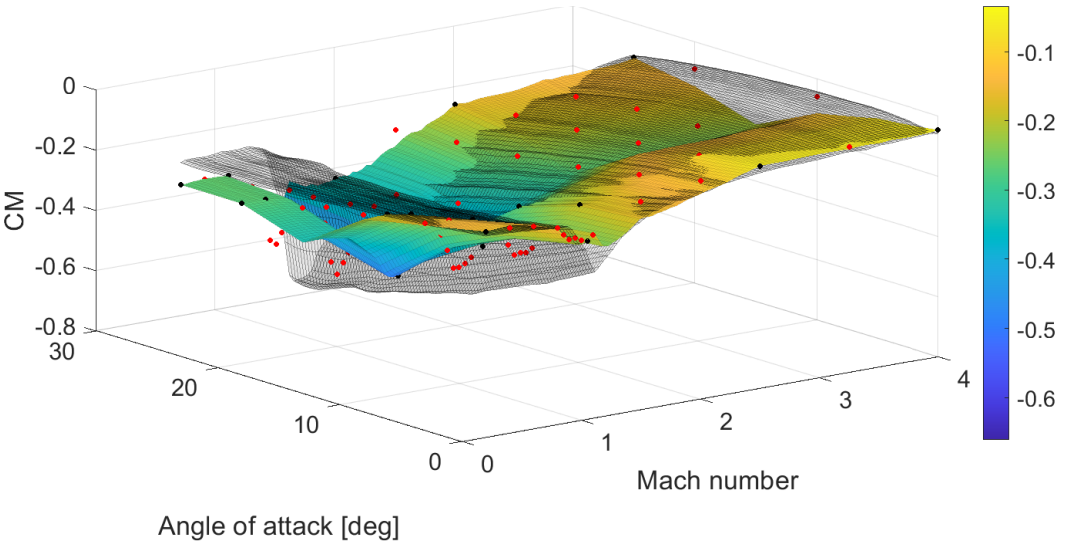
Figure 6.22: Example of relatively good surrogate models of longitudinal aerodynamic coefficients created with co-kriging using 20 expensive data points. Run#17 from Table 6.8.



(a) Lift coefficient surrogate model



(b) Drag coefficient surrogate model



(c) Pitching moment coefficient surrogate model

Figure 6.23: Example of relatively bad surrogate models of longitudinal aerodynamic coefficients created with co-kriging using 20 expensive data points. Run#27 from Table 6.8.

6.5.2.3. Case 3: 30 points

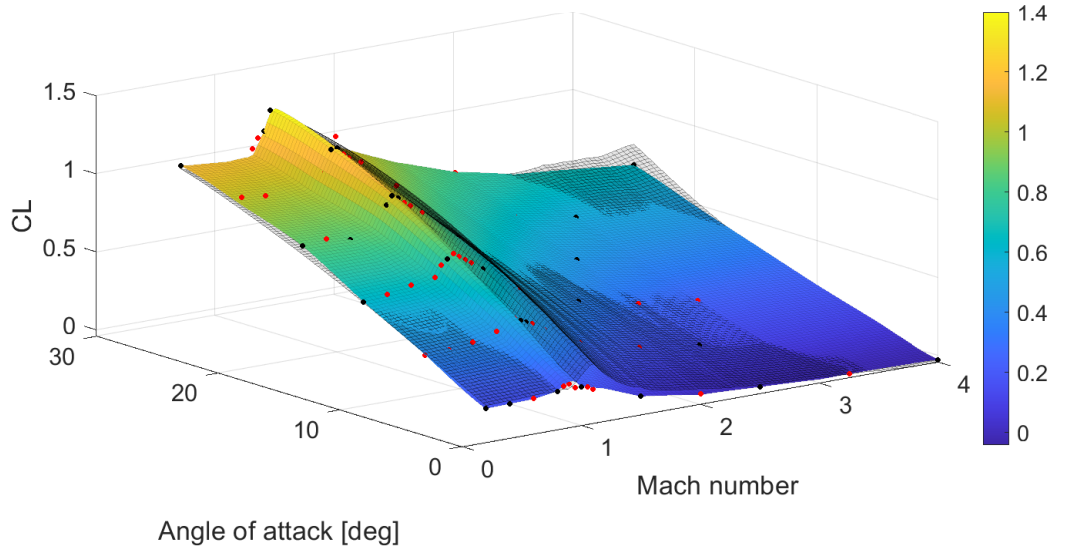
Similarly to what was done before, another ten points are added for interpolation to further analyse the change in accuracy of the models. With thirty points now used for this task, the improvement of the surrogate surfaces is clearly visible. Take for example Figure 6.24. The models generated during this run show a high enhancement in the predictions, dictated by the elevated number of points, their higher spread across the domain and a slight intensification in the region of highest changes - the transonic speed regime. All the metrics used in this study confirm the higher accuracy and consistency of the predictions made by this model. It shows that with a good distribution of points across the domain, thirty points can produce models of very high accuracy. The best produced model in the presented case has all its MAEs at around 5%, apart from pitching moment in the supersonic regime, which is not much higher at below 8%. The standard deviations of the model are mostly well below 10, indicating low uncertainty of the predictions.

In contrast to Run #25, Figure 6.25 shows that even an increased amount of points cannot compensate for their poor distribution. In some metrics, this model performs even worse than the highlighted *good* model constructed with just twenty points. This case further stresses the need for good distribution of points if a quality model is to be produced.

Overall MAEs presented in Table 6.9 show a general improvement in the predictions over the previously discussed, *less populated* cases, and, similarly, over the kriging models. Standard deviations (Table C.3) also confirm higher confidence in the predicted results, for the best models as well as on average.

Table 6.9: MAEs of the thirty runs with 4 delimiting and 26 random points used for interpolation.

	Subsonic			Transonic			Supersonic		
Run	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}
1	5.41	7.17	5.62	3.22	9.00	5.88	5.31	5.31	4.80
2	6.79	9.56	5.58	2.84	5.60	5.26	8.27	4.25	7.54
3	7.21	24.21	13.64	3.62	8.32	5.02	3.25	3.63	6.41
4	4.30	16.83	23.82	2.92	6.79	7.37	4.71	6.91	21.80
5	4.18	9.20	4.31	6.14	9.45	10.48	7.42	8.15	8.74
6	4.76	11.85	4.81	1.95	2.41	4.55	9.20	5.20	17.71
7	4.50	8.77	5.57	2.42	5.55	4.13	7.60	4.22	7.52
8	9.12	11.82	8.25	4.02	5.95	5.28	20.83	4.98	10.26
9	5.17	6.30	5.09	4.55	9.66	7.49	10.47	4.43	9.81
10	4.61	3.38	4.13	2.75	5.47	7.05	7.73	7.05	17.81
11	6.70	9.55	4.48	3.59	3.69	4.94	6.92	5.29	13.23
12	4.71	9.64	5.60	3.44	7.58	3.18	10.65	9.37	9.89
13	3.79	5.15	3.25	4.03	6.73	6.02	15.66	7.31	29.39
14	6.75	13.68	12.10	4.79	5.27	9.76	9.45	4.32	15.79
15	7.54	10.99	8.13	12.43	8.34	18.47	22.57	8.84	11.75
16	3.77	5.77	4.29	5.42	3.85	8.19	3.34	4.35	9.61
17	5.84	9.50	7.73	4.54	7.52	5.33	8.31	10.33	15.00
18	5.11	7.46	7.24	4.55	4.94	9.19	3.62	3.47	17.20
19	7.81	9.99	7.35	4.89	6.92	7.34	7.14	5.30	6.04
20	5.21	8.71	7.28	3.66	4.58	5.94	11.12	8.75	14.79
21	5.72	5.29	8.94	4.00	4.50	5.65	4.95	3.25	9.08
22	4.23	9.08	3.84	4.31	5.46	6.02	12.92	6.52	6.92
23	9.27	11.82	8.75	5.64	5.51	18.11	13.51	6.50	17.89
24	4.45	5.01	5.20	3.45	3.82	5.53	9.65	6.70	29.32
25	3.62	4.62	4.12	3.06	5.37	3.54	4.14	3.51	7.76
26	4.50	7.14	6.52	2.76	4.20	6.74	5.87	5.42	11.44
27	6.89	9.34	8.04	4.54	6.48	4.14	5.80	3.39	5.34
28	5.14	10.06	7.59	4.71	7.00	7.60	7.22	5.30	11.11
29	6.64	4.72	6.74	3.77	7.82	9.11	3.50	4.43	4.20
30	10.89	10.44	12.80	6.65	5.77	8.51	10.55	6.28	13.81
Mean	5.82	9.23	7.36	4.29	6.12	7.19	8.72	5.76	12.40
Kriging	5.62	33.19	16.98	18.94	26.15	29.68	15.39	12.90	30.20

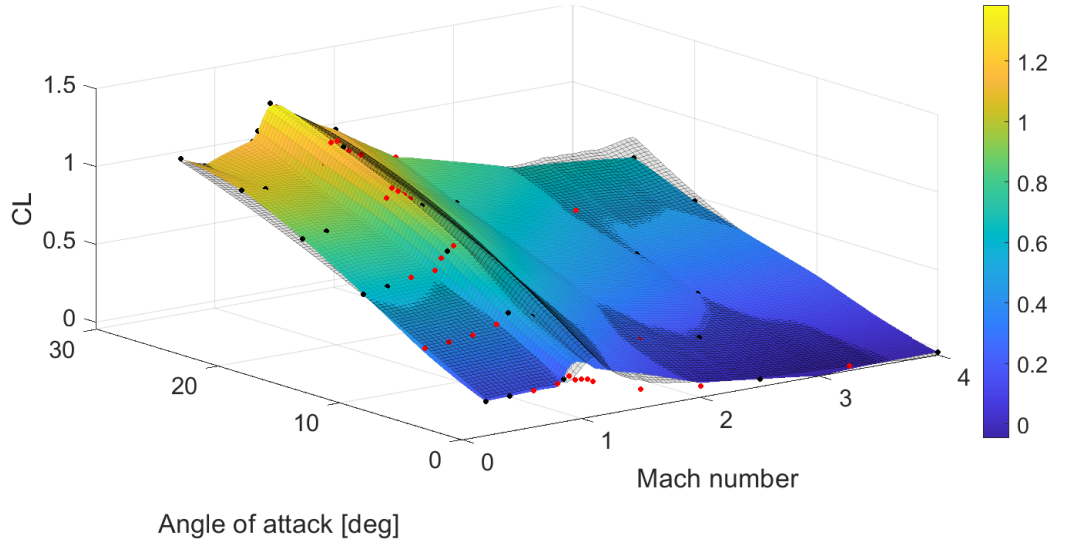


(a) Lift coefficient surrogate model

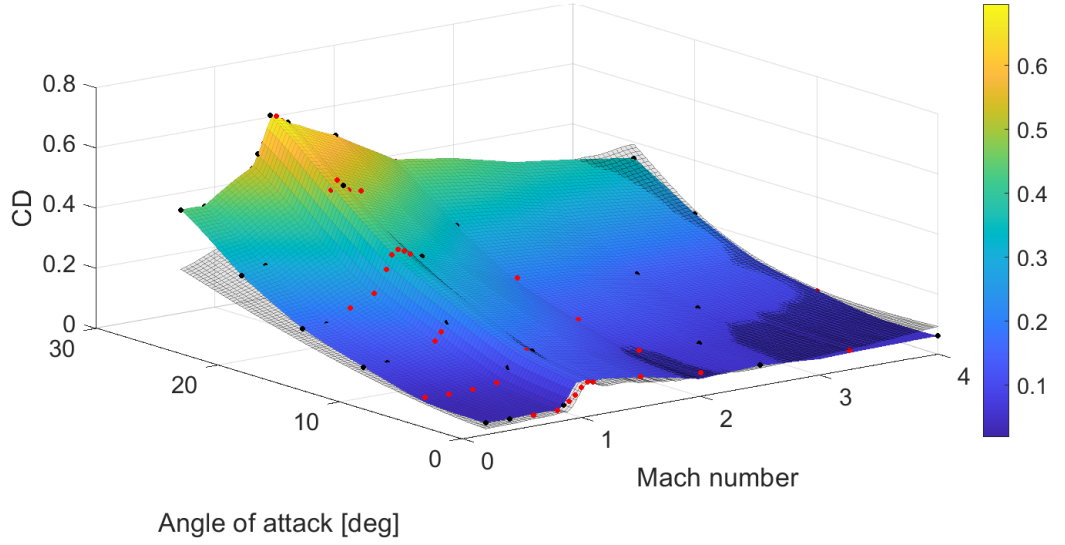
(b) Drag coefficient surrogate model

(c) Pitching moment coefficient surrogate model

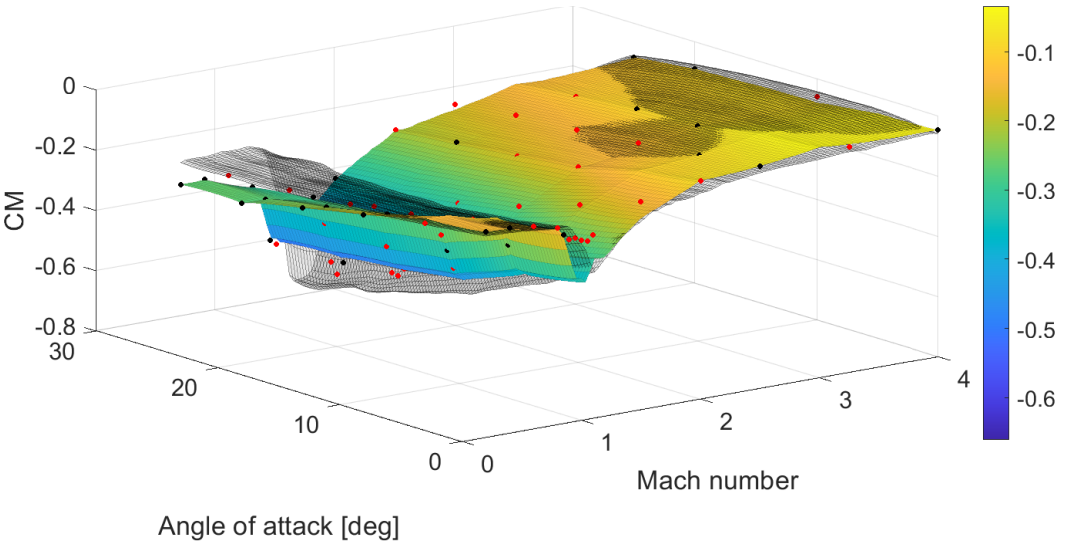
Figure 6.24: Example of relatively good surrogate models of longitudinal aerodynamic coefficients created with co-kriging using 30 expensive data points. Run#25 from Table 6.9.



(a) Lift coefficient surrogate model



(b) Drag coefficient surrogate model



(c) Pitching moment coefficient surrogate model

Figure 6.25: Example of relatively bad surrogate models of longitudinal aerodynamic coefficients created with co-kriging using 30 expensive data points. Run#15 from Table 6.9.

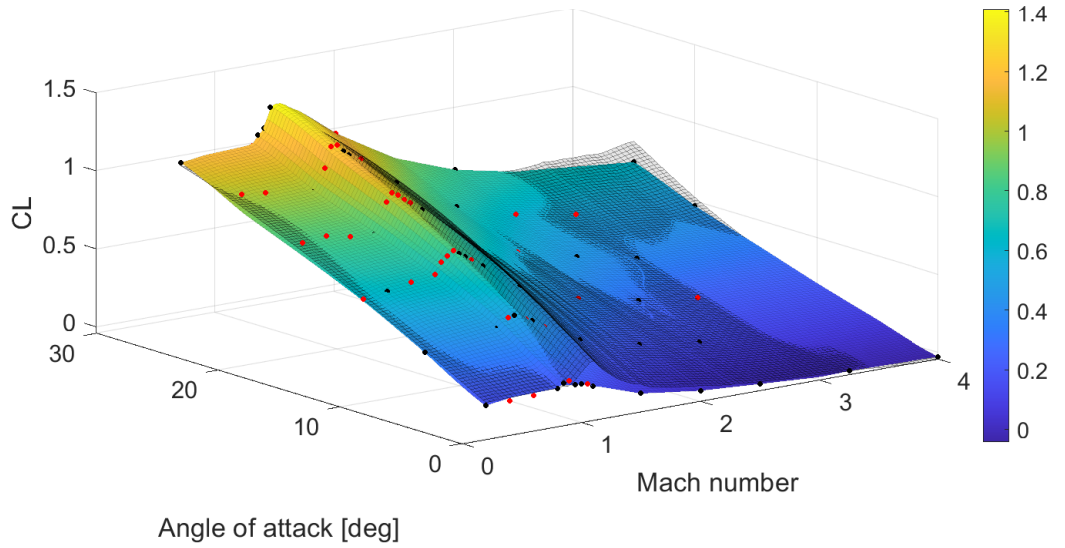
6.5.2.4. Case 4: 40 points

The last case presented in more detail adds another ten points to the interpolation pool, which now stands at forty points. Once again, thirty runs are performed with randomized interpolation and validation groups. The most accurate of the generated models (Figure 6.26) shows a very high level of accuracy in the entire domain for all the coefficients of interest. Not only are the MAEs very low (Table 6.10), mostly well within 5% error, but the relative errors at the verification points (Figures C.19-C.21) also remain consistent, and mostly close to the true values. The errors generally do not exceed $\pm$10%, bar a very few outliers, all of which are at angles of attack of 0°. Meaning, that the higher value of the relative error is mostly dictated by the low absolute value of the coefficients, which tends to inflate the errors. Standard deviations are similarly very low, mostly oscillating below 5, providing a rather high confidence in the predictions made by the models.

The contrasting run (Figures 6.27 and C.22-C.24), which is the *worst* out of the generated in this case, still performs well in comparison with most other models discussed in the sections above. The *issue* with this model is that it has very few points at low angles of attack, causing the models to make poor predictions at the low-angle-of-attack range. Due to low absolute values, this can lead to increased relative errors. However, even with this in mind, this model performs worse than most of the other ones created in this case. It yet again emphasizes the points already made during the discussion of the previous cases: good distribution of points in the domain plays a vital role in the construction of accurate surrogate models. The comparison of the *bad* models from this case with the *good* thirty-point model from the previous case shows that the raw number of points added for interpolation cannot compensate for their poor distribution.

Table 6.10: MAEs of the thirty runs with 4 delimiting and 36 random points used for interpolation.

	Subsonic			Transonic			Supersonic		
Run	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}
1	6.81	8.92	5.77	1.36	2.02	3.22	12.17	6.04	10.16
2	4.02	10.38	9.33	2.65	3.63	4.45	2.67	4.92	2.88
3	7.34	4.84	5.28	2.03	3.70	3.48	8.38	5.84	7.24
4	8.86	5.90	7.17	4.41	10.06	4.35	9.33	5.86	13.28
5	7.31	8.59	8.34	2.83	4.19	4.07	4.41	4.59	7.79
6	7.47	6.16	6.03	3.05	3.52	2.70	3.23	3.16	9.83
7	7.43	8.90	7.03	2.85	2.54	3.70	6.35	4.36	8.70
8	6.22	9.40	7.31	5.25	5.32	4.89	4.88	8.13	7.75
9	5.69	6.42	5.67	2.28	2.41	6.09	2.19	2.43	4.23
10	4.90	5.41	4.04	2.67	2.85	3.45	8.38	3.83	17.29
11	5.25	9.11	6.22	2.77	4.10	3.99	2.44	2.83	3.96
12	4.95	10.58	4.42	3.49	6.84	4.41	17.54	7.35	8.63
13	3.35	10.40	2.91	2.08	3.22	3.75	7.77	4.61	16.06
14	3.74	8.00	3.61	1.57	2.84	3.37	5.17	2.63	15.11
15	6.03	13.88	13.09	1.59	2.61	5.08	5.78	5.53	11.59
16	7.52	5.82	6.98	4.24	2.44	4.77	3.85	2.26	4.06
17	6.34	8.73	3.84	1.74	4.33	4.92	8.21	4.00	9.76
18	4.08	8.56	5.32	3.22	3.82	3.65	4.25	3.25	7.35
19	6.26	6.62	4.34	4.82	3.84	6.96	2.90	3.26	3.21
20	4.86	8.70	6.41	2.10	4.05	3.40	6.05	3.90	10.16
21	5.65	7.83	6.23	4.44	4.81	5.29	4.42	3.34	5.38
22	3.59	10.36	3.89	2.38	2.49	2.82	5.98	4.34	8.66
23	3.18	5.88	4.72	2.92	3.77	4.44	6.58	4.41	7.81
24	6.82	7.58	5.89	5.55	7.89	6.16	4.23	7.79	4.94
25	4.21	6.91	3.34	3.65	7.09	3.62	3.02	3.67	8.73
26	5.01	7.40	4.97	2.50	4.95	4.05	3.35	4.80	4.73
27	5.14	17.28	5.45	6.83	10.66	8.44	12.14	3.72	9.72
28	4.72	9.13	5.91	2.28	5.53	3.85	13.31	5.21	18.23
29	5.42	8.52	5.92	5.65	3.15	5.83	7.75	4.35	17.32
30	4.15	10.50	8.70	2.79	4.23	4.71	6.36	4.70	6.96
Mean	5.54	8.56	5.94	3.20	4.43	4.46	6.44	4.50	9.05
Kriging	5.62	33.19	16.98	18.94	26.15	29.68	15.39	12.90	30.20

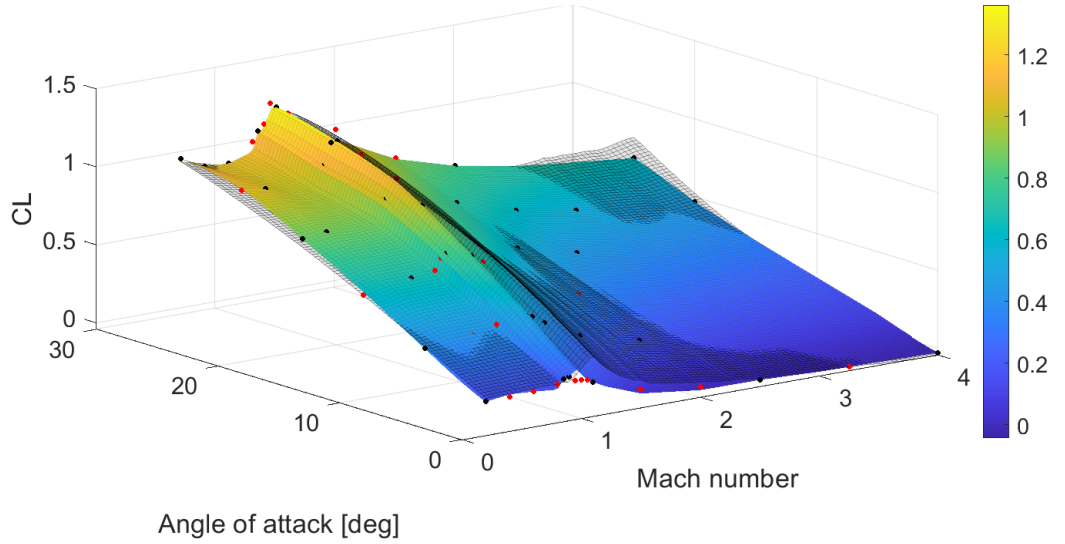


(a) Lift coefficient surrogate model

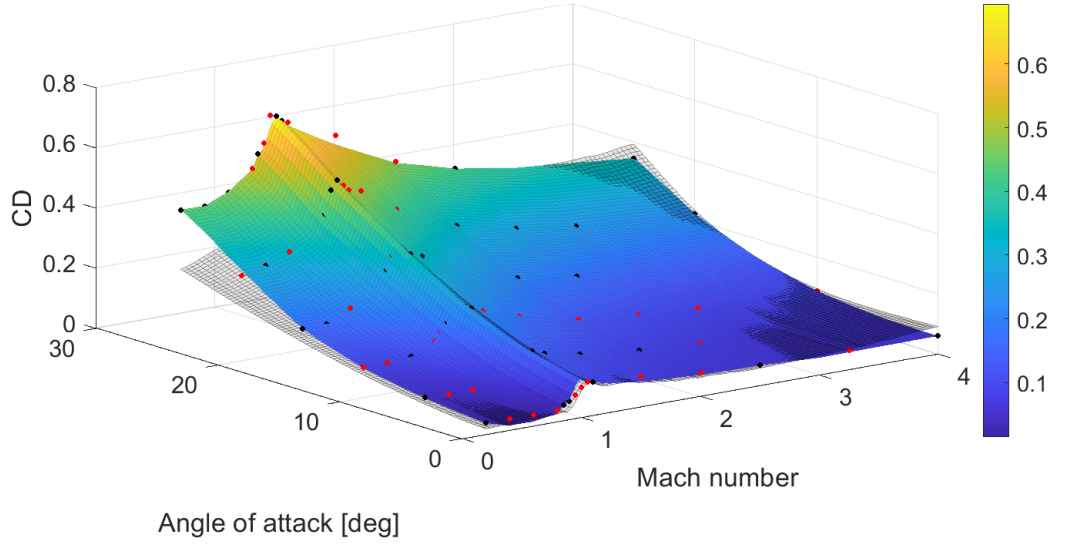
(b) Drag coefficient surrogate model

(c) Pitching moment coefficient surrogate model

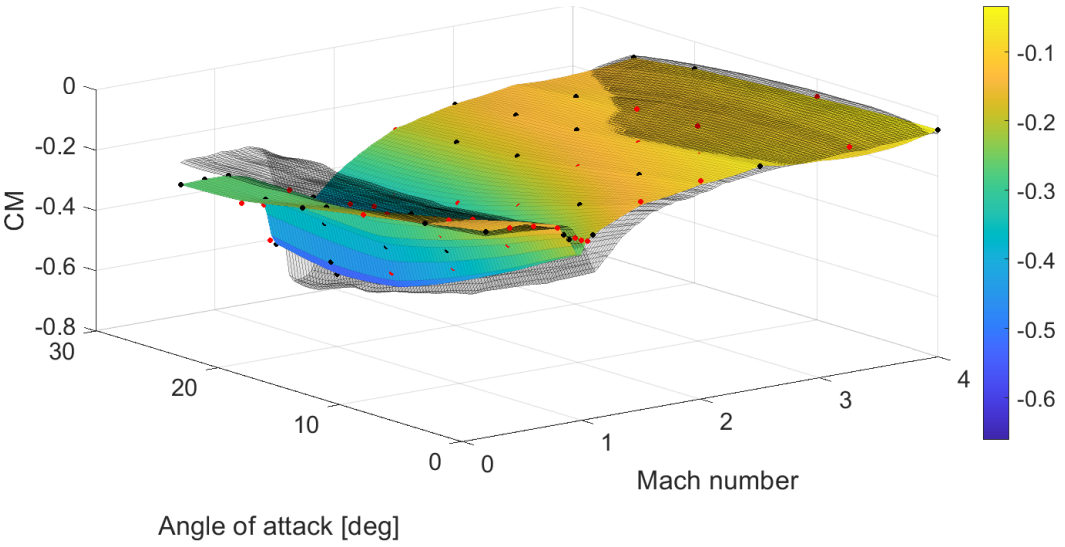
Figure 6.26: Example of relatively good surrogate models of longitudinal aerodynamic coefficients created with co-kriging using 40 expensive data points. Run#19 from Table 6.10.



(a) Lift coefficient surrogate model



(b) Drag coefficient surrogate model



(c) Pitching moment coefficient surrogate model

Figure 6.27: Example of relatively bad surrogate models of longitudinal aerodynamic coefficients created with co-kriging using 40 expensive data points. Run#27 from Table 6.10.

6.5.2.5. General observations

The general trends in the variation of the accuracy of the created models with increasing number of expensive data points are the subject of this section. In the previous sections, two factors were already observed to have a major influence on the accuracy of the models - the number of points, and the distribution of thereof in the domain of interest. It could already be seen that the accuracy of the models increases as the amount of CFD points grows, and, by extension, a more spread out distribution is provided imposed by the predetermined character of the expensive data set. Figure 6.28 further confirms this observation - as there are more points used for the interpolation, and on average a larger spread of thereof is supplied to the algorithm, the mean absolute errors as well as their standard deviations drop. All the errors converge to around 2%-3.5%, apart from pitching moment coefficient in the supersonic regime, and drag coefficient in the subsonic regime. The standard deviations are also the largest for these two coefficients, meaning that the confidence in these results would be the lowest. It should also be noted that above 50 points employed for the interpolation, the errors and their standard deviations, on average, do not change significantly.

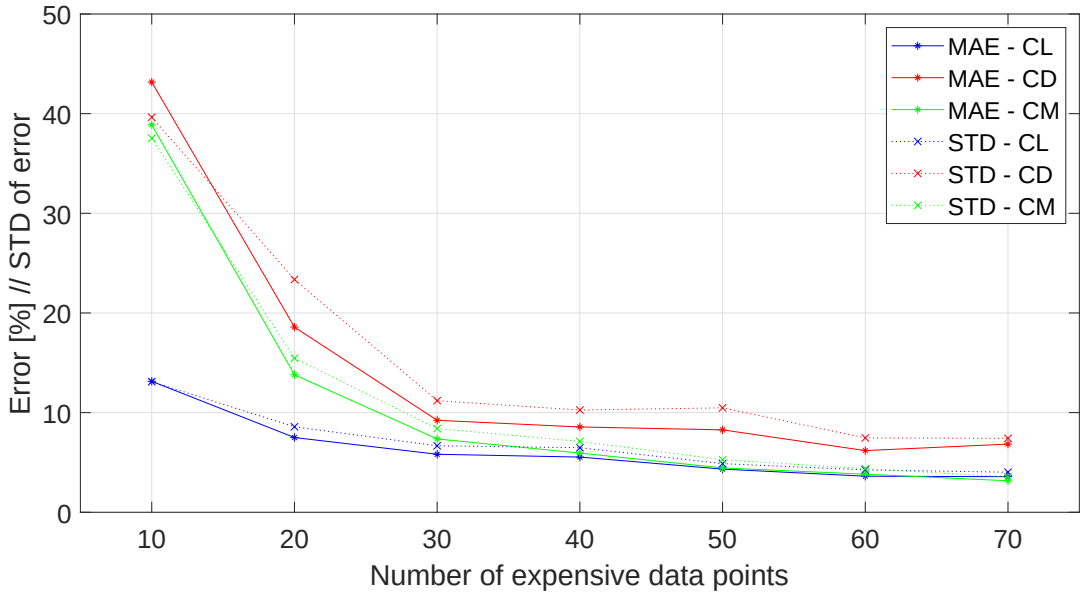
During the research, a high-performance computing cluster was used for the high-fidelity calculations. Even with this advanced technology each simulation required between 6 and 12 hours for most cases, but sometimes even more if mesh refinement was found to be necessary. From this, it becomes obvious that even a minor decrease in the quantity of performed high-fidelity simulations amounts to big time saving. By employing this data fusion technique, even just 30 points are enough to limit the expected error to well below 10% for the analysed coefficients, and this number could be further reduced if a better point distribution is found through, e.g., sampling algorithms.

Table 6.11: Mean absolute errors (in percent) across all the analysed cases, averaged over 30 runs.

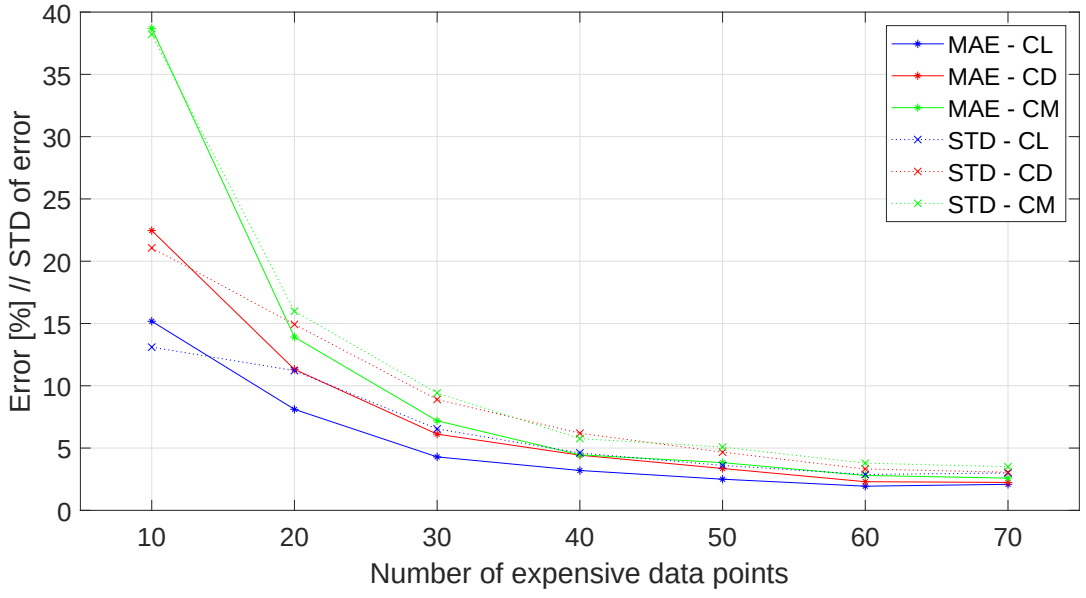
	Subsonic			Transonic			Supersonic		
Points	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}	ϵ_{C_L}	ϵ_{C_D}	ϵ_{C_M}
10	13.14	43.17	38.88	15.18	22.46	38.70	41.20	32.41	84.17
20	7.50	18.58	13.80	8.11	11.32	13.91	18.74	13.52	22.89
30	5.82	9.23	7.36	4.29	6.12	7.19	8.72	5.76	12.40
40	5.54	8.56	5.94	3.20	4.43	4.46	6.44	4.50	9.05
50	4.33	8.27	4.45	2.50	3.36	3.83	4.51	3.96	6.98
60	3.62	6.19	3.82	1.94	2.30	2.80	4.73	3.61	6.61
70	3.57	6.84	3.16	2.09	2.24	2.59	3.45	3.29	5.15

Table 6.12: Standard deviation of mean absolute errors across all the analysed cases, averaged over 30 runs.

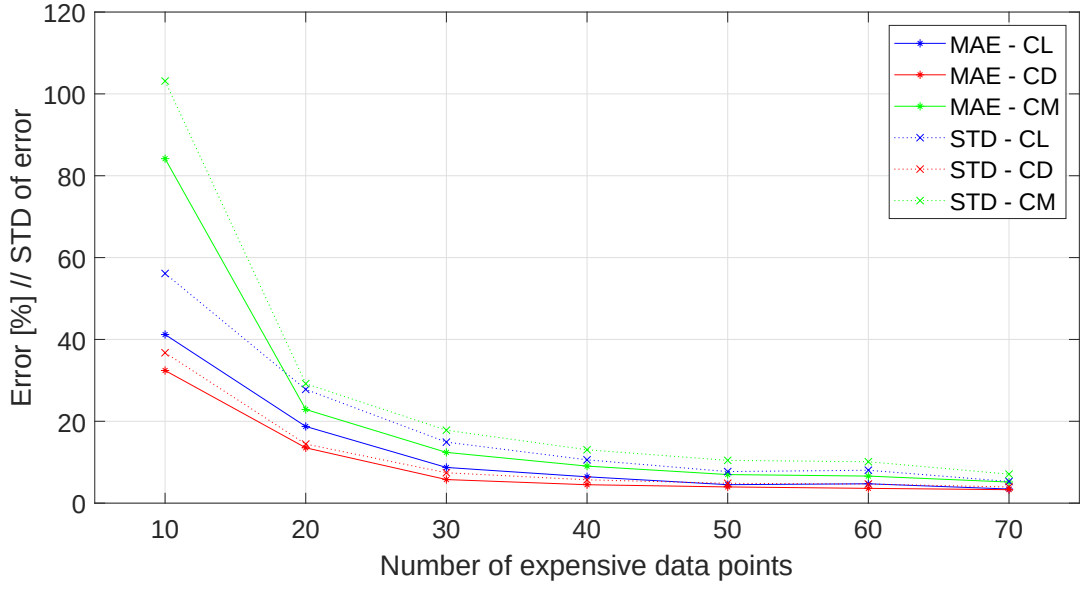
	Subsonic			Transonic			Supersonic		
Points	σ_{C_L}	σ_{C_D}	σ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}
10	13.10	39.64	37.54	13.10	21.07	38.21	56.12	36.75	103.14
20	8.58	23.36	15.46	11.22	14.93	15.99	27.79	14.43	29.19
30	6.67	11.20	8.40	6.54	8.90	9.42	14.91	7.42	17.82
40	6.48	10.26	7.11	4.61	6.19	5.76	10.58	5.70	13.04
50	4.88	10.48	5.27	3.61	4.67	5.08	7.71	4.85	10.45
60	4.25	7.46	4.37	2.88	3.32	3.79	8.04	4.67	10.12
70	4.02	7.42	3.55	3.00	3.06	3.50	5.30	3.94	7.08



(a) Subsonic speed regime



(b) Transonic speed regime



(c) Supersonic speed regime

Figure 6.28: Variation of relative error (in percent) and its standard deviation with increasing amount of expensive data points of the longitudinal coefficients divided in the three analysed speed regimes.

7

Conclusions, recommendations, outlook

This graduation project presented a refreshed approach to aerospace vehicle design, particularly to HTHL SSTO spaceplanes. In this project, a framework for aerodynamic design analysis was created to assist in preliminary design of such vehicles. The built-in geometry module employs a flexible and intuitive parametrization of the outer mould line that allows for modeling of different, spaceplane-characteristic configurations. Small, as well as bigger adjustments to a baseline geometry can be easily introduced, and it can be automatically analysed in terms of its aerodynamics with a low-fidelity aerodynamic code - ZONAIR. This low-fidelity data is interpolated using the kriging technique to create an aerodynamic database. The geometry can also be exported as a STEP-file for further analysis in other tools, e.g., using high-fidelity CFD. The results of the high-fidelity analysis can be fed back to the framework to improve the accuracy of the low-fidelity aerodynamic database through a multi-fidelity interpolation scheme called co-kriging. This final chapter of this report presents and recaps all conclusions of the presented research in **Section 7.1**. Then, **Section 7.2** gives recommendations for potential future work. Lastly, **Section 7.3**, makes digressions about potential outlook for the created framework.

7.1. Conclusions

The main goal of this research project was the creation of a framework to assist in preliminary design of spaceplanes. In the current state, the framework consists of a geometry module and an aerodynamic analysis module. The main aim of the aerodynamic analysis module is the building of aerodynamic databases with accuracy comparable to high-fidelity CFD calculations, but lower computational overhead to make it applicable for preliminary analysis where the geometry of the design can still change significantly.

During the research it was found that the multi-fidelity interpolation technique co-kriging can be used to generate accurate surrogate models. The study of the variation of accuracy of the produced models consisted of generating thirty models for each selected number of expensive data points: 10, 20, 30, 40, 50, 60, and 70. Each model used four fixed delimiting points, and the rest of the points were randomly selected from the pool of the remaining points. This randomization of the selection allowed for simultaneous analysis of the influence of the distribution of points in the domain. The following metrics were used in this analysis:

- relative error between the predicted value and the true value at each validation point;
- mean absolute error of each coefficient in the subsonic, transonic, and supersonic speed

ranges;

- standard deviation of the errors of each coefficient in the subsonic, transonic, and supersonic speed ranges;
- qualitative analysis of the produced surrogate surface of each coefficient.

In the study it was found that using just ten points did not allow to increase the accuracy of the prediction above pure kriging, whatever their distribution. However, just twenty expensive points can already produce models of higher accuracy than those generated with kriging, provided that the distribution of these points is adequate. Obviously, in a real world application this distribution would not be random, therefore it can be stated that adding twenty well-distributed points is enough to improve the predictions of the surrogate models. Furthermore, with thirty expensive points the expected error can drop well below 10% with low standard deviations, which can be generally considered acceptable.

Moreover, it was found that the distribution of the points plays a bigger role than just the raw number of points. On multiple occasions models generated with a lower number of points but a good distribution were observed to perform better than *poorly distributed* but *more populated* models. A well thought out distribution can lead to bigger improvements than just random addition of extra data points. The best models were usually characterised by high heterogeneity of the Mach-α coordinates of the points. Moreover, the density of expensive data points should be high around the locations where big gradients are expected, and lower where the values do not change rapidly. Lastly, the points should generally be spread out as much as possible across the domain, and balance should be kept between points at the edges of the domain and those in the middle of it.

On a high performance cluster, which was employed during the study to conduct the CFD simulations, the time necessary to generate one data point usually varied between 6 and 12 hours, depending on the combination of speed and angle. It shows that limiting the quantity of expensive data points by even a small amount leads to big decrease in time necessary to evaluate a design. Therefore, an application of co-kriging to create surrogate models of the aerodynamic coefficients can translate to big time saving and, thus, more iterations during the design phase allowing for better exploration of the design space in shorter time.

7.2. Recommendations

Even though the project realized all of the intended goals, the work is certainly not complete. With the groundwork laid out, there are a couple of ways the project could be driven even farther, especially with the knowledge already gained by the development of the current framework.

Debugging

One of the major improvements to the existing application would come from a thorough debugging of the entire Python code. Due to tight time constraints of the project, the time foreseen for this activity was limited to ensure basic functionality for a selected few cases. Therefore, there is a high probability that there are undetected issues in the code that can prevent smooth functioning of the framework in some cases. Furthermore, some bugs were indeed detected in the code, however, due to their negligible impact on the presented work, they were at the time ignored because of higher priority tasks that still needed to be performed. Furthermore, if more time is dedicated to improve the code, the framework might also experience an increase in its speed, since the main goal for the current project was to create a functional code, and computational time of the code was not of the highest priority.

Framework development

In its current state, the framework consists of only two main modules - the geometry module, and the aerodynamic analysis module. In order to tackle a problem as complex as spaceplane design, just these two would surely not suffice. The extremely convoluted nature of vehicle design, as discussed in **Chapter 2**, makes it impossible to optimise a design without considering more disciplines. Particularly a vehicle as complex as a spaceplane, which experiences some of the most extreme conditions of any man-made vehicle, and whose disciplines are so intertwined (vide **Section 1.1.3**). These disciplines and their interdependence requires an in-depth analysis at an as-early-as-possible stage of the design process.

This framework was being created with this potential development already in mind, making an integration of different modules responsible for analyses of other disciplines possible, and, hopefully, not too troublesome. One of the reasons why *ZONAIR* was chosen for the analysis of the aerodynamics is because of its capabilities going beyond aerodynamic calculations. With this software alone it would also be possible to analyse aeroelasticity and aerothermodynamics around the designed vehicle with only minor adjustments to the existing framework. The interpolation of the results for a desired Mach-α range (or using angle of sideslip (β) instead of angle of attack) would also be possible with the same data fusion algorithm already implemented.

A structural analysis module would require a bit more work but it would also greatly benefit the design process. In the current state, only the outer mould line is modeled. For the analysis of the spaceplane's structure, the internal components would need to be added.

With the information coming from the aerodynamic performance of the spaceplane, the effectiveness of its control surfaces, the flight loads (both aerodynamic and aerothermodynamic), and with additional engine simulator, another module could also be added to the workflow - a trajectory optimisation module. This module would find the best trajectory for a given design parameter, e.g., lowest thermal loads, lowest empty mass, etc.

Interpolation

During the presented study, the distribution of the expensive data points was mostly random, thus the accuracy of the produced surrogate models was not controlled, but rather a result of chance. It was found that the accuracy of the co-kriging interpolation scheme is highly dependent on the distribution of the points across the domain. Obviously, during a real life application, this distribution would not be random, but in one way or another, controlled. Sampling schemes should be researched and applied in order to limit the amount of points necessary to obtain a certain required level of accuracy. Techniques such as latin hypercube sampling or orthogonal sampling could be considered as potential candidates.

7.3. Outlook

The presented framework in its current state has been validated and proven to provide reliable results. Hence, the question is, where and how could it be applied? On one hand, the answer might seem obvious, since it can be found in the title of this thesis - *[...] framework for preliminary spaceplane design*. However, this was nothing but a mere driver, a leading idea during the development process. First of all, yes, this framework can surely be applied in the preliminary design of spaceplanes. But nothing stops it from being used even in, e.g., late conceptual design stage. If no expensive data points are fed to the algorithm, the low-fidelity analysis module was shown to provide results with acceptable accuracy to give a quick overview of a proposed design. Furthermore, the level of fidelity of the aerodynamic analysis can be easily manipulated by choosing more complex turbulence models than the one-equation Spalart-Allmaras used in this project (such as Large-eddy simulation or other closure models briefly discussed in **Paragraph 3.2.3.3.1**), or simpler CFD methods such as Thin-layer

Navier-Stokes, Boundary-layer equations, or Euler equations. This can make the framework relevant and applicable to both earlier and later stages of the design.

Furthermore, the application of this framework does not need to be limited to just spaceplanes. The geometry module is flexible enough to allow for modeling of other types of aircraft and spacecraft without introducing any changes to the parametrization. A commercial-style aircraft, a fighter jet, or even a missile could also be created and analysed within the proposed workflow, with the flight conditions adequate for each style of vehicle. The framework was validated for a specific range of Mach numbers and angles of attack, and solely in the longitudinal plane of motion. However, its application can easily be extended to other flight conditions, and for full 6DOF motion.

References

[1] M. Conner. *Orbital Sciences Corporation X-34*. https://www.nasa.gov/centers/armstrong/history/experimenatl_aircraft/X-34.html. Consulted on 18 August 2022.

[2] M. A. Bentley. *Spaceplanes: From Airport to Spaceport*. New York, NY: Springer New York, 2009. ISBN: 978-0-387-76509-9 978-0-387-76510-5. DOI: 10.1007/978-0-387-76510-5.

[3] *World War I: Aviation and Aircraft of WWI*. https://www.ducksters.com/history/world_war_i/aviation_and_aircraft_of_ww1.php. Consulted on 10 December 2021.

[4] *Wright Airplanes*. https://www.wright-brothers.org/Information_Desk/Just_the_Facts/Airplanes/Wright_Airplanes.htm. Consulted on 21 December 2021.

[5] W. Leitch. *God's Glory in the Heavens*. A. Strahan, 1867.

[6] J. T. Andrews. *Red Cosmos: K. E. Tsiolkovskii, Grandfather of Soviet Rocketry*. College Station: Texas A&M University Press, July 2004. ISBN: 978-1-60344-168-1.

[7] H.-N. Teodorescu. "Hermann Oberth and His Professional Geography in the European Context of the XX Century". In: *Technical University of Iaşi, "Al. I. Cuza" University of Iaşi, and the Institute of Theoretical Informatics of the Romanian Academy* (2004).

[8] J. D. Hunley. "The Enigma of Robert H. Goddard". In: *Technology and Culture* vol. 36 (Apr. 1995). ISSN: 0040-165X. DOI: 10.2307/3106375.

[9] E. Sanger and J. Bredt. *A Rocket Drive For Long Range Bombers*. Deutsche Luftfahrtforschung, Aug. 1944.

[10] C. Reuter. *The V2 and the German, Russian and American Rocket Program*. S.R. Research & Publishing; 2nd edition, 2000. ISBN: 978-1-894643-05-4.

[11] R. Ford. *Germany's Secret Weapons in World War II*. Osceola, Wisc: MBI Pub. Co, 2000. ISBN: 978-0-7603-0847-9.

[12] W. Späte and R. Bateson. "Messerchmitt Me 163 Komet". In: *Profile Publications* (1971).

[13] A. J. Pelletier. *Bell Aircraft Since 1935*. Annapolis, Md: Naval Institute Press, 1992. ISBN: 978-1-55750-056-4.

[14] S. P. Butchart et al. *Toward Mach 2: The Douglas D-558 Program*. The NASA History Series, 1999.

[15] Y. Gibbs. *NASA Dryden Fact Sheets - X-15 Hypersonic Research Program*. http://www.nasa.gov/centers/armstrong/news/FactSheets/FS-052-DFRC.html. Consulted on 10 December 2021.

[16] L. Shiner. *X-15 Walkaround*. https://www.airspacemag.com/history-of-flight/x-15-walkaround-22512890/. Consulted on 21 December 2021.

[17] Y. Gibbs. *NASA Dryden Fact Sheet - Lifting Bodies*. http://www.nasa.gov/centers/armstrong/news/FactSheets/FS-011-DFRC.html. Consulted on 10 December 2021.

[18] T. A. Heppenheimer. *The Space Shuttle Decision*. NASA History Series, 1999.

[19] B. Hendrickx and B. Vis. *Energiya-Buran: The Soviet Space Shuttle*. Springer Science, Jan. 2007.

[20] *X-37B OTV 1, 2, 3, 4, 5, 6*. Gunter's Space Page. https://space.skyrocket.de/doc_sdat/x-37.htm. Consulted on 10 December 2021.

[21] D. Sivolella. *The Space Shuttle Program*. Cham: Springer International Publishing, 2017. ISBN: 978-3-319-54944-6 978-3-319-54946-0. DOI: 10.1007/978-3-319-54946-0.

[22] *The Delta Clipper Experimental Archive*. https://www.hq.nasa.gov/pao/History/x-33/dc-xa.htm. Consulted on 10 December 2021.

[23] SpaceX. *Falcon Heavy: The World's Most Powerful Rocket*. https://www.spacex.com/vehicles/falcon-heavy/. Consulted on 21 December 2021.

[24] SpaceX. *Falcon 9: Firt Orbital Class Rocket Capable of Reflight*. https://www.spacex.com/vehicles/falcon-9/. Consulted on 21 December 2021.

[25] SpaceX. *Flight Test: Starship SN15*. https://www.spacex.com/vehicles/starship/. Consulted on 21 December 2021.

[26] *Space Launch Report 2020 Launch Stats*. https://www.spacelaunchreport.com/log2021.html. Consulted on 10 December 2021.

[27] E. H. Hirschel. *Basics of Aerothermodynamics*. Cham: Springer International Publishing, 2015. ISBN: 978-3-319-14372-9 978-3-319-14373-6. DOI: 10.1007/978-3-319-14373-6.

[28] C. Weiland. *Aerodynamic Data of Space Vehicles*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. ISBN: 978-3-642-54167-4 978-3-642-54168-1. DOI: 10.1007/978-3-642-54168-1.

[29] *X-33 Advanced Technology Demonstrator Photo Collection Large Contact Sheet*. https://www.dfrc.nasa.gov/Gallery/Photo/X-33/Large/index.html. Consulted on 10 December 2021.

[30] G. J. Brauckmann. "X-34 Vehicle Aerodynamic Characteristics". In: *Journal of Spacecraft and Rockets* vol. 36.2 (Mar. 1999), pp. 229–239. ISSN: 0022-4650, 1533-6794. DOI: 10.2514/2.3453. URL: https://arc.aiaa.org/doi/10.2514/2.3453.

[31] H. D. Fuhrmann, J. Hildebrand, and T. Lalicata. "Aerothermodynamic Overview, X-34". In: *Journal of Spacecraft and Rockets* vol. 36.2 (Mar. 1999), pp. 153–159. ISSN: 0022-4650, 1533-6794. DOI: 10.2514/2.3437. URL: https://arc.aiaa.org/doi/10.2514/2.3437.

[32] B. I. Larrimer. *Promise denied: NASA's X-34 and the Quest for Cheap Reusable Access to Space*. Ed. by U. States. NASA aeronautics book series. Washington, DC: National Aeronautics and Space Administration, 2020. ISBN: 978-1-62683-049-3 978-1-62683-050-9.

[33] F. Gand et al. "Flow dynamics past a simplified wing body junction". In: *Physics of Fluids* vol. 22.11 (Nov. 2010). DOI: 10.1063/1.3500697.

[34] J. Coombs et al. "Assessment of turbulence models for a wing-in-junction flow". In: *18th Australasian Fluid Mechanics Conference* (Dec. 2012).

[35] H. Babinsky and J. K. Harvey. *Shock wave-boundary layer interactions*. Cambridge University Press, 2011. ISBN: 978-0-521-84852-7.

[36] G. L. Morini and D. Li. *Encyclopedia of Microfluidics and Nanofluidics*. Springer US, 2008. ISBN: 978-0-387-48998-8.

[37] C. A. Lind. "Effect of Geometry on the Unsteady Type-IV Shock Interaction". In: *Journal of Aircraft* vol. 34.1 (Jan. 1997).

[38] J. D. Anderson. *Hypersonic and High Temperature Gas Dynamics*. 2nd. American Institute of Aeronautics and Astronautics, 2006. ISBN: 978-1-56347-780-5.

[39] J. Rom. *High Angle of Attack Aerodynamics*. Springer-Verlag, 1992. ISBN: 978-1-4612-2824-0.

[40] G. La Rocca. *Knowledge Based Engineering Techniques to Support Aircraft Design and Optimization [PhD Thesis]*. PhD thesis. TU Delft. Apr. 2011.

[41] J. D. Anderson. *Aircraft Performance and Design*. Tata McGraw-Hill. Boston: WCB/McGraw-Hill, 1999. ISBN: 978-0-07-001971-3.

[42] D. P. Raymer. *Aircraft Design: A Conceptual Approach*. 6th. Reston, VA: AIAA education series, 2018. ISBN: 978-1-62410-490-9.

[43] *Space Shuttle design process*. https://en.wikipedia.org/w/index.php?title=Space_Shuttle_design_process&oldid=1035528784. Consulted on 13 December 2021.

[44] D. E. Calkins, N. Egging, and C. Scholz. "Knowledge-Based Engineering (KBE) Design Methodology at the Undergraduate and Graduate Levels". In: *International Journal of Engineering Education* vol. 26 (Jan. 2000).

[45] H.-H. Altfeld. *Commercial Aircraft Projects: Managing the Development of Highly Complex Products*. Farnham, England; Burlington, VT: Ashgate Pub, 2010. ISBN: 978-0-7546-7753-6 978-0-7546-9494-6.

[46] M. E. J. Newman. "Power laws, Pareto distributions and Zipf's law". In: *Contemporary Physics* vol. 46, no. 5.5 (Sept. 2005). ISSN: 0010-7514, 1366-5812. DOI: 10.1080/00107510500052444. URL: http://arxiv.org/abs/cond-mat/0412004.

[47] V. Pareto. *Cours d'Économie Politique*. Librairie Droz, 1964. ISBN: 978-2-600-04014-3. DOI: 10.3917/droz.paret.1964.01. URL: https://www.cairn.info/cours-d-economie-politique-tomes-1-et-2--9782600040143.htm.

[48] S. Gudmundsson. *General Aviation Aircraft Design: Applied Methods and Procedures*. Butterworth-Heinemann, Elsevier, 2021. ISBN: 978-0-12-822647-6. URL: https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=2464466.

[49] T. Gwosch et al. *Frontloading in Aircraft Development Process by Integration of a new Validation Method*. KIT Scientific Working Papers, 2020. DOI: 10.5445/IR/1000117783. URL: https://publikationen.bibliothek.kit.edu/1000117783.

[50] D. Schrage et al. "White Paper on Current State of the Art". In: *AIAA Technical Committee on Multidisciplinary Design Optimization* (1991).

[51] A. Raju Kulkarni et al. "A Knowledge Based Engineering Tool To Support Front-Loading and Multidisciplinary Design Optimization of the Fin-Rudder Interface". In: *Aerospace Europe 6th CEAS Conference* 232 (Oct. 2017).

[52] P. E. Bézier, W. R. Hawthorne, and G. R. Edwards. "Example of an Existing System in the Motor Industry: the UNISURF System". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* vol. 321, no. 1545.1545 (Feb. 1971). DOI: 10.1098/rspa.1971.0027. URL: https://royalsocietypublishing.org/doi/10.1098/rspa.1971.0027.

[53] L. A. Schmit. "Structural Design by Systematic Synthesis". In: *2nd Conference on Electronic Computation, New York, NY* (1960).

[54] L. A. Schmit and W. A. Thornton. " Synthesis of an Airfoil at Supersonic Mach Number". In: *Technical Report CR 144, NASA* (Jan. 1965).

[55] R. T. Haftka. “Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements”. In: *Technical Report TN D-7264, NASA Langley Research Center, Hampton, VA* (July 1973).

[56] R. T. Haftka. “Optimization of Flexible Wing Structures Subject to Strength and Induced Drag Constraints”. In: *AIAA Journal* vol. 15.8 (Aug. 1977). DOI: 10.2514/3.7400.

[57] J. Giesing and J.-F. Barthelemy. “A Summary of Industry MDO Applications and Needs”. In: *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (Sept. 1998). DOI: 10.2514/6.1998-4737. URL: https://arc.aiaa.org/doi/10.2514/6.1998-4737.

[58] O. de Weck et al. “State-of-the-Art and Future Trends in Multidisciplinary Design Optimization”. In: *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (Apr. 2007). DOI: 10.2514/6.2007-1905. URL: https://arc.aiaa.org/doi/10.2514/6.2007-1905.

[59] M. F. M. Hoogreef et al. “A Multidisciplinary Design Optimization Advisory System for Aircraft Design”. In: *5th CEAS Air & Space Conference* 41 (Sept. 2015).

[60] J. Sobieszczanski-Sobieski et al. *Multidisciplinary Design Optimization Supported by Knowledge Based Engineering*. Chichester, UK: John Wiley & Sons, Ltd, 2015. ISBN: 978-1-118-89707-2 978-1-118-49212-3. DOI: 10.1002/9781118897072. URL: https://onlinelibrary.wiley.com/doi/10.1002/9781118897072.fmatter.

[61] G. La Rocca. “Knowledge Based Engineering: Between AI and CAD. Review of a Language Based Technology to Support Engineering Design”. In: *Advanced Engineering Informatics* vol. 26 (Apr. 2012). ISSN: 14740346. DOI: 10.1016/j.aei.2012.02.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S1474034612000092.

[62] N. R. Milton. *Knowledge Technologies*. Vol. vol. 3. Polimetrica, 2008. ISBN: 978-88-7699-100-4.

[63] C. Chapman and M. Pinfold. “Design Engineering - A Need to Rethink the Solution Using Knowledge Based Engineering”. In: *Knowledge Based Systems* vol. 12 (Oct. 1999). DOI: 10.1016/S0950-7051(99)00013-1.

[64] ParaPy. *ParaPy*. https://www.parapy.nl/. Consulted on 17 December 2021.

[65] TechnoSoft, Inc. *Adaptive Modeling Language*. http://www.technosoft.com/application-software/adaptive-modeling-language/. Consulted on 17 December 2021.

[66] Eventive Engineering, Inc. *Enventive*. https://www.enventive.com/. Consulted on 17 December 2021.

[67] *Gendl*. https://gendl.org/. Consulted on 17 December 2021.

[68] Genworks International. *Genworks GDL*. https://genworks.com. Consulted on 17 December 2021.

[69] PACE Aerospace Engineering and Information Technology GmbH. *Pacelab Suite*. https://pace.txtgroup.com. Consulted on 17 December 2021.

[70] Tacit Connexions. *PCPACK6*. http://www.tacitconnexions.com/index-11.html. Consulted on 17 December 2021.

[71] A. Rizzi and J. Oppelstrup. *Aircraft Aerodynamic Design with Computational Software*. 1st. Cambridge University Press, 2021. ISBN: 978-1-139-09467-2 978-1-107-01948-5. DOI: 10.1017/9781139094672. URL: https://www.cambridge.org/core/product/identifier/9781139094672/type/book.

[72] Federal Aviation Administration. *Pilot’s Handbook of Aeronautical Knowledge*. U.S. Department of Transportation, 2016. URL: https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/phak/media/pilot_handbook.pdf.

[73] J. Roskam. *Airplane Design Parts I-VIII*. DAR Corpotation, 1985.

[74] D. Raymer. *Dan Raymer's Aircraft Design & RDS Website*. http://www.aircraftdesign.com/. Consulted on 10 January 2022.

[75] DAR Corporation. *Advanced Aircraft Analysis*. https://www.darcorp.com/advanced-aircraft-analysis-software/. Consulted on 10 January 2022.

[76] D. E. Hoak et al. "USAF Stability and Control DATCOM". In: *Flight Control Division, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base* (Apr. 1978).

[77] IHS Markit. *ESDU*. https://ihsmarkit.com/products/esdu.html. Consulted on 10 January 2022.

[78] P. W. Sacher et al. "Special Course on Engineering Methods in Aerodynamic Analysis and Design of Aircraft". In: *AGARD report #783* (Jan. 1992). presented as an AGARD Special Course at the Middle East Technical University, Ankara, Turkey, 6th -10th May 1991, at the von Kármán Institute for Fluid Dynamics, Rhode-St-Genèse, Belgium 13th -17th May 1991 and at the Universidad Politecnica de Madrid, Spain 20th -24th May 1991.

[79] J. D. Anderson. *Fundamentals of Aerodynamics*. Sixth. McGraw-Hill series in aeronautical and aerospace engineering. New York, NY: McGraw Hill Education, 2017. ISBN: 978-1-259-12991-9.

[80] F. W. Lanchester. *Aerodynamics. Constituting the first volume of a complete work on aerial flight*. London, A. Constable & Co., Ltd., 1907. URL: http://archive.org/details/aerodynamicscons00lanc.

[81] L. Prandtl. *Tragflügeltheorie*. Königliche Gesellschaft der Wissenschaften zu Göttingen, 1918.

[82] P. Boschetti et al. "Stability and Performance of a Light Unmanned Airplane in Ground Effect". In: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (Apr. 2010). DOI: 10.2514/6.2010-293. URL: https://arc.aiaa.org/doi/10.2514/6.2010-293.

[83] L. Erickson. *Panel methods: An introduction*. Ames Research Center, Jan. 1991.

[84] J. Blazek. *Computational Fluid Dynamics - Principles and Applications*. Third. Butterworth-Heinemann/Elsevier, 2007.

[85] J. H. Ferziger, M. Perić, and R. L. Street. *Computational Methods for Fluid Dynamics*. Fourth. Springer Nature Switzerland AG, 2020. URL: https://doi.org/10.1007/978-3-319-99693-6.

[86] C. Hirsch. *Numerical Computation of Internal and External Flows: Fundamentals of Computational Fluid Dynamics*. Second. Oxford ; Burlington, MA: Elsevier/Butterworth-Heinemann, 2007. ISBN: 978-0-7506-6594-0.

[87] C. Speziale. "Turbulence modeling for time-dependent RANS and VLES : a review". In: *AIAA Journal* vol. 36(2) (Feb. 1998). ISSN: 0001-1452. DOI: 10.2514/2.7499.

[88] D. C. Wilcox. *Turbulence Modeling for CFD*. Third. DCW Industries, Inc, 2006.

[89] F. Palacios et al. "Stanford University Unstructured (SU2): An Open-Source Integrated Computational Environment for Multi-Physics Simulation and Design". In: *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (Jan. 2013). DOI: 10.2514/6.2013-287. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2013-287.

[90] M. A. Leschziner. *Reynolds-Averaged Navier-Stokes Methods*. Encyclopedia of Aerospace Engineering, John Wiley & Sons, Ltd., 2010.

[91] P. Spalart and S. Allmaras. “A One-Equation Turbulence Model for Aerodynamic Flows”. In: *30th Aerospace Sciences Meeting and Exhibit* (Jan. 1992). DOI: 10.2514/6.1992-439. URL: https://arc.aiaa.org/doi/10.2514/6.1992-439.

[92] B. E. Launder and S. D. B. “The Numerical Computation of Turbulent Flows”. In: *Computer Methods in Applied Mechanics and Engineering* vol. 3 (Mar. 1974).

[93] D. C. Wilcox. “Reassessment of the scale-determining equation for advanced turbulence models”. In: *AIAA Journal* vol. 26.11 (Nov. 1988).

[94] ManchesterCFD. *All there is to know about different mesh types in CFD!* https://www.manchestercfd.co.uk//post/all-there-is-to-know-about-different-mesh-types-in-cfd. Consulted on January 14 2022.

[95] A. I. J. Forrester, A. Sóbester, and A. J. Keane. *Engineering Design via Surrogate Modelling*. John Wiley & Sons, Ltd., 2008.

[96] J. Sacks et al. “Design and Analysis of Computer Experiments (with discussion)”. In: *Statistical Science* vol. 4 (Nov. 1989), pp. 409–435.

[97] D. R. Jones. “A Taxonomy of Global Optimization Methods Based on Response Surfaces”. In: *Journal of Global Optimization* (Dec. 2001). DOI: 10.1023/A:1012771025575. URL: https://www.researchgate.net/publication/216300861_A_Taxonomy_of_Global_Optimization_Methods_Based_on_Response_Surfaces.

[98] H. Theil. *Principles of Econometrics*. Wiley/Hamilton Publication, Karreman Mathematics Research Collection, 1971. ISBN: 978-0-4718-5845-4.

[99] M. C. Kennedy and A. O’Hagan. “Predicting the output from a complex computer code when fast approximations are available”. In: *Biometrika, Oxford University Press* vol. 87.1 (Mar. 2000).

[100] A. De Marco et al. “Automatic modeling of aircraft external geometries for preliminary design workflows”. In: *Aerospace Science and Technology* vol. 98 (Mar. 2020). ISSN: 12709638. DOI: 10.1016/j.ast.2019.105667. URL: https://linkinghub.elsevier.com/retrieve/pii/S1270963819316943.

[101] N. Siddiqui, W. Asrar, and E. Sulaeman. “Literature Review: Biomimetic and Conventional Aircraft Wing Tips”. In: *International Journal of Aviation, Aeronautics, and Aerospace* vol. 4 (May 2017). DOI: 10.15394/ijaa.2017.1172.

[102] D. J. Kinney. *Aero-Thermodynamics for Conceptual Design*. 42nd AIAA Aerospace Sciences Meeting and Exhibit. NASA Ames Research Center: American Institute of Aeronautics and Astronautics, 2004.

[103] R. in Flight Company. *FlightStream*. https://researchinflight.com/index.html. Consulted on 21 July 2022.

[104] *USAF Stability and Control DATCOM*. McDonnell Douglas Corporation, Douglas Aircraft Division, for the Flight Controls Division, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, October 1960.

[105] R. Carmichael. *PANAIR: Predicting Subsonic Or Supersonic Linear Potential Flows About Arbitrary Configurations Using A Higher Order Panel Method*. http://ckw.phys.ncku.edu.tw/public/pub/Notes/Computers/Lectures/Resources/www.pdas.com/panair.htm. Consulted on 21 July 2022.

[106] D. Adamczak and A. A. Bhungalia. *Supersonic Hypersonic Arbitrary Body Program (S/HABP)*. https://tfaws.nasa.gov/TFAWS03/software_training/shabp_afrl_demo.pdf#:~:text=The%20Supersonic%2FHypersonic%20Arbitrary%20Body%20Program%20%28SHABP%29%20calculates%20the,25%2C%20and%20for%20angles%20between%200-%20180%20degrees.. Consulted on 21 July 2022.

[107] *ZAERO User's Manual, v.9.3*. ZONA Technology, Inc., 2019.

[108] *ZONAIR User's Manual, v.4.7*. ZONA Technology, Inc., 2017.

[109] P. C. Chen, D. D. Liu, and L. Tang. *ZONAIR for RLV/TPS Design and Analysis, from SHABP to ZONAIR*. https://tfaws.nasa.gov/TFAWS03/software_training/shapb_zonair_demo.pdf. Consulted on 21 July 2022.

[110] J. Katz and P. Allen. *Low-Speed Aerodynamics: From Wing Theory to Panel Methods*. International Edition. McGraw-Hill, Inc., 1991.

[111] J. L. Hess. "Panel Methods in Computational Fluid Dynamics". In: *Annual Reviews In Fluid Mechanics* vol. 22 (Jan. 1990).

[112] J. Ballmann, R. Eppler, and W. Hackbusch. *Panel Methods in Fluid Mechanics with Emphasis on Aerodynamics*. Wiesbaden: Vieweg+Teubner Verlag, 1988. ISBN: 978-3-528-08095-2 978-3-663-13997-3. DOI: 10.1007/978-3-663-13997-3. URL: http://link.springer.com/10.1007/978-3-663-13997-3.

[113] R. Whitford. *Course AE4245: Design for Air Combat*. TU Delft, Faculty of Aerospace Engineering, Jan. 1987.

[114] Ansys, Inc. *Ansys Fluent*. https://www.ansys.com/products/fluids/ansys-fluent. Consulted on 21 July 2022.

[115] T. Cebeci, ed. *Computational fluid dynamics for engineers: from panel to Navier-Stokes methods with computer programs*. Long Beach, Calif. : Berlin: Horizons Pub. Inc.,; Springer, 2005. ISBN: 978-0-9766545-0-6 978-3-540-24451-6.

[116] J. H. Ferziger, M. Perić, and R. L. Street. *Computational Methods for Fluid Dynamics*. Cham: Springer International Publishing, 2020. ISBN: 978-3-319-99691-2 978-3-319-99693-6. DOI: 10.1007/978-3-319-99693-6. URL: http://link.springer.com/10.1007/978-3-319-99693-6.

[117] C. Hirsch. *Numerical computation of internal and external flows: fundamentals of computational fluid dynamics*. 2nd. Elsevier/Butterworth-Heinemann, 2007. ISBN: 978-0-7506-6594-0.

[118] ANSYS, Inc. *ANSYS Fluent User's Guide, Release 15.0*. Nov. 2013.

[119] A. I. J. Forrester, A. Sóbester, and A. J. Keane. *Website Code November 2010*. https://www.dropbox.com/s/3somv6jd5414avj/Website%20Code%20November%2010.zip?dl=0&file_subpath=%2FWebsite+Code+November+10. Consulted on 01 August 2022.

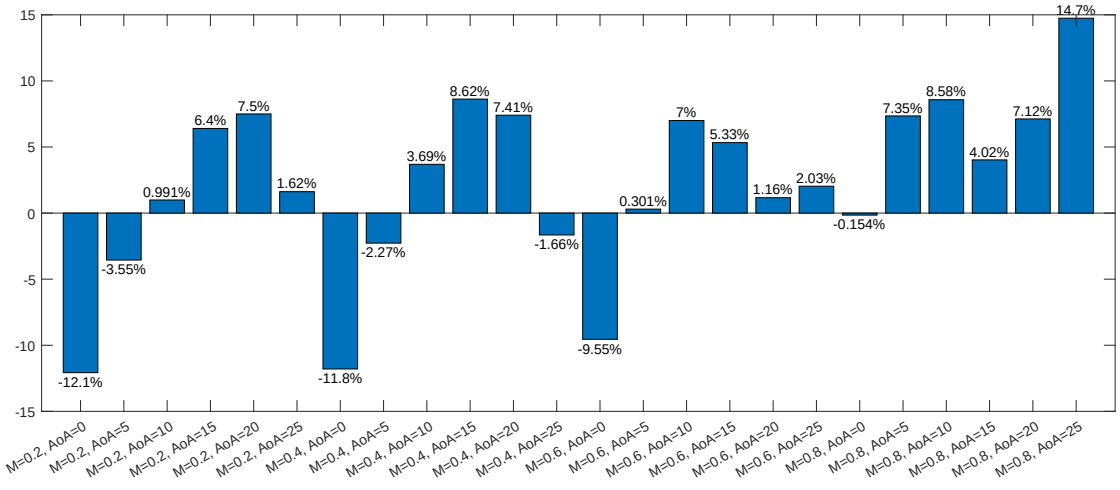
[120] B. N. Pamadi et al. "Aerodynamic Characteristics, Database Development, and Flight Simulation of the X-34 Vehicle". In: *Journal of Spacecraft and Rockets* vol. 38.3 (May 2001), pp. 334–344.

Appendix A: Ansys Fluent results

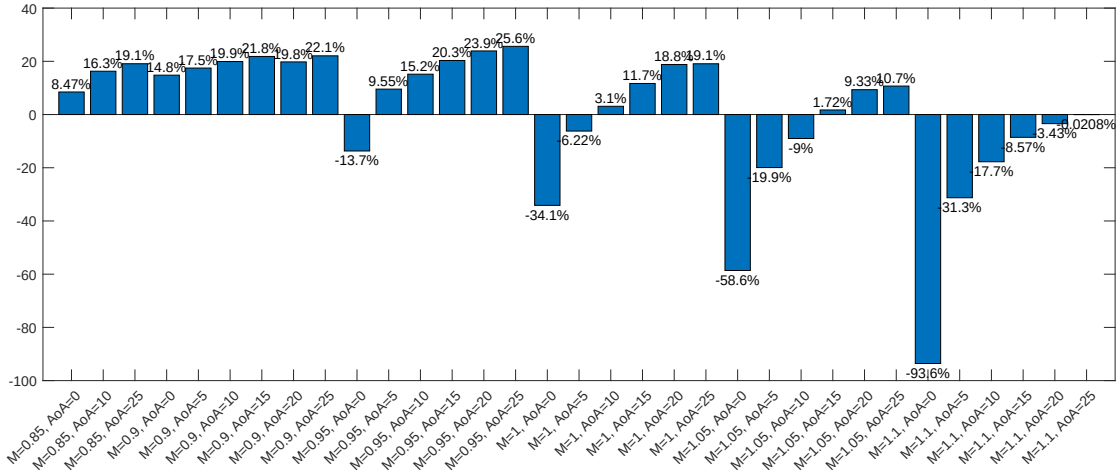
Table A.1: Numerical results of Ansys Fluent simulations.

Mach	AoA	C_L	C_D	C_M	Mach	AoA	C_L	C_D	C_M
0.2	0	0.173	0.0378	-0.1175	1	20	1.264	0.5443	-0.5589
0.2	5	0.396	0.0607	-0.1523	1	25	1.364	0.6972	-0.5206
0.2	10	0.623	0.1018	-0.1833	1.05	0	0.199	0.1146	-0.2079
0.2	15	0.863	0.1687	-0.2222	1.05	5	0.497	0.1574	-0.3078
0.2	20	1.057	0.2842	-0.2692	1.05	10	0.778	0.2422	-0.4054
0.2	25	1.141	0.4410	-0.2676	1.05	15	1.027	0.3673	-0.4832
0.4	0	0.177	0.0367	-0.1205	1.05	20	1.219	0.5234	-0.5246
0.4	5	0.407	0.0589	-0.1541	1.05	25	1.325	0.6789	-0.4854
0.4	10	0.645	0.1026	-0.1950	1.1	0	0.178	0.1113	-0.1920
0.4	15	0.882	0.1691	-0.2349	1.1	5	0.478	0.1522	-0.2975
0.4	20	1.039	0.3019	-0.2679	1.1	10	0.750	0.2336	-0.3884
0.4	25	1.066	0.4382	-0.2636	1.1	15	0.998	0.3538	-0.4636
0.6	0	0.187	0.0358	-0.1294	1.1	20	1.187	0.5055	-0.4947
0.6	5	0.426	0.0602	-0.1668	1.1	25	1.312	0.6682	-0.4467
0.6	10	0.678	0.1083	-0.2109	1.2	5	0.425	0.1388	-0.2661
0.6	15	0.850	0.2100	-0.2372	1.2	10	0.696	0.2185	-0.3591
0.6	20	0.952	0.3346	-0.2514	1.2	15	0.941	0.3372	-0.4280
0.6	25	1.058	0.4721	-0.2649	1.2	20	1.150	0.4939	-0.4625
0.8	0	0.204	0.0370	-0.1466	1.5	0	0.076	0.0997	-0.1107
0.8	5	0.468	0.0688	-0.1952	1.5	5	0.301	0.1152	-0.1815
0.8	10	0.698	0.1453	-0.2401	1.5	10	0.524	0.1776	-0.2485
0.8	15	0.837	0.2428	-0.2604	1.5	15	0.744	0.2699	-0.2976
0.8	20	0.989	0.3711	-0.2899	1.5	20	0.955	0.4137	-0.3307
0.8	25	1.171	0.5376	-0.3166	1.5	25	1.152	0.5986	-0.3387
0.85	0	0.229	0.0515	-0.1736	2	0	0.026	0.0778	-0.0746
0.85	10	0.768	0.1737	-0.2963	2	5	0.204	0.0907	-0.1156
0.85	25	1.230	0.5812	-0.3455	2	10	0.383	0.1335	-0.1521
0.9	0	0.239	0.0613	-0.1941	2	15	0.561	0.2096	-0.1760
0.9	5	0.528	0.1130	-0.2712	2	20	0.733	0.3241	-0.1927
0.9	10	0.804	0.2015	-0.3509	2	25	0.905	0.4761	-0.2121
0.9	15	1.031	0.3166	-0.3997	2.5	0	0.001	0.0681	-0.0615
0.9	20	1.132	0.4390	-0.3833	2.5	5	0.151	0.0800	-0.0866
0.9	25	1.269	0.6142	-0.3831	2.5	10	0.300	0.1133	-0.1079
0.95	0	0.208	0.0770	-0.1919	2.5	15	0.461	0.1806	-0.1259
0.95	5	0.535	0.1362	-0.3084	2.5	20	0.618	0.2788	-0.1373
0.95	10	0.833	0.2307	-0.4142	2.5	25	0.785	0.4159	-0.1615
0.95	15	1.086	0.3608	-0.4909	3	10	0.250	0.1005	-0.0853
0.95	20	1.261	0.5161	-0.5144	3	15	0.385	0.1531	-0.0914
0.95	25	1.396	0.7021	-0.5038	3	20	0.549	0.2506	-0.1122
1	0	0.206	0.1001	-0.2039	3.25	0	-0.017	0.0636	-0.0521
1	5	0.510	0.1491	-0.3061	4	0	-0.027	0.0599	-0.0472
1	10	0.805	0.2427	-0.4145	4	10	0.185	0.0864	-0.0599
1	15	1.060	0.3751	-0.5030	4	20	0.469	0.2226	-0.0889
					4	25	0.631	0.3444	-0.1131

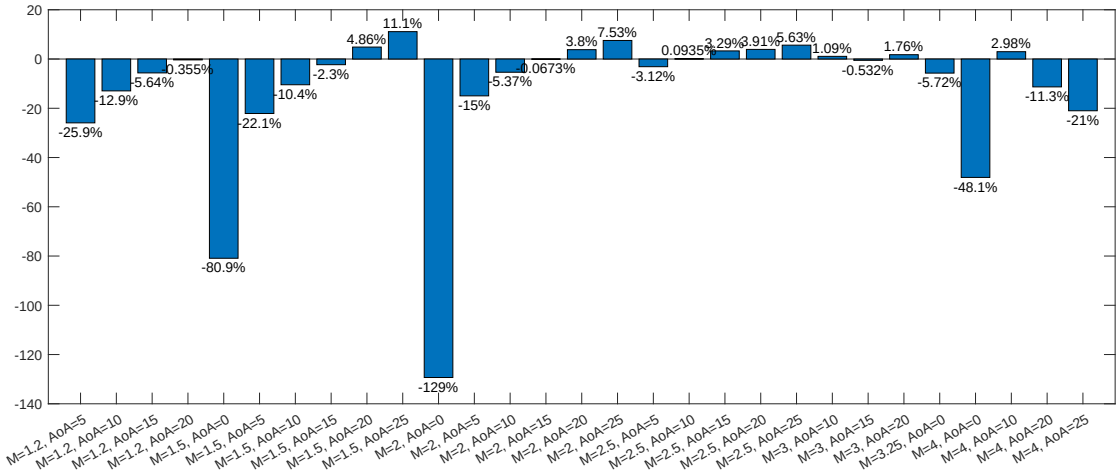
Appendix B: Kriging error plots



(a) Subsonic regime

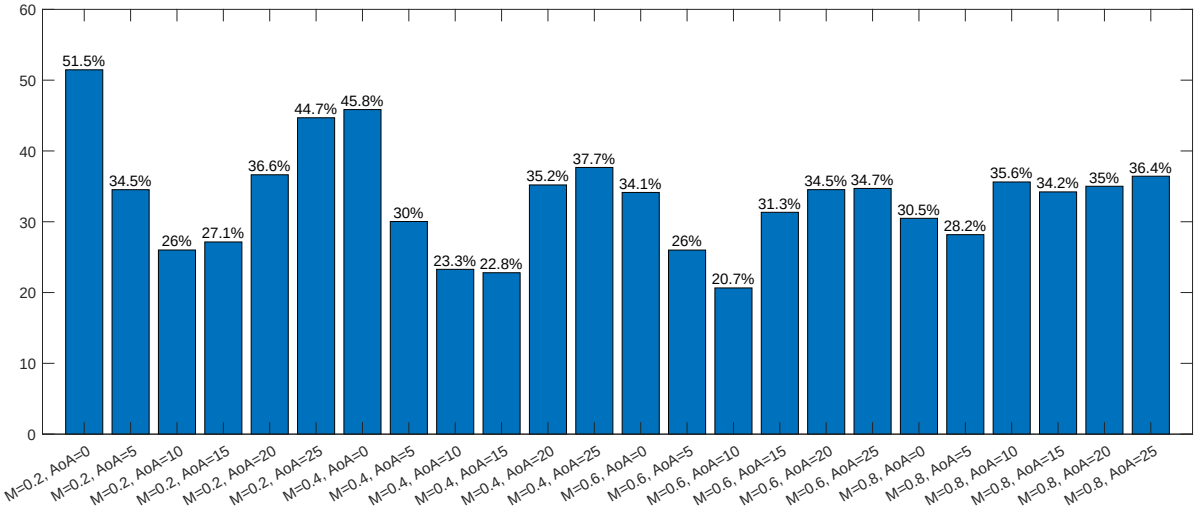


(b) Transonic regime

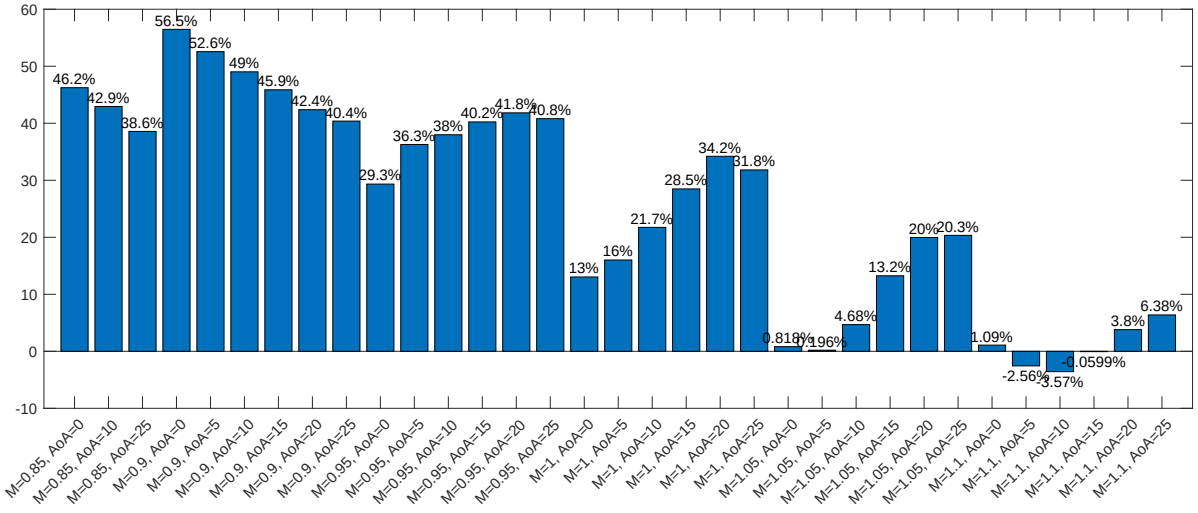


(c) Supersonic regime

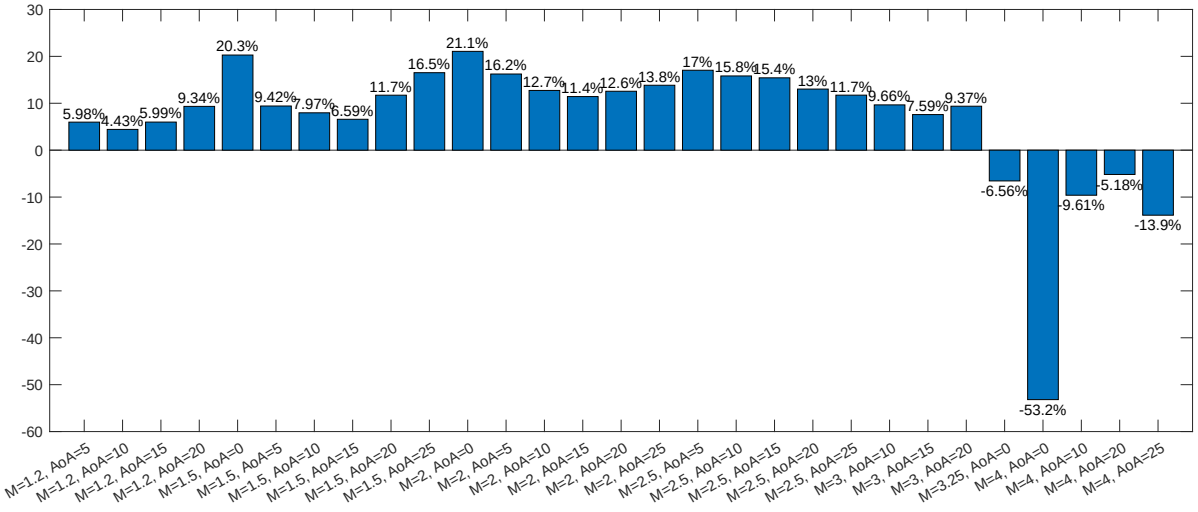
Figure B.1: Relative errors of the kriging surrogate model of lift coefficient.



(a) Subsonic regime

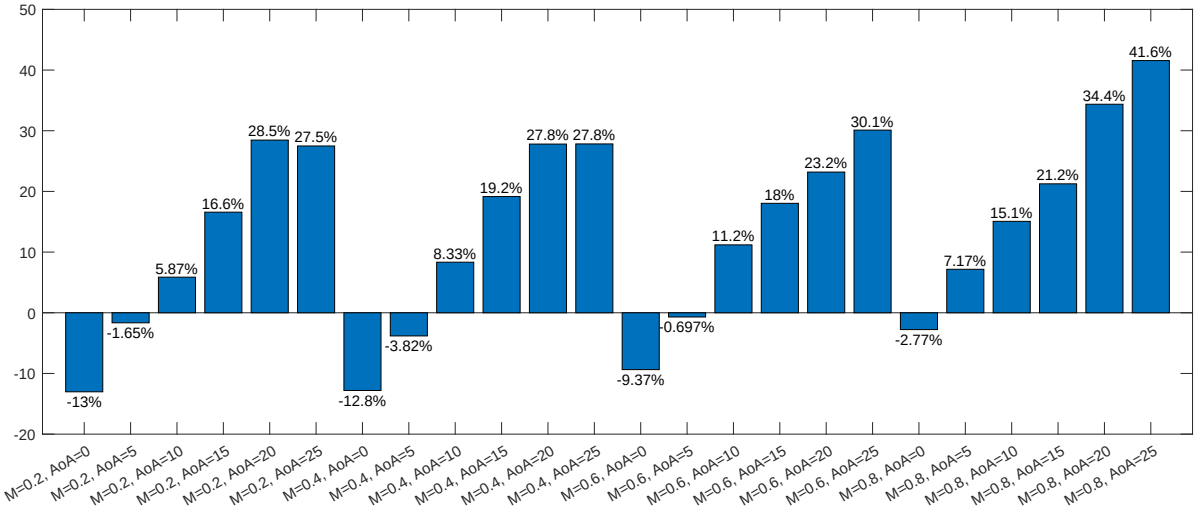


(b) Transonic regime

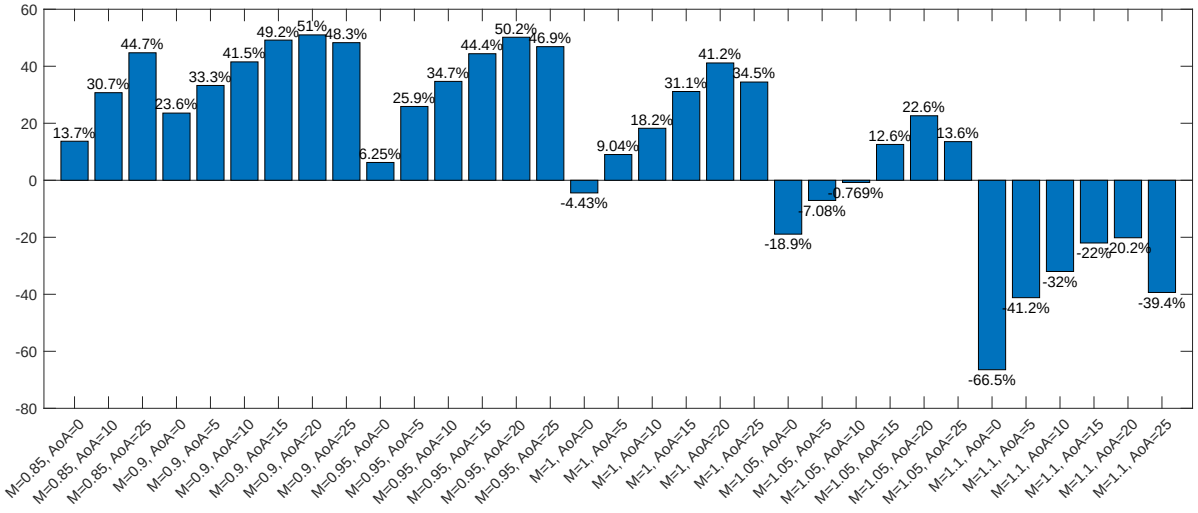


(c) Supersonic regime

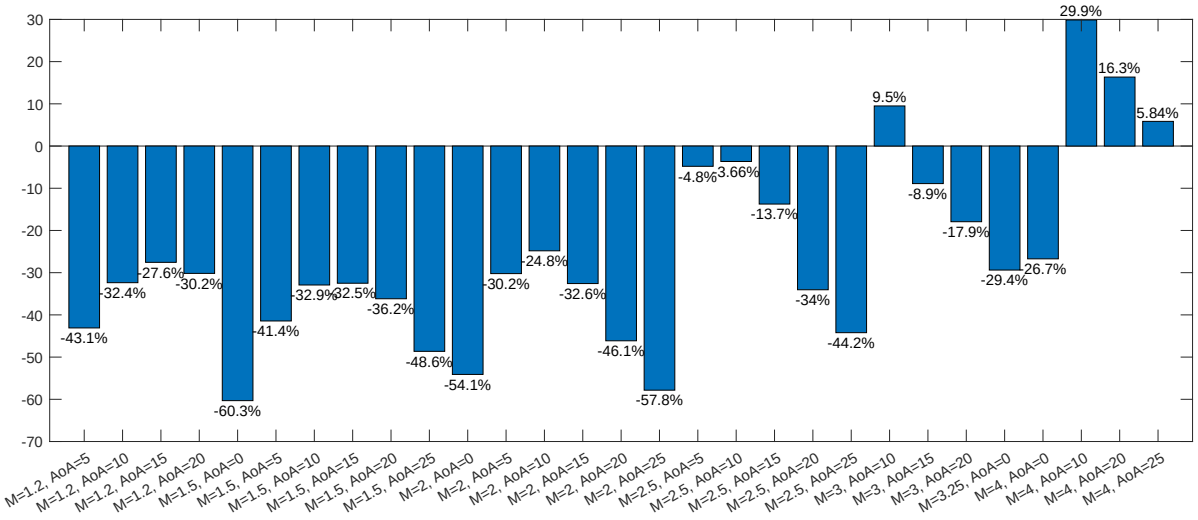
Figure B.2: Relative errors of the kriging surrogate model of drag coefficient.



(a) Subsonic regime



(b) Transonic regime



(c) Supersonic regime

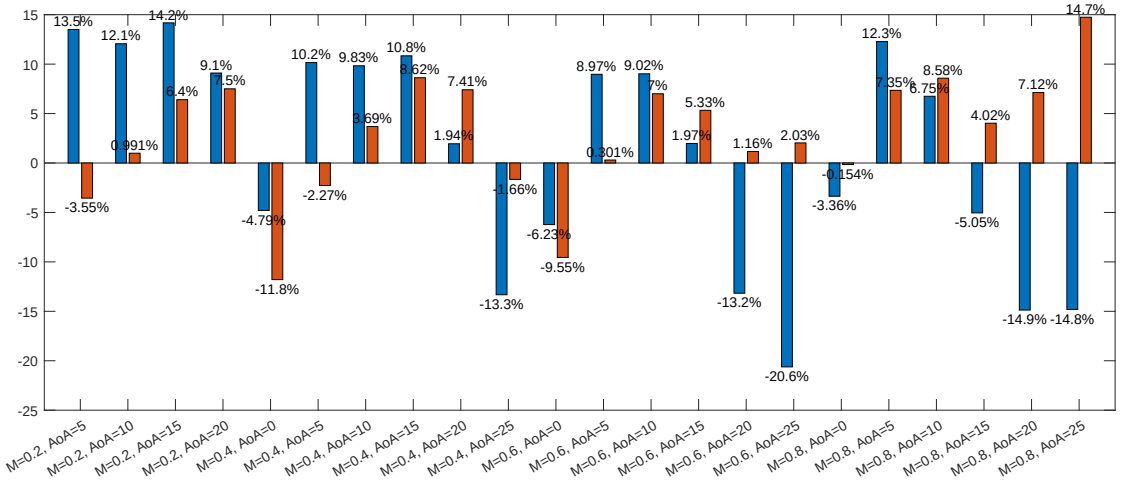
Figure B.3: Relative errors of the kriging surrogate model of pitching moment coefficient.

Appendix C: Co-kriging error plots

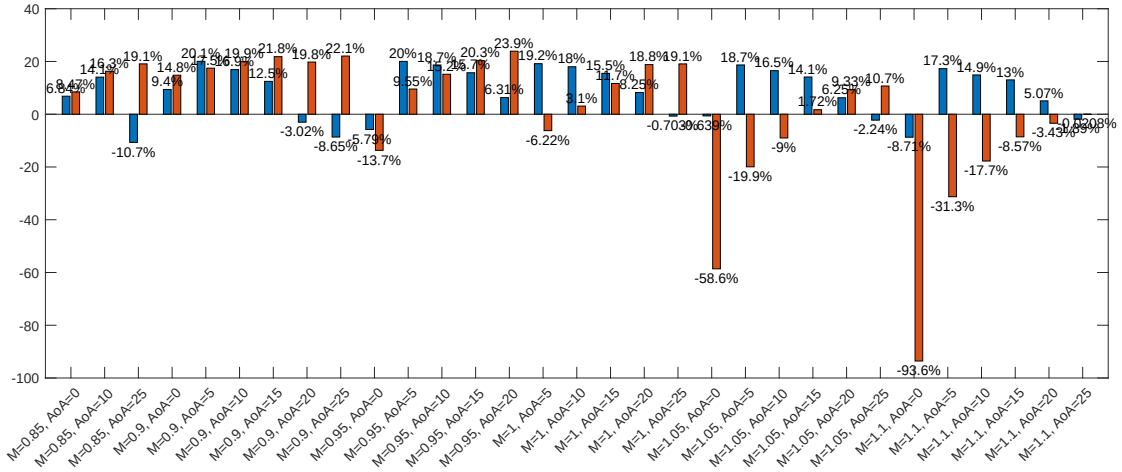
C.1. Case 1: 10 points

Table C.1: STDs of the thirty runs with 4 delimiting and 6 random points used for interpolation.

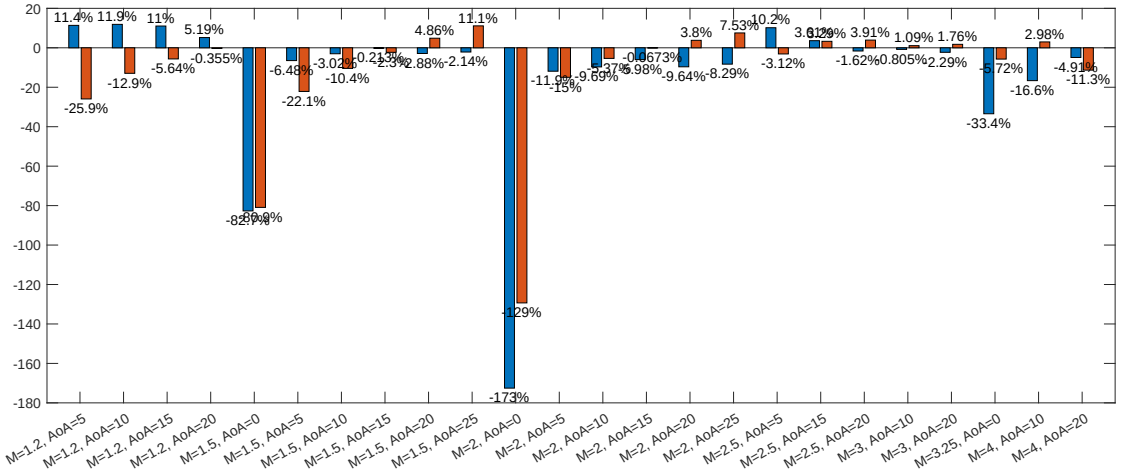
	Subsonic			Transonic			Supersonic		
Run	σ_{C_L}	σ_{C_D}	σ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}
1	12.08	56.18	17.69	9.42	26.32	16.24	57.78	64.00	30.10
2	7.60	39.15	15.82	7.96	24.57	35.08	58.08	15.21	26.30
3	19.75	39.64	21.49	22.99	26.96	39.49	74.68	59.39	59.01
4	16.34	25.80	72.13	24.36	22.23	77.77	59.16	17.31	120.04
5	8.01	22.90	12.52	4.05	9.64	12.95	123.01	46.85	39.49
6	9.54	28.47	31.29	7.36	17.11	18.91	47.60	17.85	58.06
7	7.00	34.88	29.27	18.68	21.88	46.38	73.16	18.77	99.47
8	9.86	46.67	13.81	15.78	27.19	26.14	38.08	13.94	27.48
9	15.24	22.60	48.10	8.90	9.28	64.42	60.25	9.97	36.79
10	24.78	23.71	40.78	18.46	10.92	18.74	45.78	78.28	90.72
11	11.85	26.75	40.20	7.81	16.20	33.88	32.33	13.52	30.71
12	9.43	47.78	78.93	7.75	33.50	81.71	54.00	50.57	373.03
13	7.86	84.29	14.10	8.76	35.52	33.15	44.96	65.04	32.95
14	19.06	33.60	19.48	18.57	17.23	30.97	40.55	38.95	103.41
15	33.63	63.67	15.18	36.51	28.50	24.65	113.56	28.96	32.14
16	18.37	52.44	20.94	12.79	26.86	13.93	115.38	52.67	48.18
17	7.61	38.61	23.54	7.40	20.02	32.20	44.08	23.69	29.60
18	10.28	30.92	63.17	13.25	14.49	57.26	98.88	16.49	162.23
19	16.65	117.84	97.43	13.38	46.11	54.68	22.22	72.62	279.46
20	10.02	39.02	8.70	9.84	29.17	24.19	35.59	17.37	76.24
21	9.38	36.37	8.12	11.61	20.74	19.24	43.09	20.57	17.81
22	10.83	30.64	57.75	9.13	13.28	31.37	38.24	28.38	161.41
23	25.81	31.88	64.93	16.83	15.24	79.06	93.76	60.24	277.39
24	7.80	34.13	13.33	10.04	18.68	30.90	45.93	35.73	140.90
25	7.14	22.11	25.15	10.11	6.60	24.31	22.72	10.63	35.56
26	7.49	15.48	25.19	6.99	17.69	49.37	40.38	28.95	119.43
27	11.09	42.37	16.90	9.73	28.25	32.83	38.87	11.99	37.07
28	16.86	23.91	102.01	15.74	11.77	40.12	66.05	131.39	378.99
29	11.89	42.02	49.57	19.35	21.03	28.26	23.44	37.43	90.45
30	9.74	35.51	78.55	9.60	15.10	68.15	32.10	15.88	79.76
Mean	13.10	39.64	37.54	13.10	21.07	38.21	56.12	36.75	103.14
Kriging	6.66	7.34	15.59	25.87	18.60	31.01	29.21	14.28	22.20



(a) Subsonic regime

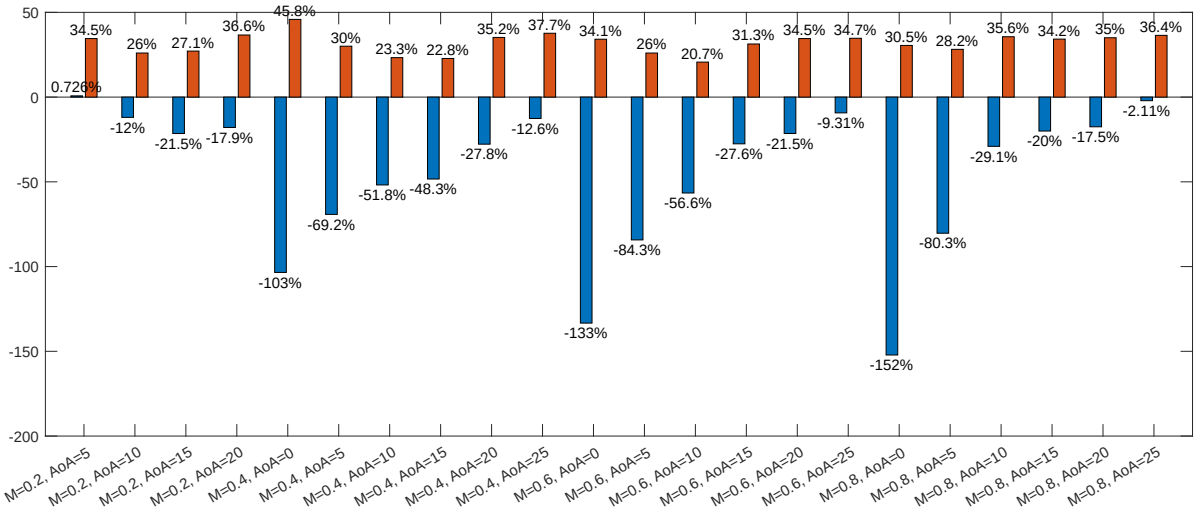


(b) Transonic regime

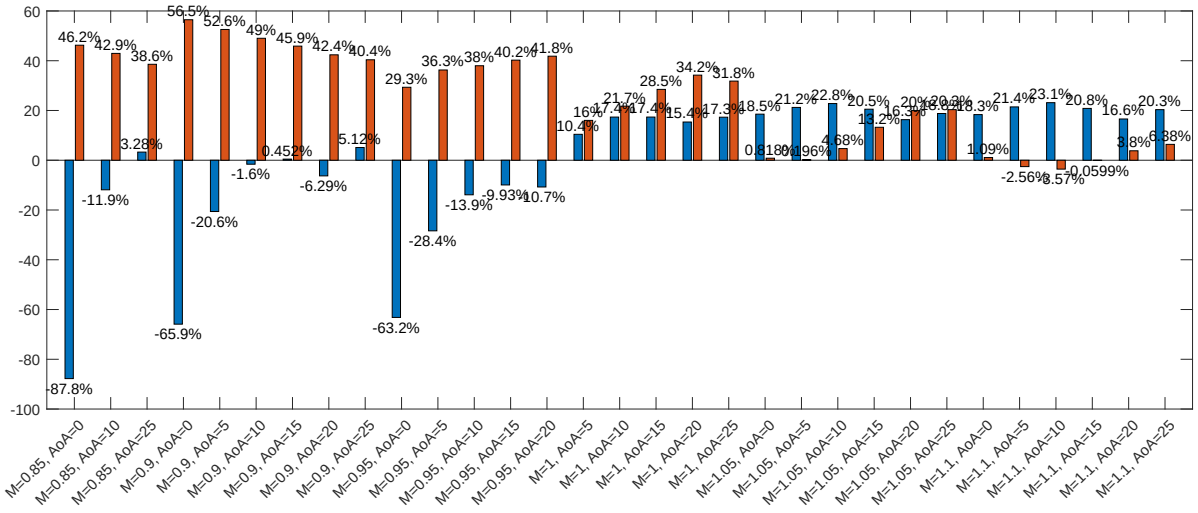


(c) Supersonic regime

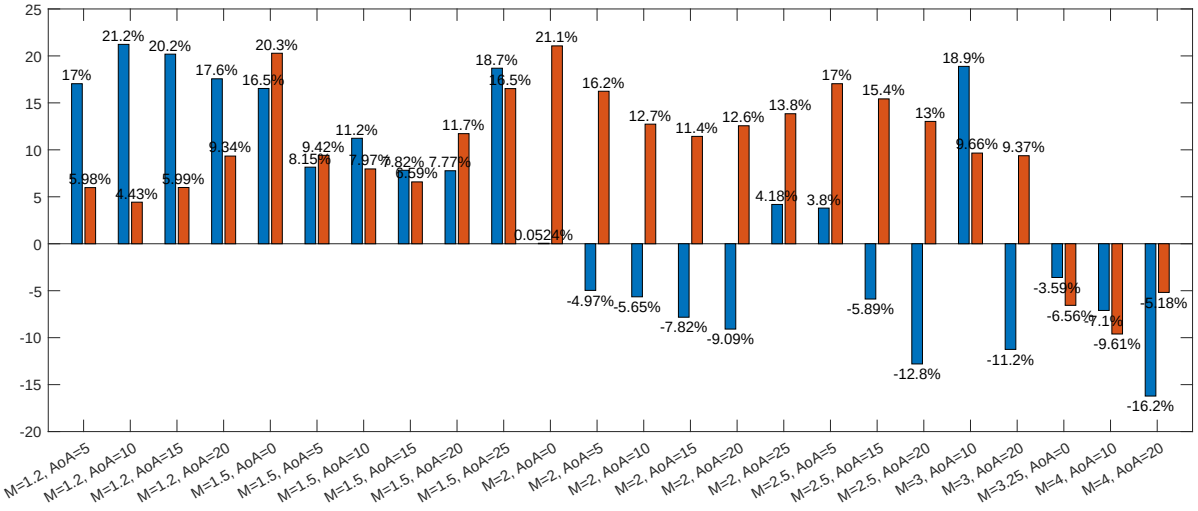
Figure C.1: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **lift coefficient**. 10 expensive points used for co-kriging; Run#27 from Table 6.7.



(a) Subsonic regime

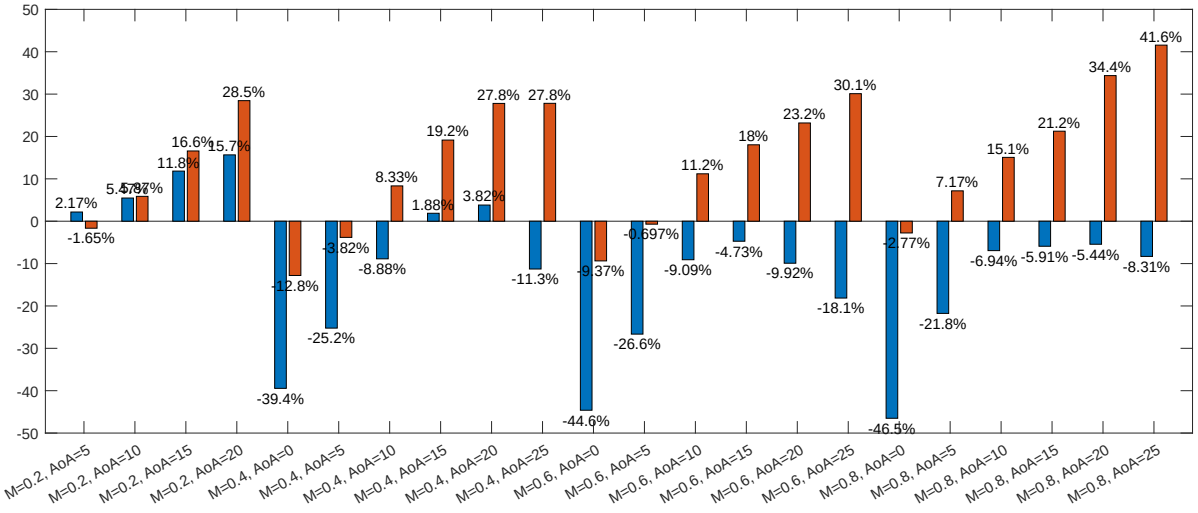


(b) Transonic regime

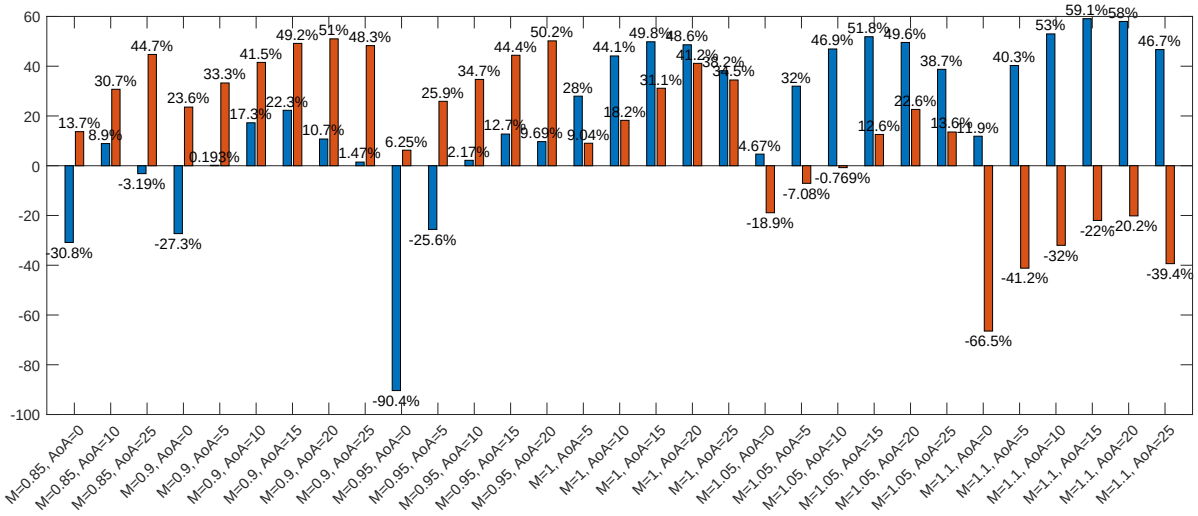


(c) Supersonic regime

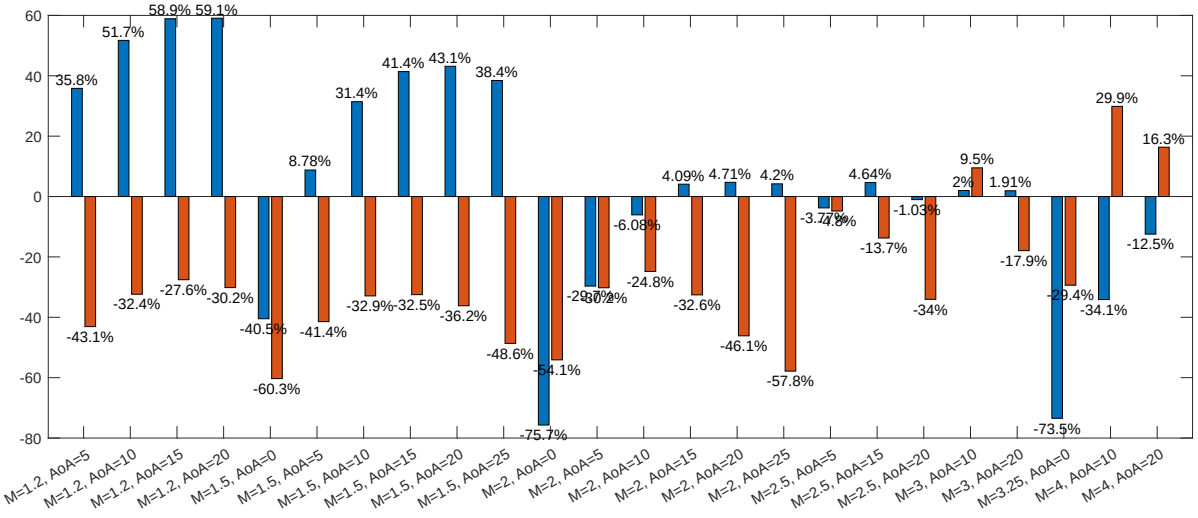
Figure C.2: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **drag coefficient**. 10 expensive points used for co-kriging; Run#27 from Table 6.7.



(a) Subsonic regime

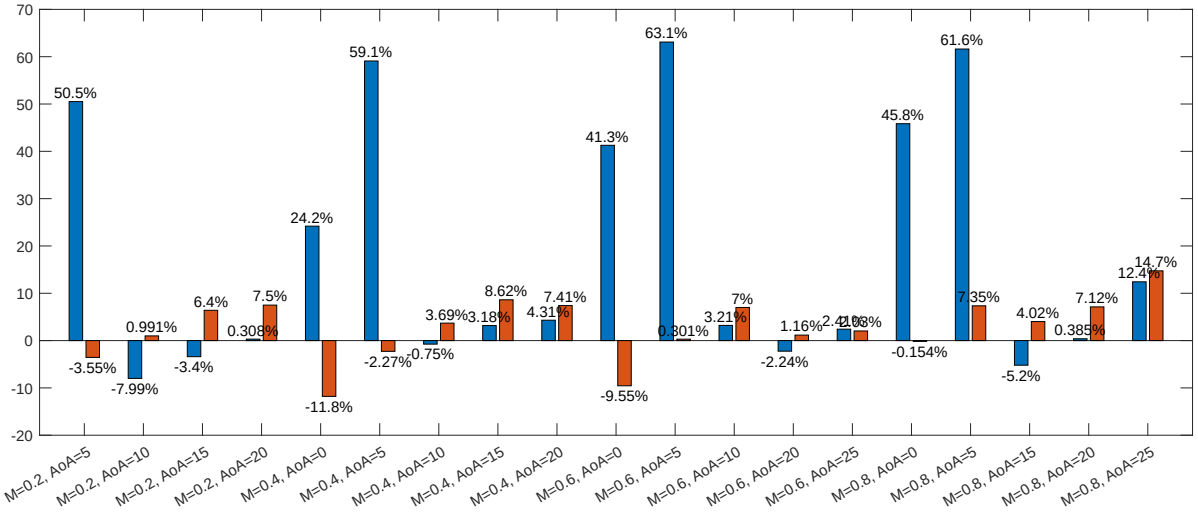


(b) Transonic regime

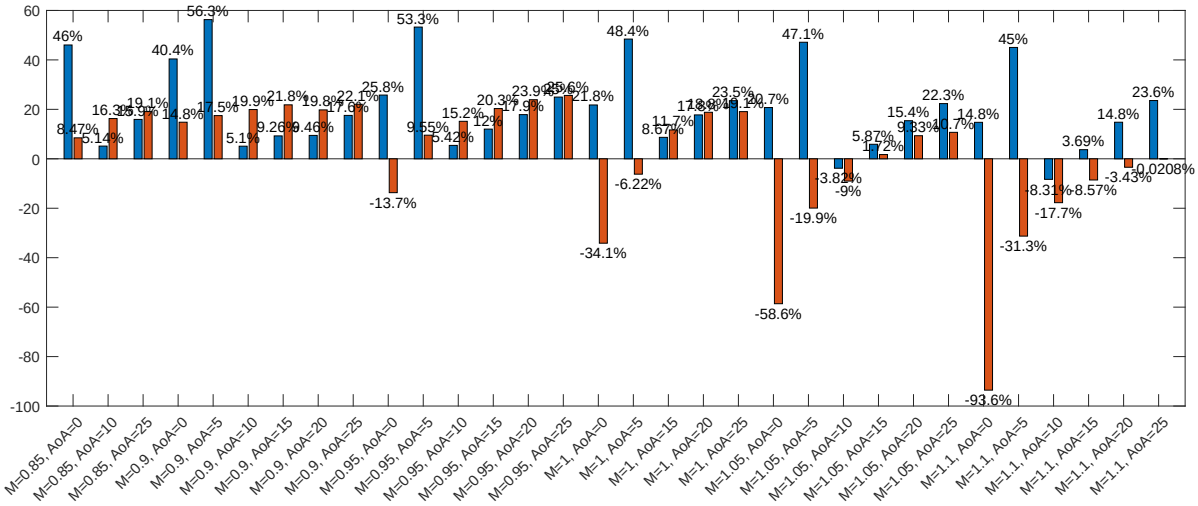


(c) Supersonic regime

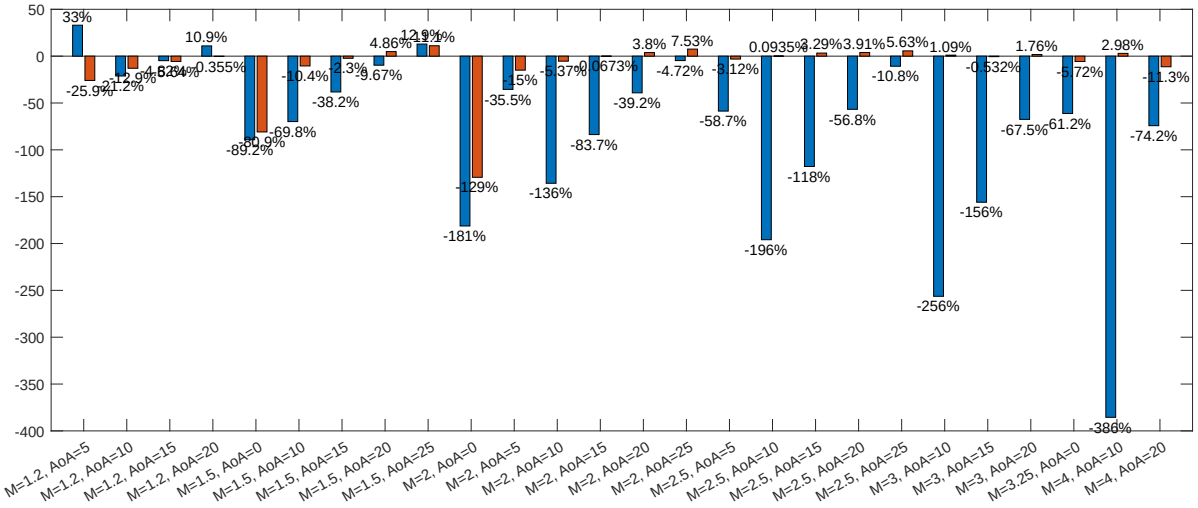
Figure C.3: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **pitching moment coefficient**. 10 expensive points used for co-kriging; Run#27 from Table 6.7.



(a) Subsonic regime

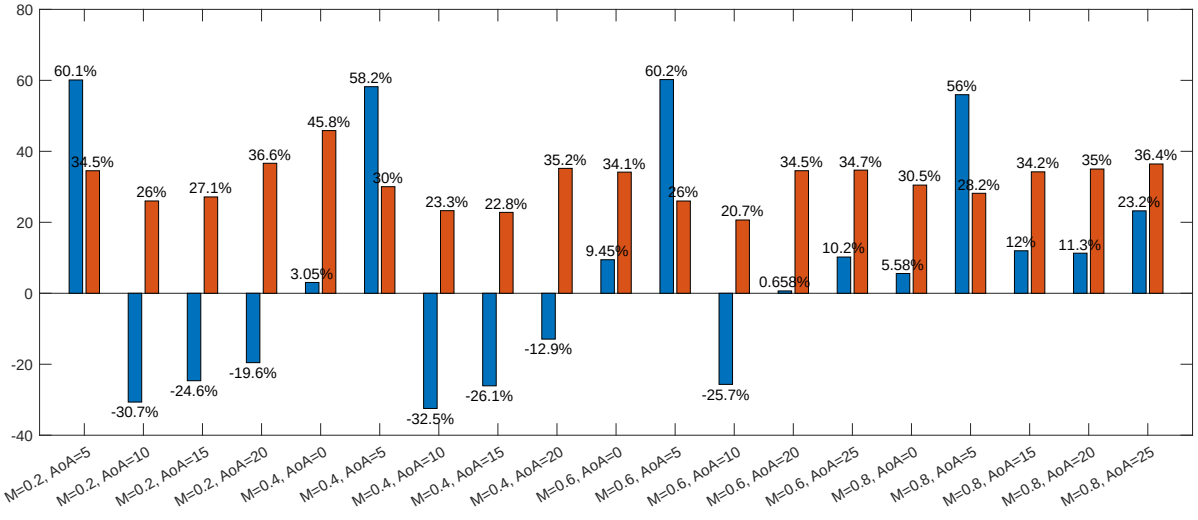


(b) Transonic regime

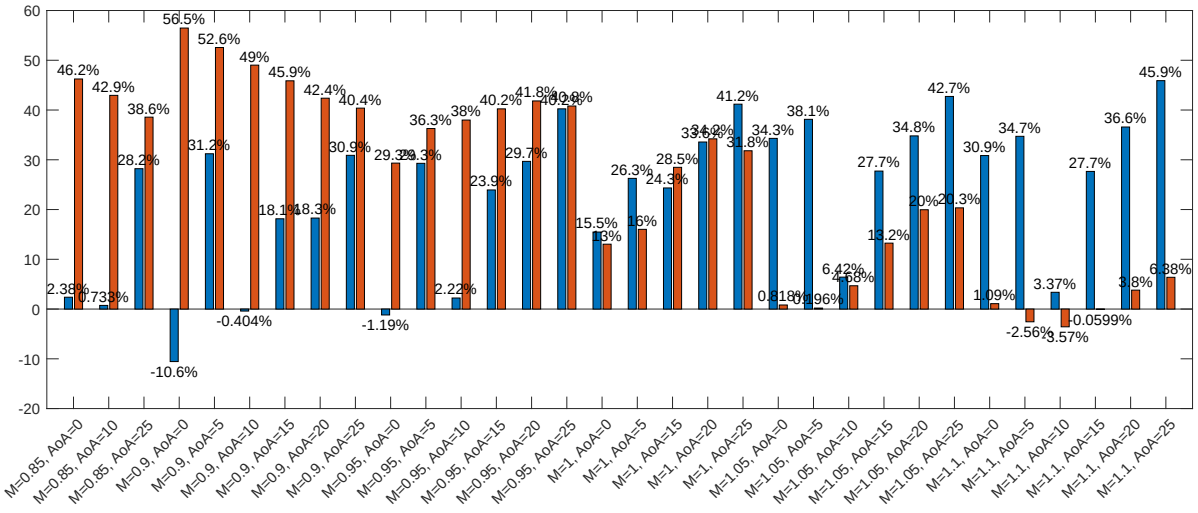


(c) Supersonic regime

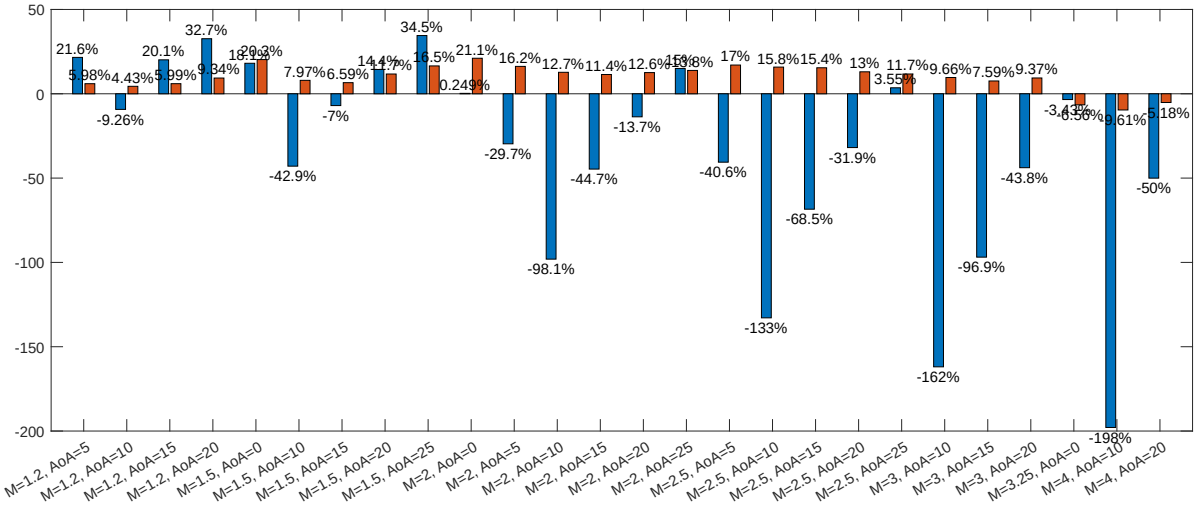
Figure C.4: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **lift coefficient**. 10 expensive points used for co-kriging; Run#23 from Table 6.7.



(a) Subsonic regime

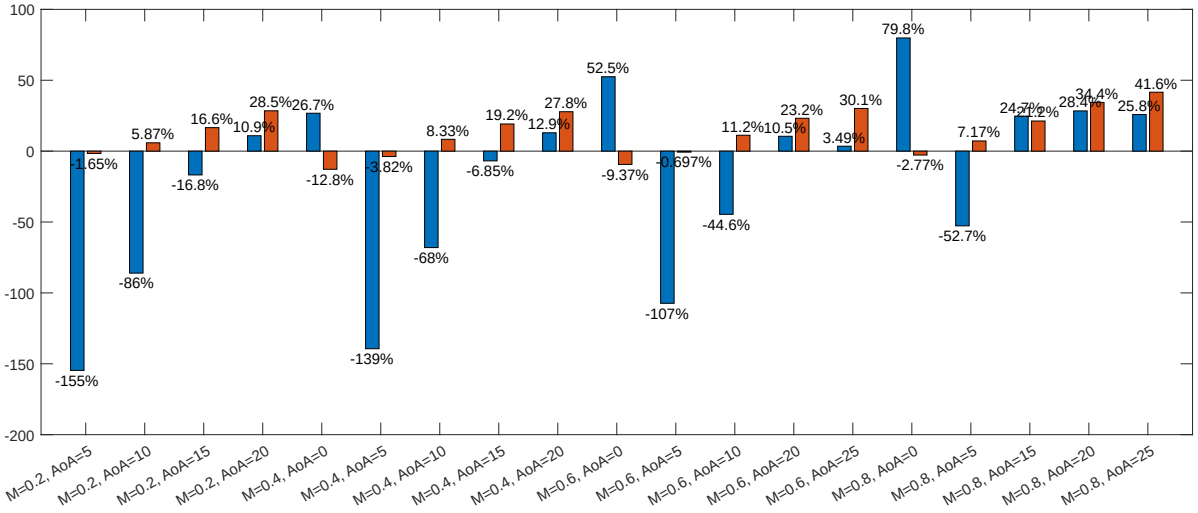


(b) Transonic regime

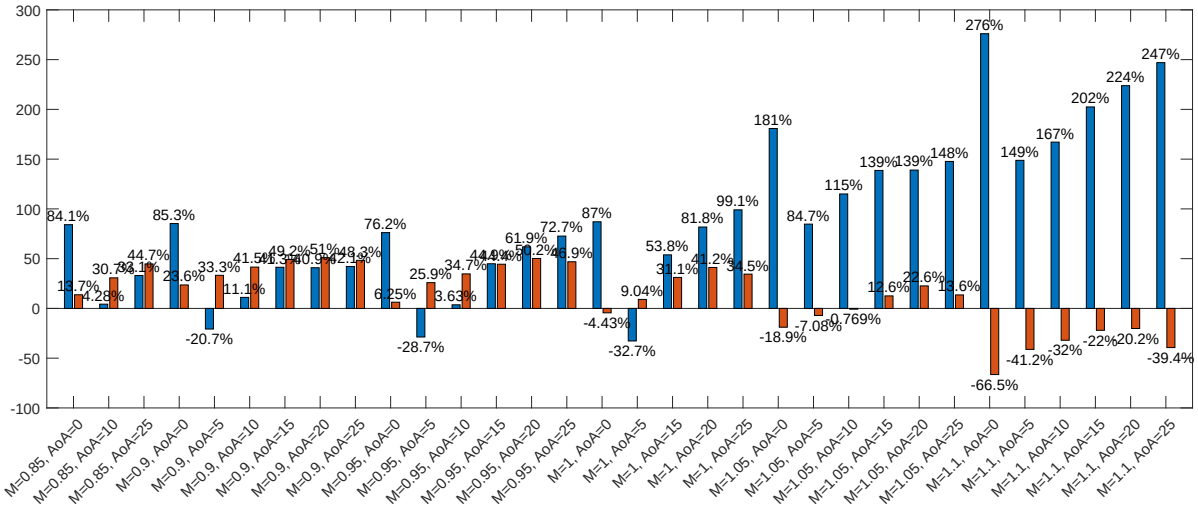


(c) Supersonic regime

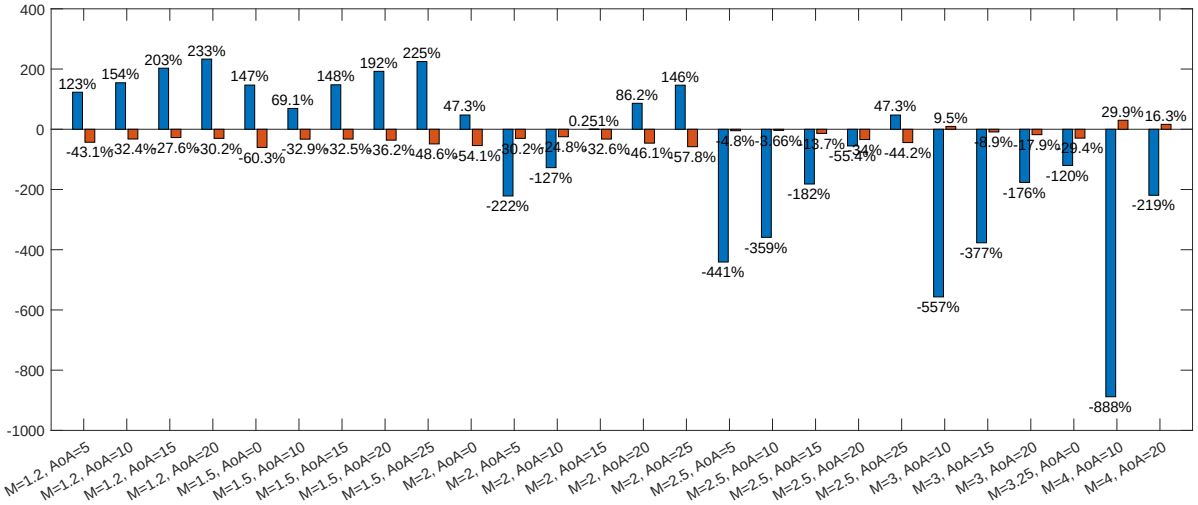
Figure C.5: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **drag coefficient**. 10 expensive points used for co-kriging; Run#23 from Table 6.7.



(a) Subsonic regime



(b) Transonic regime



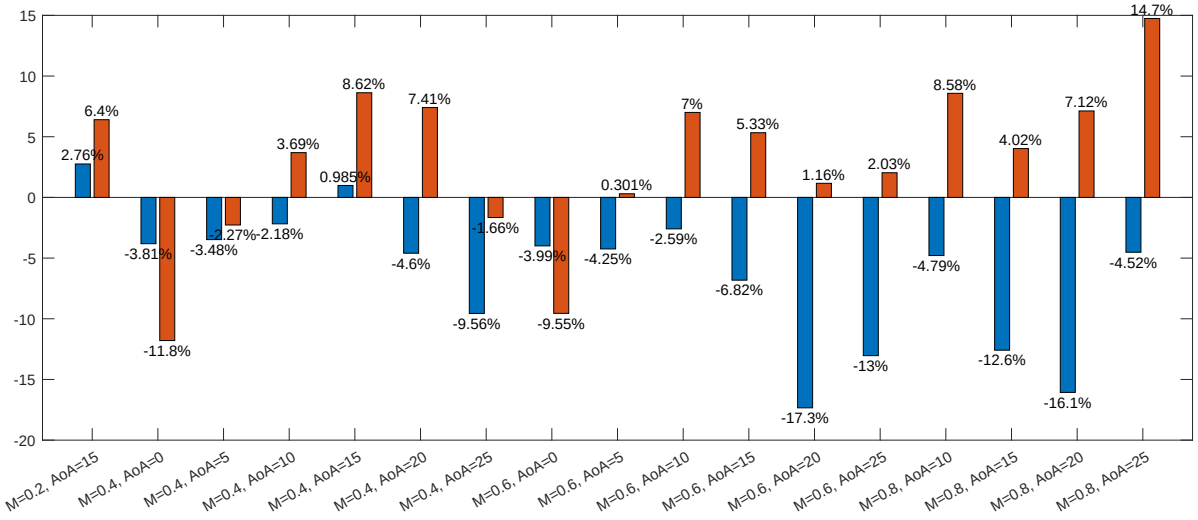
(c) Supersonic regime

Figure C.6: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **pitching moment coefficient**. 10 expensive points used for co-kriging; Run#23 from Table 6.7.

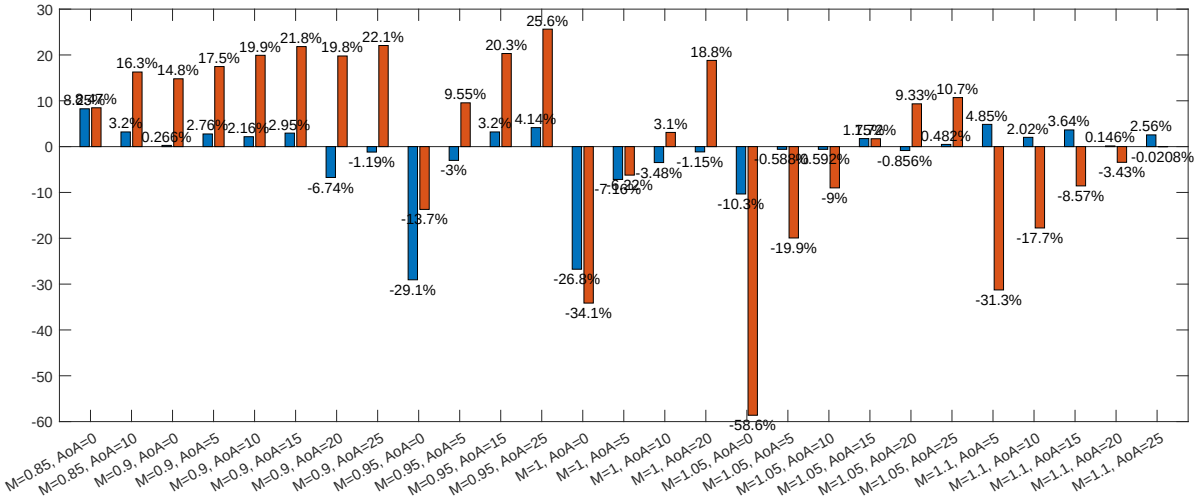
C.2. Case 2: 20 points

Table C.2: STDs of the thirty runs with 4 delimiting and 16 random points used for interpolation.

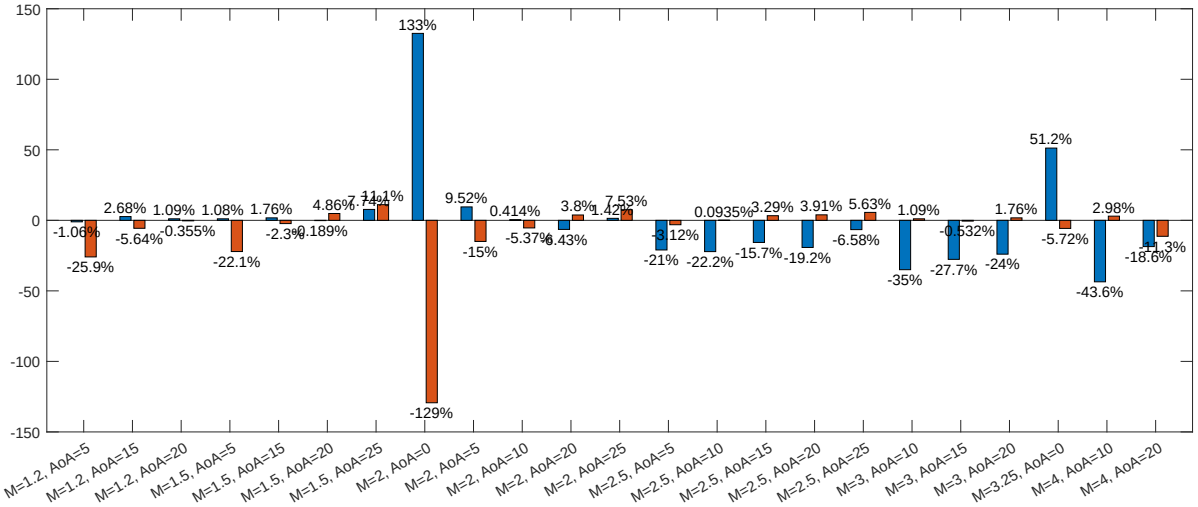
	Subsonic			Transonic			Supersonic		
Run	σ_{C_L}	σ_{C_D}	σ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}
1	6.04	11.38	19.56	4.77	8.58	6.54	29.54	12.50	46.48
2	7.37	16.84	13.61	11.69	10.49	17.48	12.84	9.47	15.17
3	4.32	13.03	7.78	4.17	7.51	4.98	28.41	19.43	26.02
4	12.96	45.29	25.30	22.76	25.27	23.64	48.62	12.29	27.96
5	5.13	29.03	17.45	12.02	16.58	12.59	13.51	7.67	21.98
6	4.57	17.49	21.27	4.37	14.71	7.99	42.77	14.56	64.53
7	9.42	26.70	12.96	36.38	21.75	30.47	31.15	12.97	33.64
8	5.59	29.97	13.59	18.96	26.07	20.62	18.40	7.82	14.36
9	9.90	17.49	19.67	6.68	15.58	32.96	16.67	73.41	43.20
10	7.90	12.77	19.33	16.60	8.50	13.70	34.49	7.80	68.20
11	11.77	28.86	18.17	5.73	8.95	14.21	40.97	19.09	64.70
12	11.49	42.63	20.25	16.33	23.57	17.89	25.49	12.85	19.12
13	11.90	20.44	31.73	4.93	11.35	16.35	17.56	9.10	40.64
14	7.15	11.52	6.73	14.20	12.48	17.44	43.78	7.30	11.65
15	8.22	26.16	10.40	4.60	19.84	27.68	10.53	5.42	32.37
16	3.70	9.89	7.07	16.24	5.29	14.10	23.40	10.91	16.08
17	5.65	6.89	6.76	8.50	8.42	10.02	34.88	8.91	22.19
18	15.40	61.11	31.46	22.45	40.12	33.19	30.12	6.09	32.15
19	9.49	33.34	8.74	13.56	19.62	20.96	22.31	14.34	14.92
20	6.77	27.13	8.89	3.90	15.75	8.38	29.79	12.17	34.07
21	7.49	16.60	9.58	4.83	9.34	5.86	21.49	9.51	15.48
22	11.06	24.37	13.78	5.51	16.84	10.71	17.51	10.29	9.65
23	11.67	24.32	15.60	9.89	10.04	14.42	51.75	10.23	43.25
24	15.08	18.07	7.91	10.31	13.09	9.95	27.61	10.46	12.57
25	5.39	10.73	8.96	10.18	5.56	15.54	18.77	18.43	24.28
26	7.73	12.47	16.82	7.48	8.65	11.31	41.37	13.61	32.23
27	7.85	37.82	22.79	6.10	22.54	14.36	21.19	28.39	44.24
28	7.87	17.50	6.52	6.13	14.46	8.75	38.57	29.58	13.27
29	13.05	37.41	31.00	12.28	20.25	12.52	10.62	9.19	19.80
30	5.45	13.54	10.24	15.03	6.74	25.22	29.57	9.11	11.54
Mean	8.58	23.36	15.46	11.22	14.93	15.99	27.79	14.43	29.19
Kriging	6.66	7.34	15.59	25.87	18.60	31.01	29.21	14.28	22.20



(a) Subsonic regime

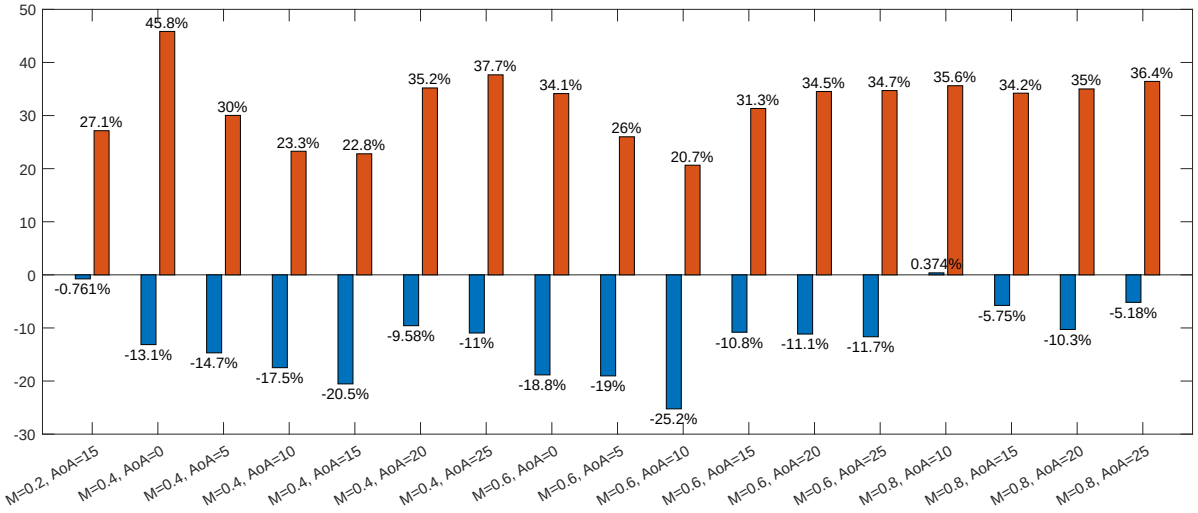


(b) Transonic regime

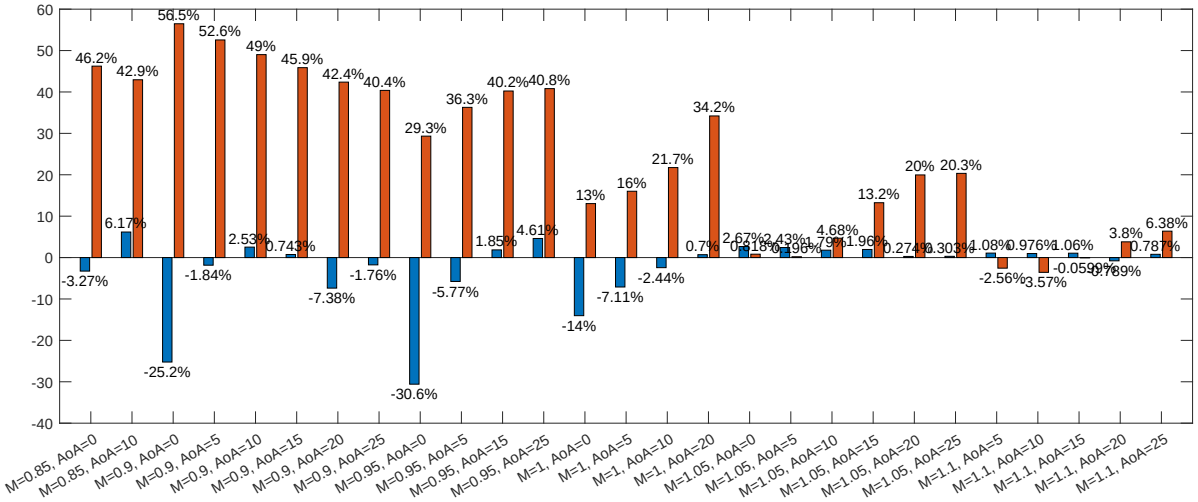


(c) Supersonic regime

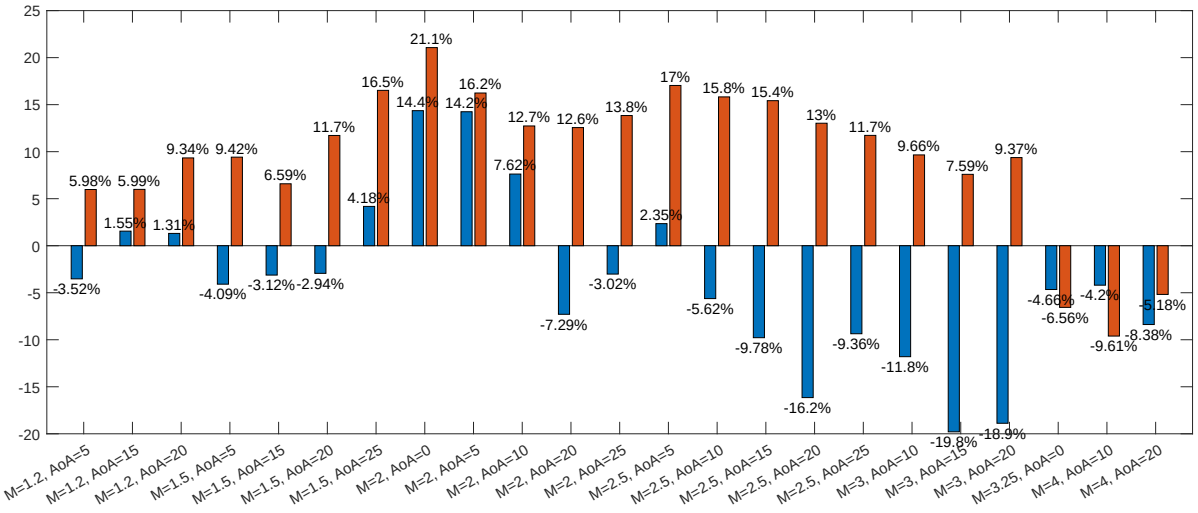
Figure C.7: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **lift coefficient**. 20 expensive points used for co-kriging; Run#17 from Table 6.8.



(a) Subsonic regime

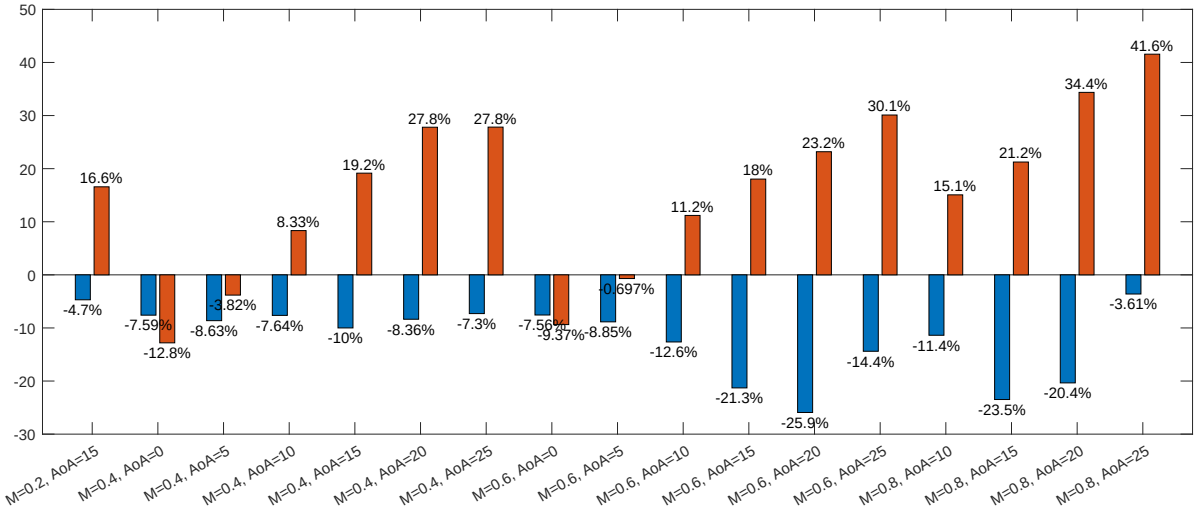


(b) Transonic regime

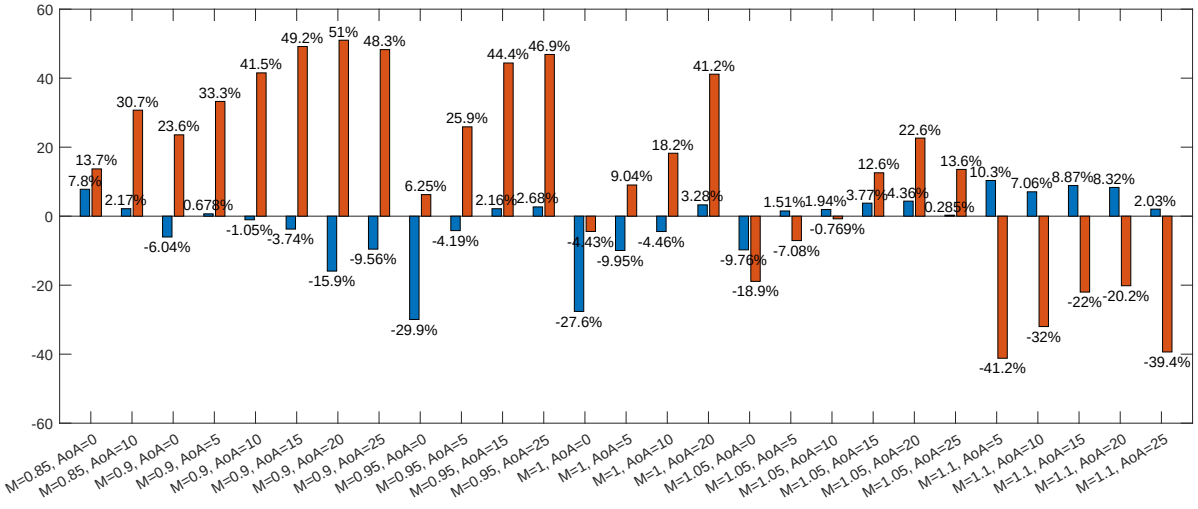


(c) Supersonic regime

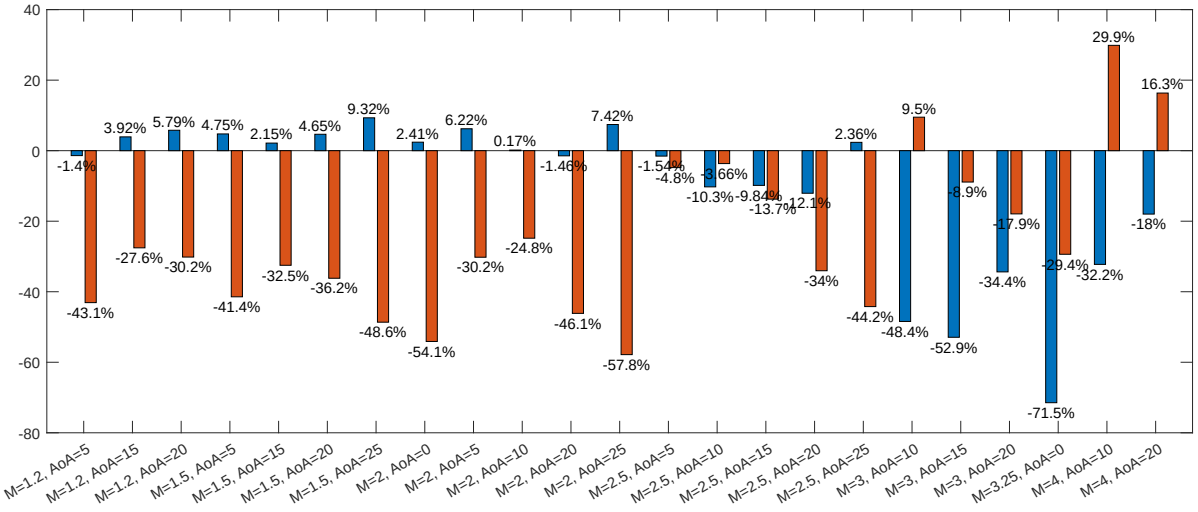
Figure C.8: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **drag coefficient**. 20 expensive points used for co-kriging; Run#17 from Table 6.8.



(a) Subsonic regime

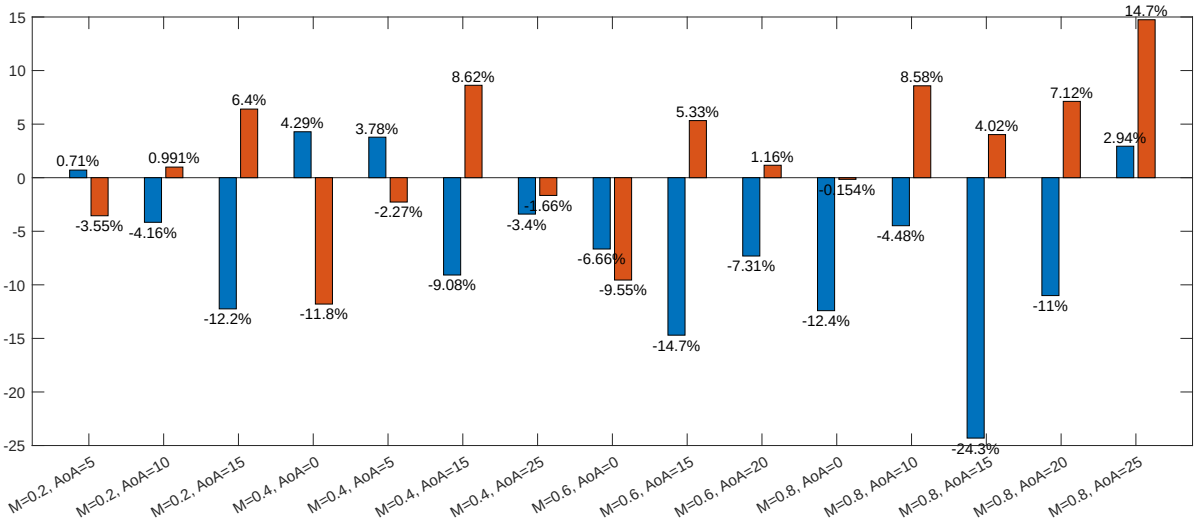


(b) Transonic regime

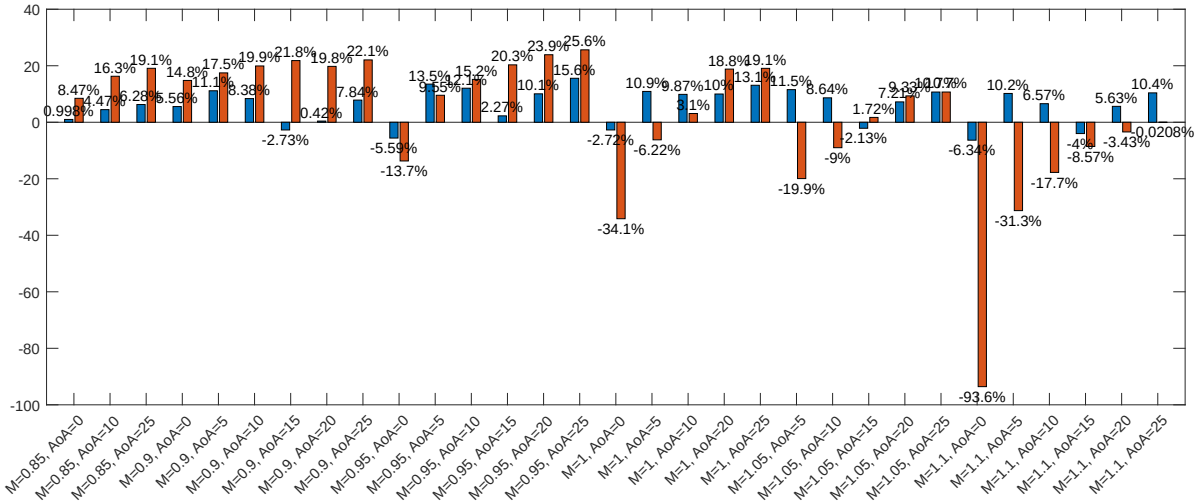


(c) Supersonic regime

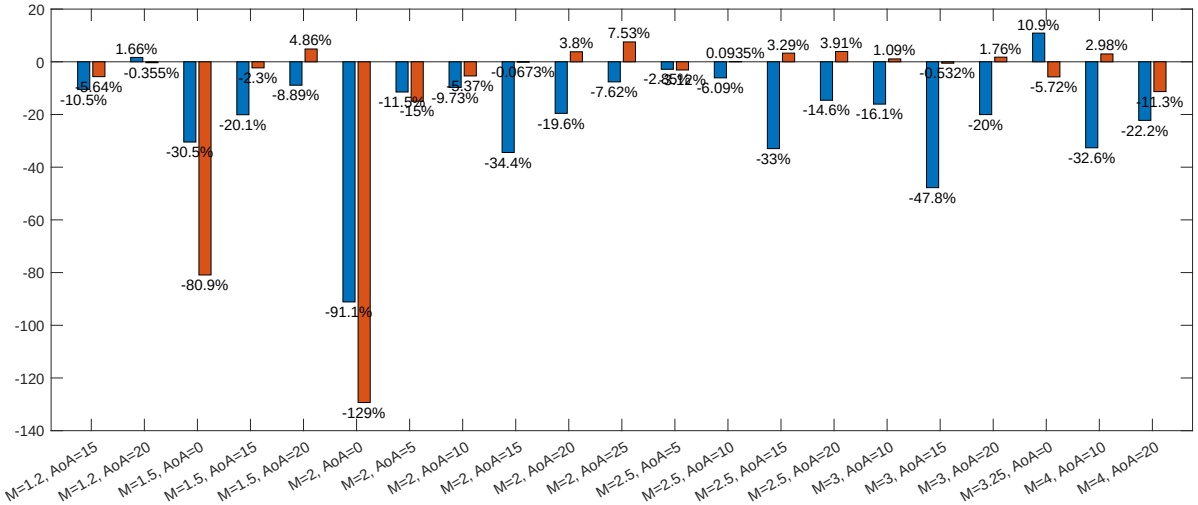
Figure C.9: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **pitching moment coefficient**. 20 expensive points used for co-kriging; Run#17 from Table 6.8.



(a) Subsonic regime

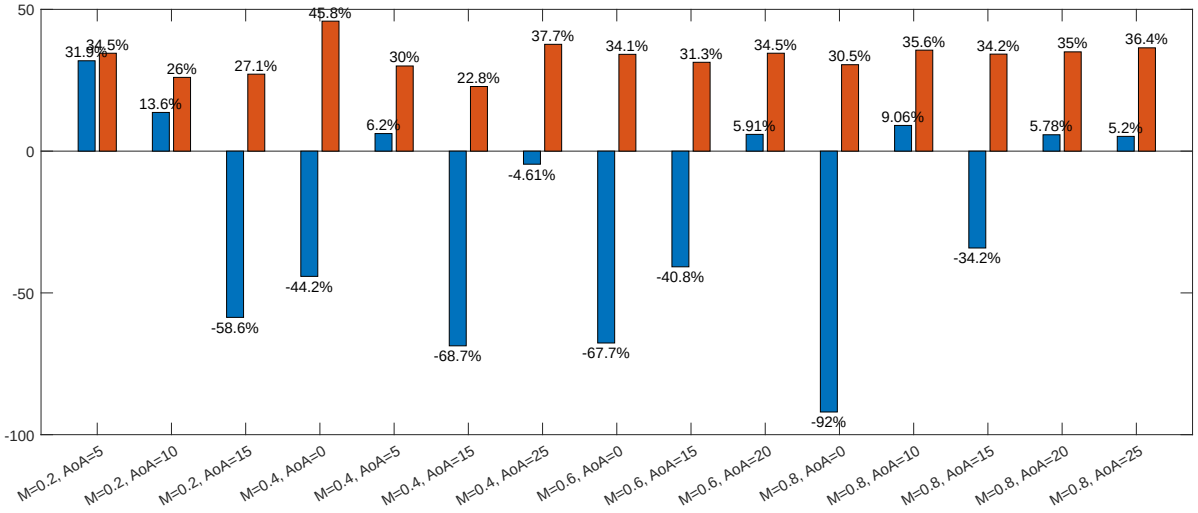


(b) Transonic regime

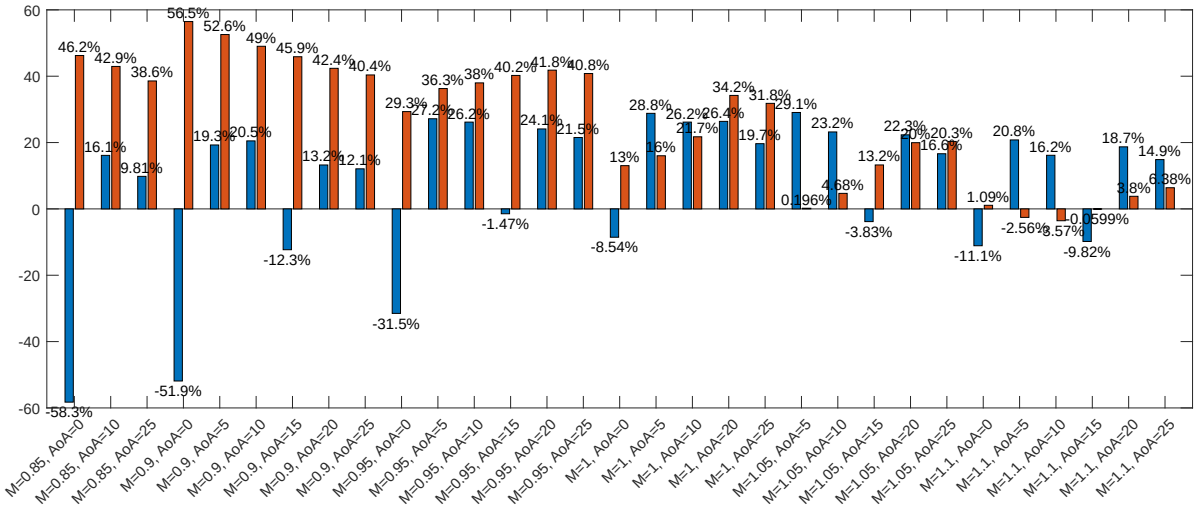


(c) Supersonic regime

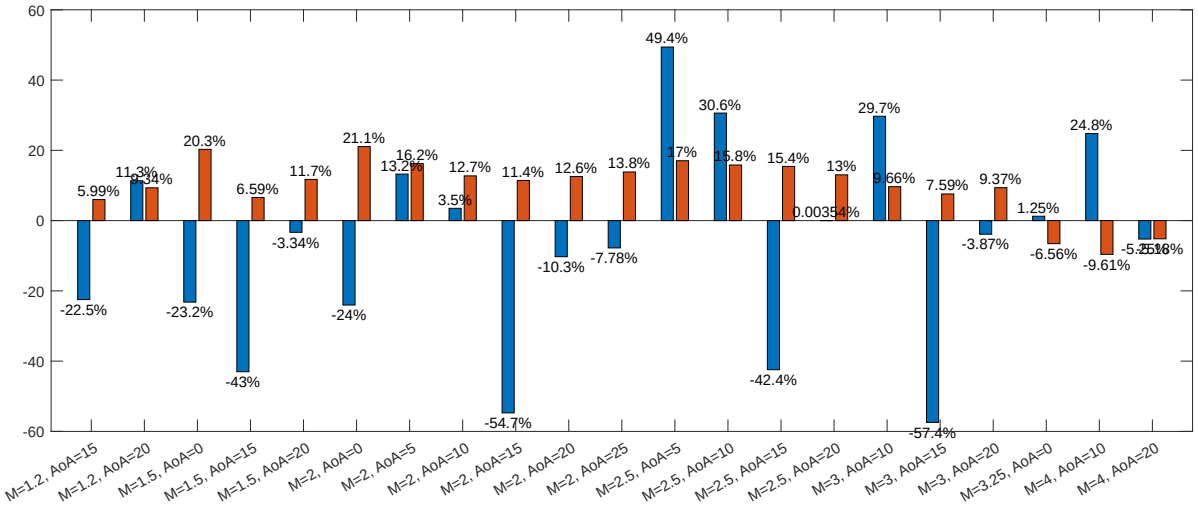
Figure C.10: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **lift coefficient**. 20 expensive points used for co-kriging; Run#27 from Table 6.8.



(a) Subsonic regime

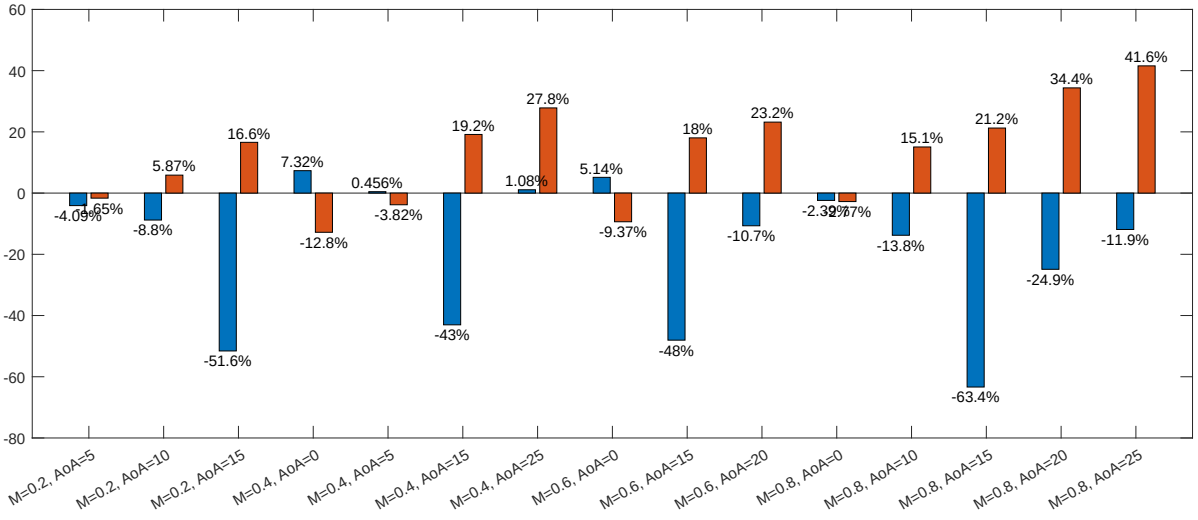


(b) Transonic regime

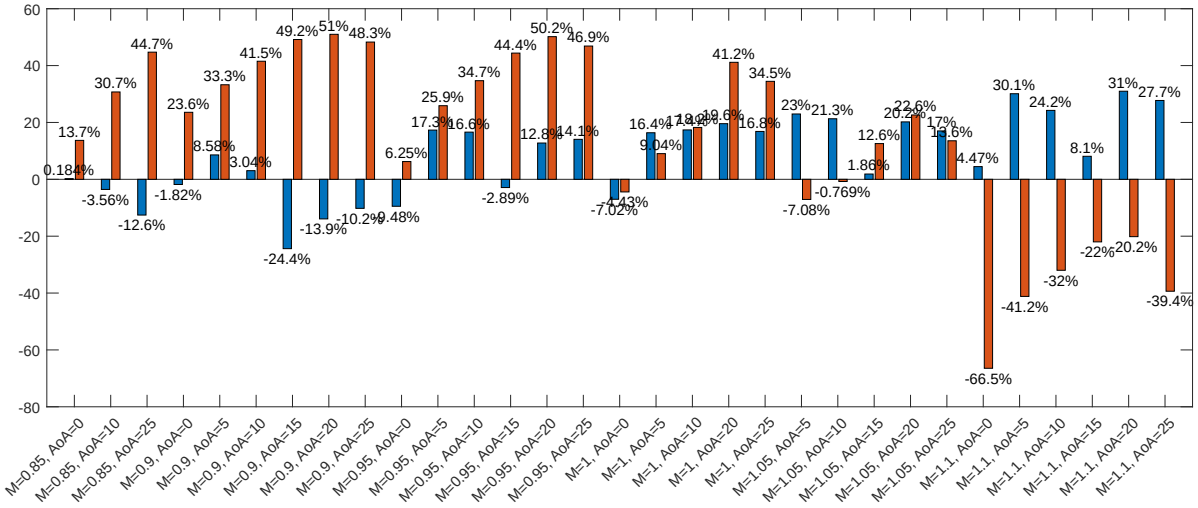


(c) Supersonic regime

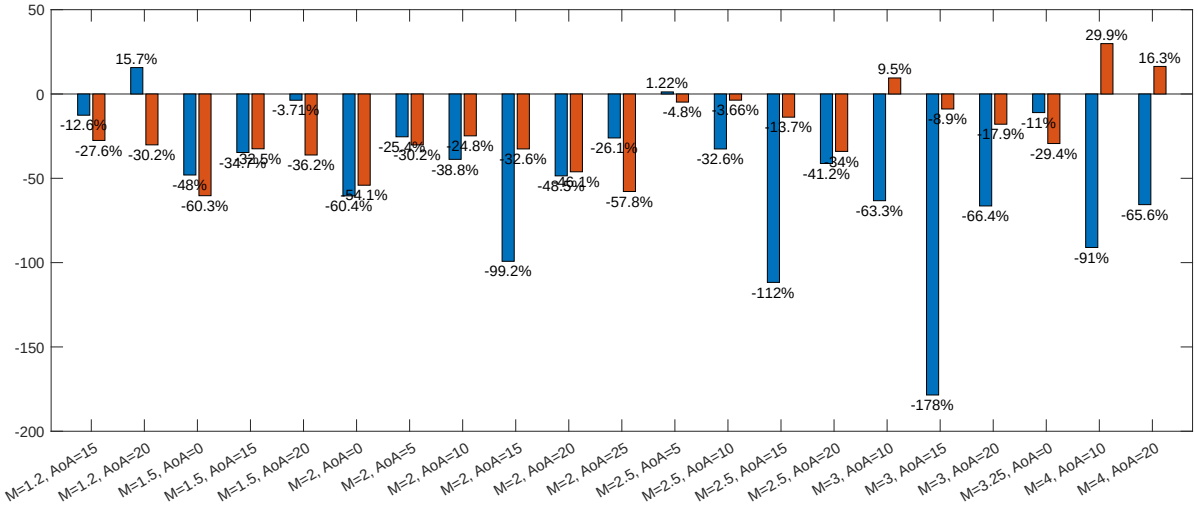
Figure C.11: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **drag coefficient**. 20 expensive points used for co-kriging; Run#27 from Table 6.8.



(a) Subsonic regime



(b) Transonic regime



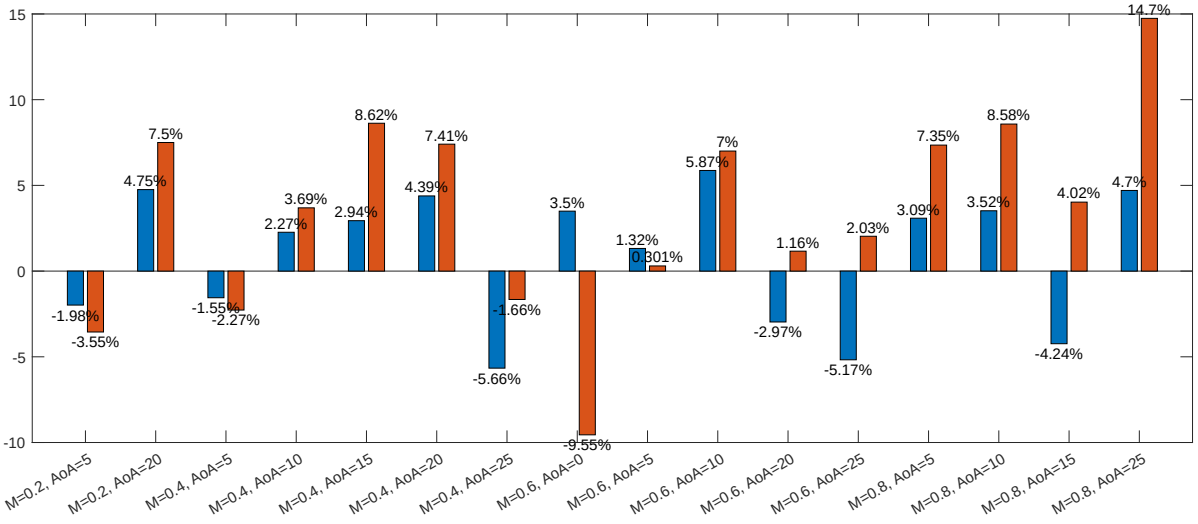
(c) Supersonic regime

Figure C.12: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **pitching moment coefficient**. 20 expensive points used for co-kriging; Run#27 from Table 6.8.

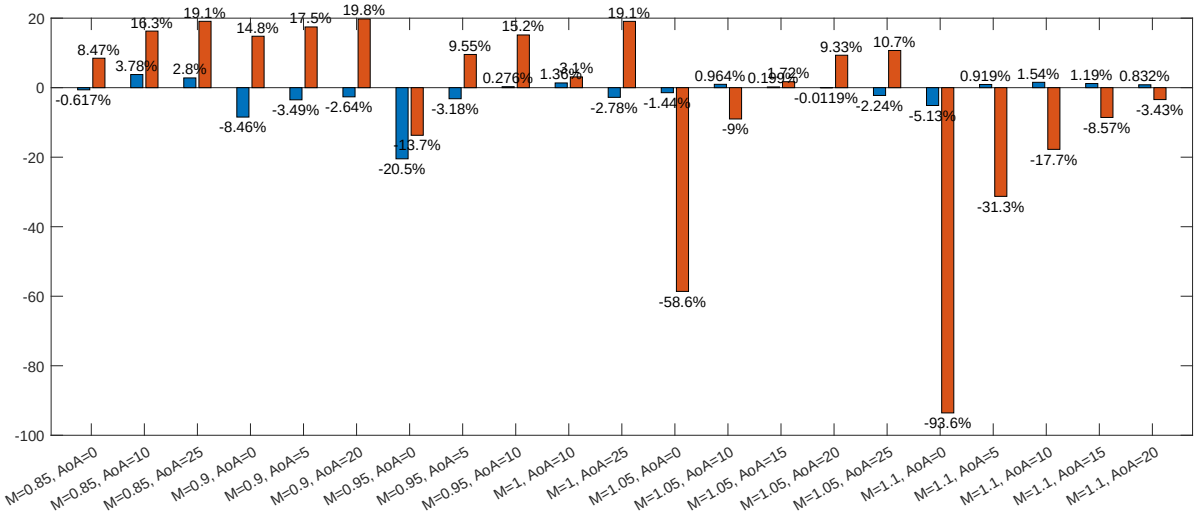
C.3. Case 3: 30 points

Table C.3: STDs of the thirty runs with 4 delimiting and 26 random points used for interpolation.

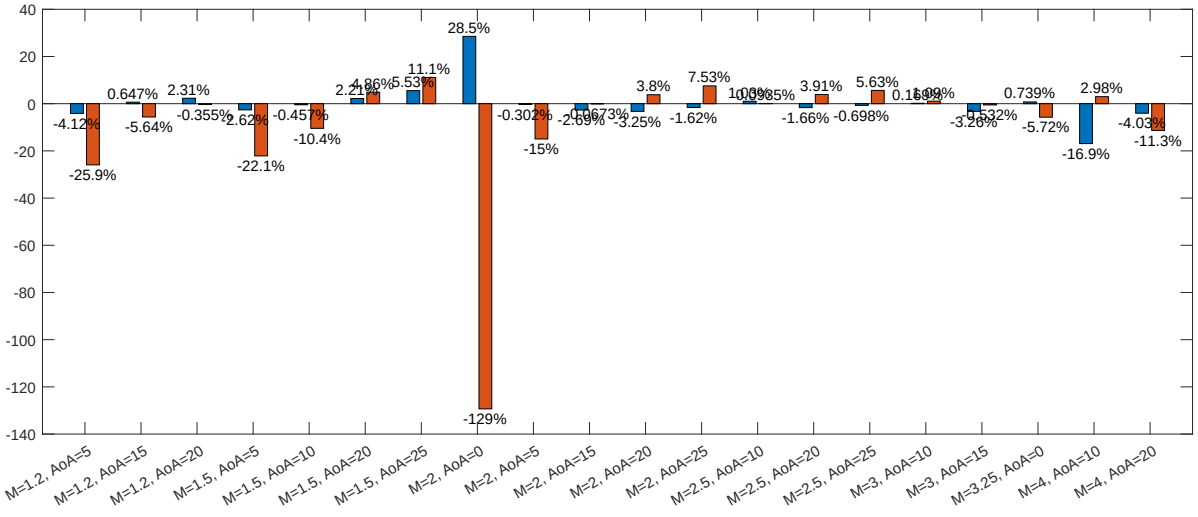
	Subsonic			Transonic			Supersonic		
Run	σ_{C_L}	σ_{C_D}	σ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}
1	6.86	8.85	7.32	5.74	14.44	7.31	7.72	6.69	6.28
2	7.26	12.66	6.95	3.71	7.98	6.36	13.41	4.74	9.65
3	5.09	21.30	9.65	6.01	14.20	6.76	3.84	4.23	10.54
4	4.60	14.95	20.25	4.65	7.25	8.71	9.40	8.63	45.27
5	4.80	13.15	5.59	10.26	13.90	14.85	8.91	9.59	10.63
6	5.23	17.08	5.64	2.58	3.01	6.37	14.75	4.08	22.00
7	5.44	11.90	5.97	4.56	9.94	5.40	15.57	5.56	11.51
8	11.50	17.52	10.11	6.60	11.65	6.81	42.73	5.56	13.96
9	6.18	7.62	6.90	8.24	15.61	9.44	12.05	5.57	9.60
10	5.56	4.60	4.57	4.10	6.72	8.78	15.21	10.64	27.34
11	7.61	14.34	5.34	3.92	4.53	5.55	12.06	7.22	19.76
12	5.70	11.17	6.55	5.51	11.13	4.50	21.32	12.97	12.08
13	4.87	5.67	3.87	5.94	7.90	7.98	26.30	9.00	49.18
14	6.74	18.18	16.52	8.25	7.68	12.16	14.85	6.42	22.33
15	8.26	12.42	9.81	20.40	13.31	24.94	37.46	11.20	15.69
16	4.95	7.00	5.60	8.45	4.93	10.64	3.74	5.69	11.71
17	7.23	9.37	8.63	6.29	10.83	6.87	13.32	13.26	18.33
18	5.43	9.40	9.78	7.84	6.75	12.90	6.44	4.79	23.38
19	9.42	13.42	8.84	4.71	9.29	7.42	8.01	7.31	6.43
20	5.89	12.86	8.59	4.61	6.28	7.78	17.86	11.27	20.06
21	6.83	6.13	9.13	5.83	5.68	6.87	6.44	5.04	12.23
22	5.73	9.28	5.06	6.12	6.38	7.69	25.11	8.20	8.78
23	9.26	17.51	8.74	7.27	7.35	25.76	29.67	8.92	25.78
24	3.99	6.20	5.97	4.54	4.66	7.24	13.90	8.78	51.51
25	3.89	6.09	5.07	5.14	8.95	5.10	8.00	4.45	12.67
26	6.21	8.40	8.72	3.70	6.21	8.69	13.54	8.67	14.07
27	8.09	12.62	9.48	7.06	10.00	6.37	13.71	4.25	9.47
28	6.44	13.00	8.95	7.48	10.21	10.34	10.63	6.58	12.00
29	7.62	4.82	7.75	6.96	13.15	11.84	5.26	5.45	5.84
30	13.41	8.57	16.64	9.67	7.09	11.27	15.96	7.91	16.63
Mean	6.67	11.20	8.40	6.54	8.90	9.42	14.91	7.42	17.82
Kriging	6.66	7.34	15.59	25.87	18.60	31.01	29.21	14.28	22.20



(a) Subsonic regime

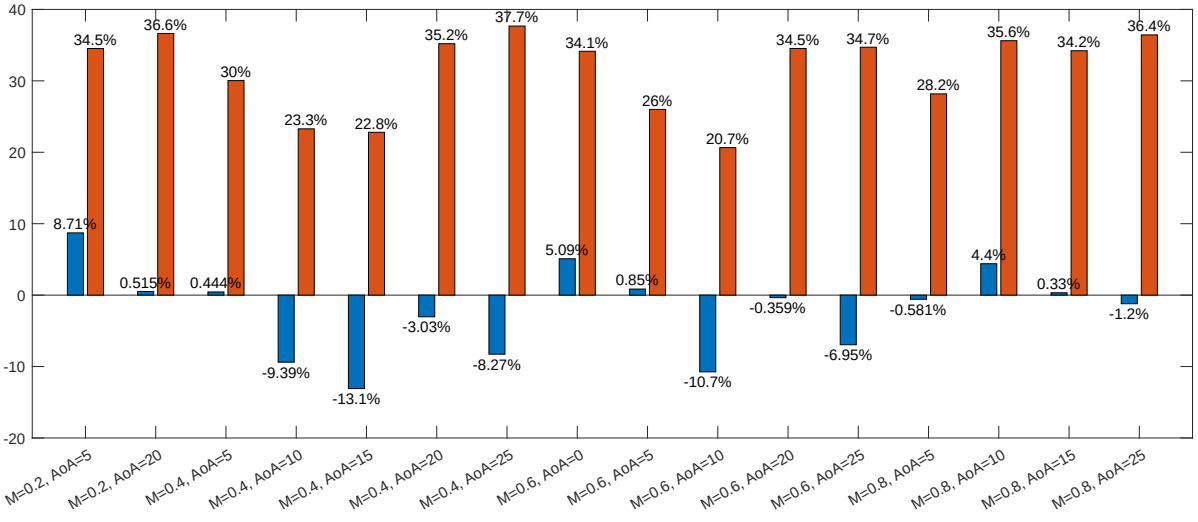


(b) Transonic regime

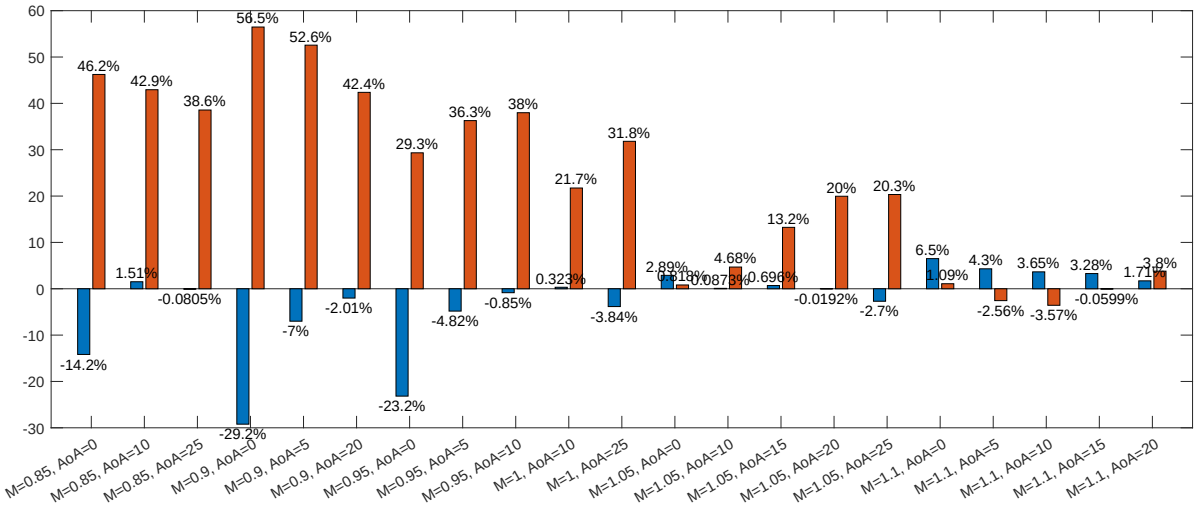


(c) Supersonic regime

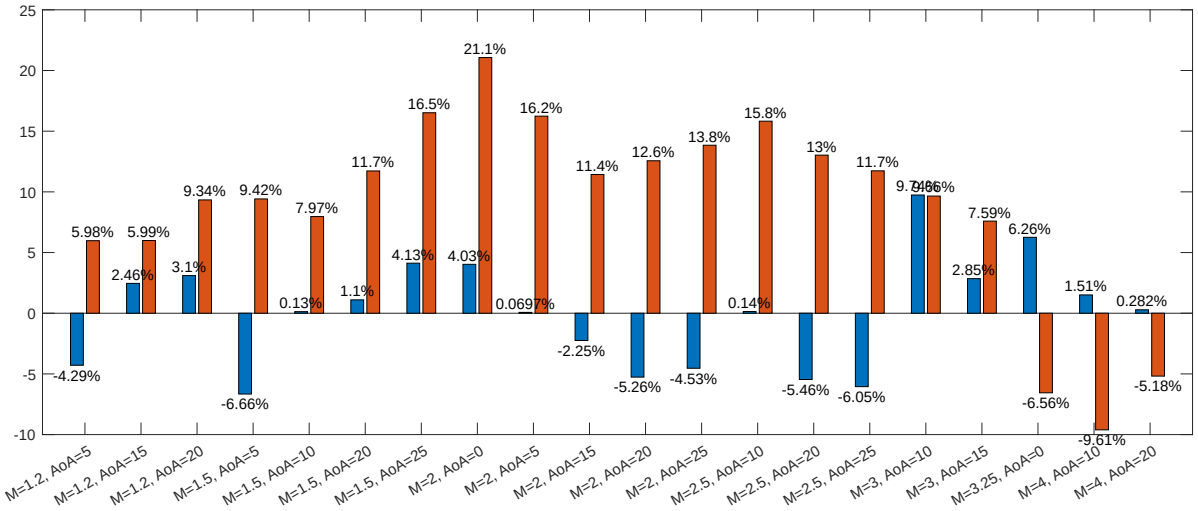
Figure C.13: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **lift coefficient**. 30 expensive points used for co-kriging; Run#25 from Table 6.9.



(a) Subsonic regime

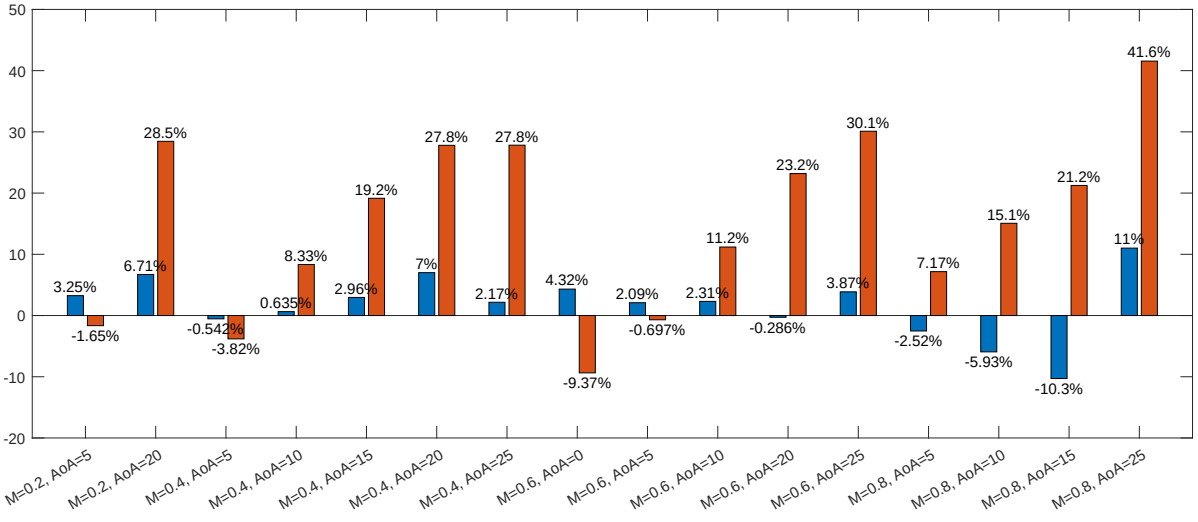


(b) Transonic regime

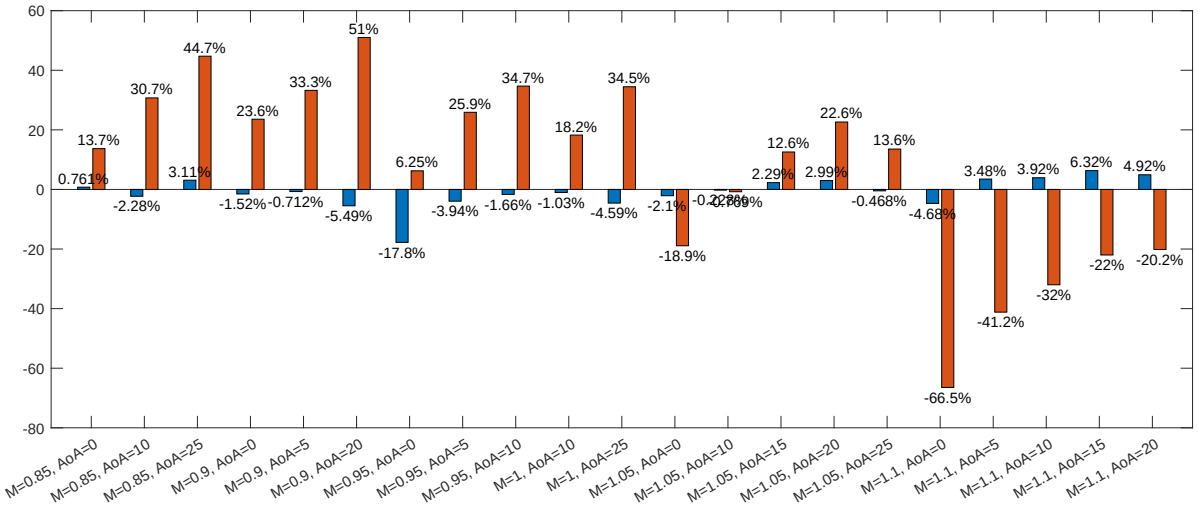


(c) Supersonic regime

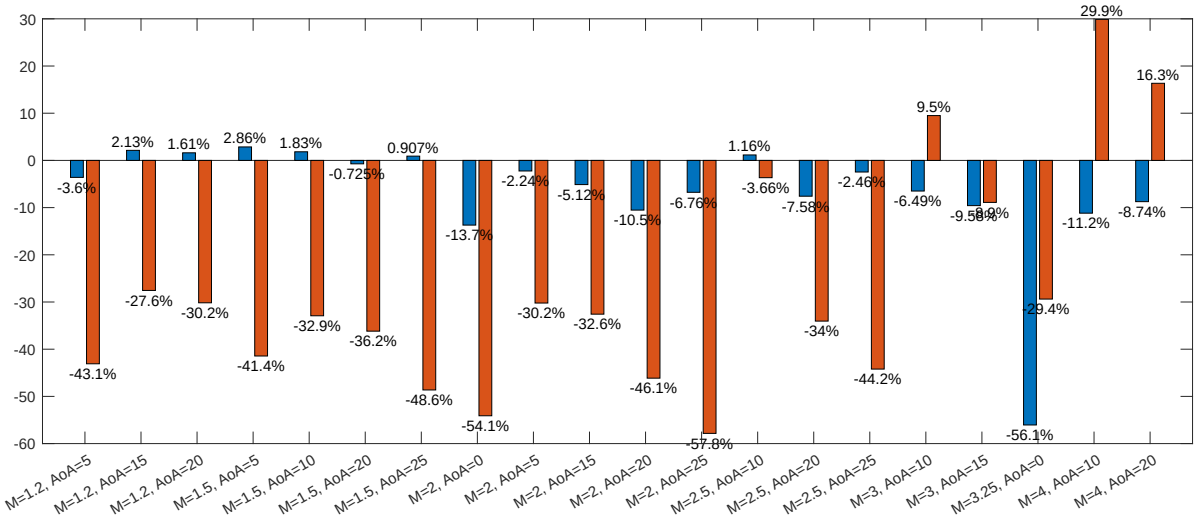
Figure C.14: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **drag coefficient**. 30 expensive points used for co-kriging; Run#25 from Table 6.9.



(a) Subsonic regime

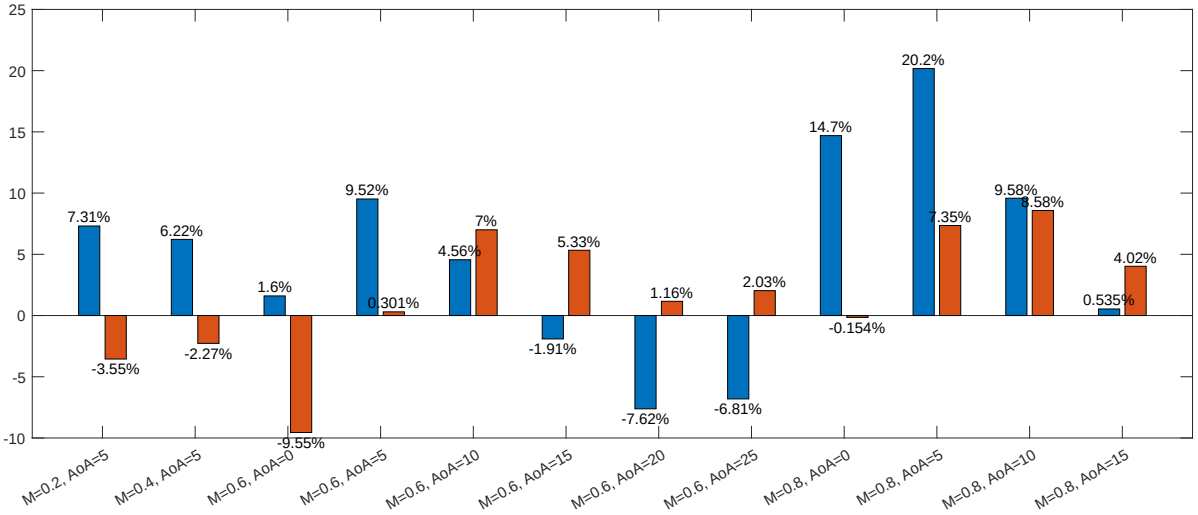


(b) Transonic regime

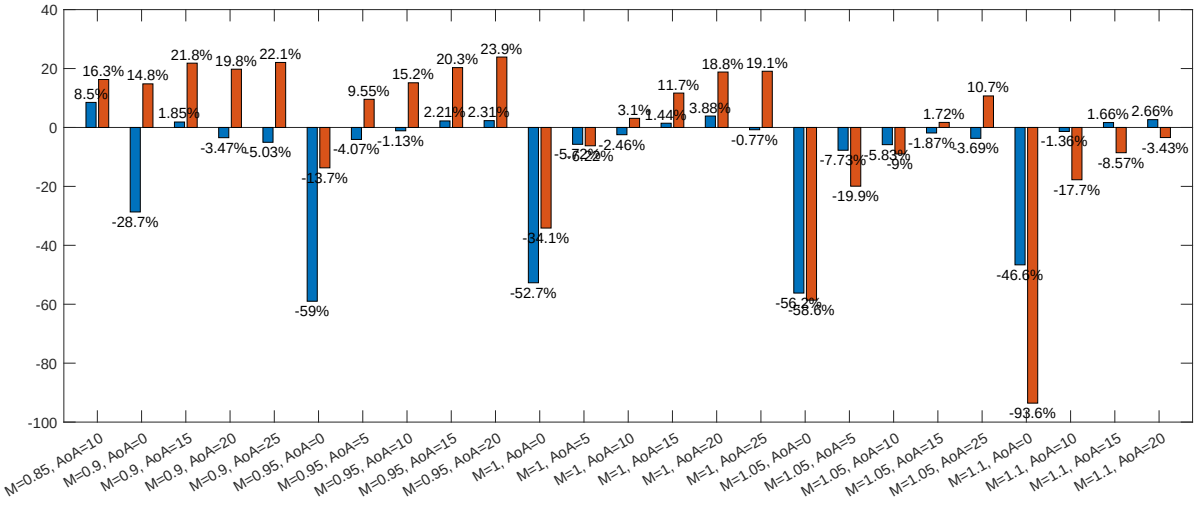


(c) Supersonic regime

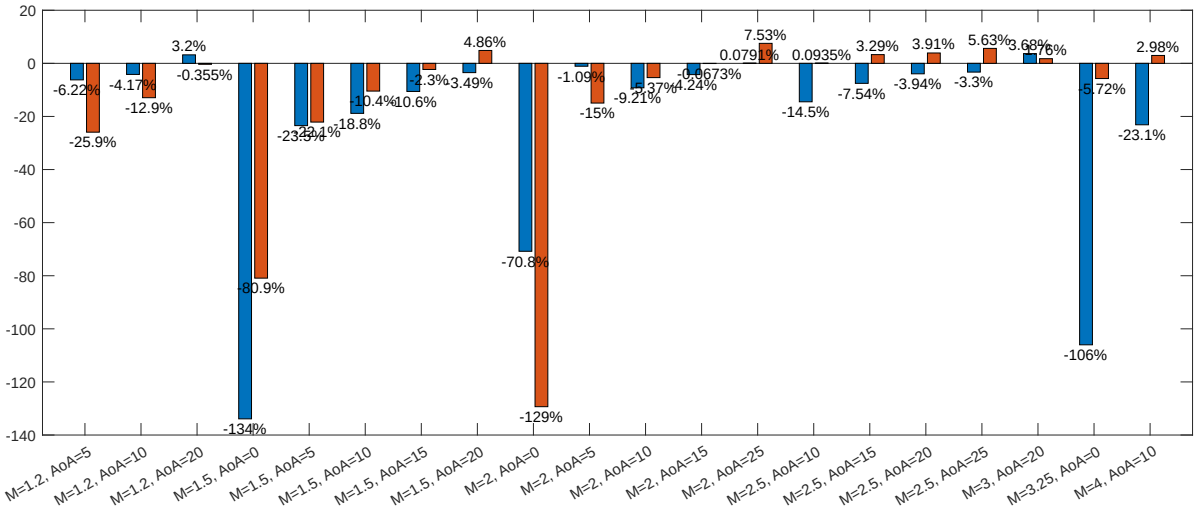
Figure C.15: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **pitching moment coefficient**. 30 expensive points used for co-kriging; Run#25 from Table 6.9.



(a) Subsonic regime

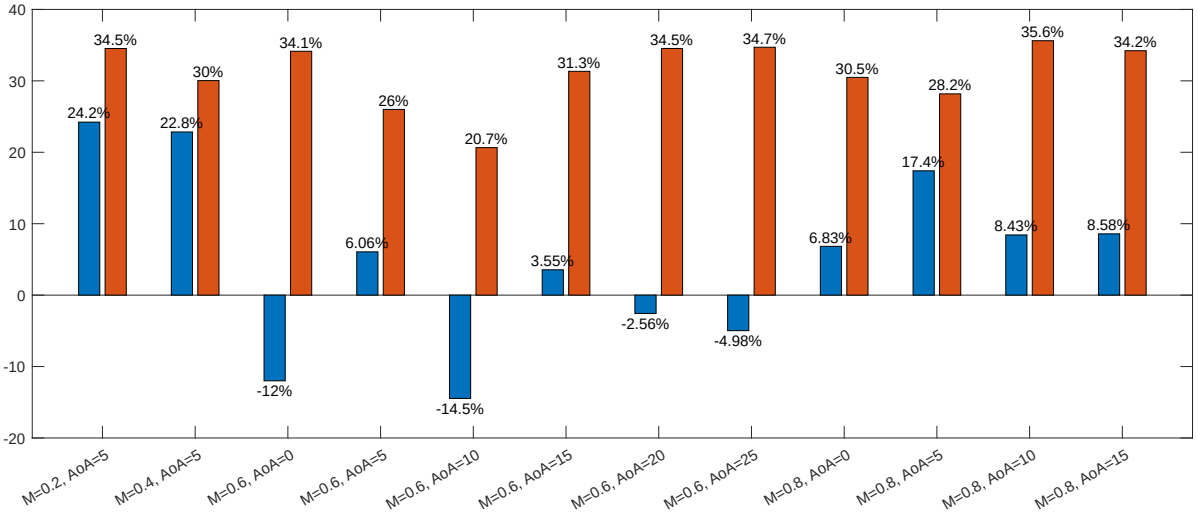


(b) Transonic regime

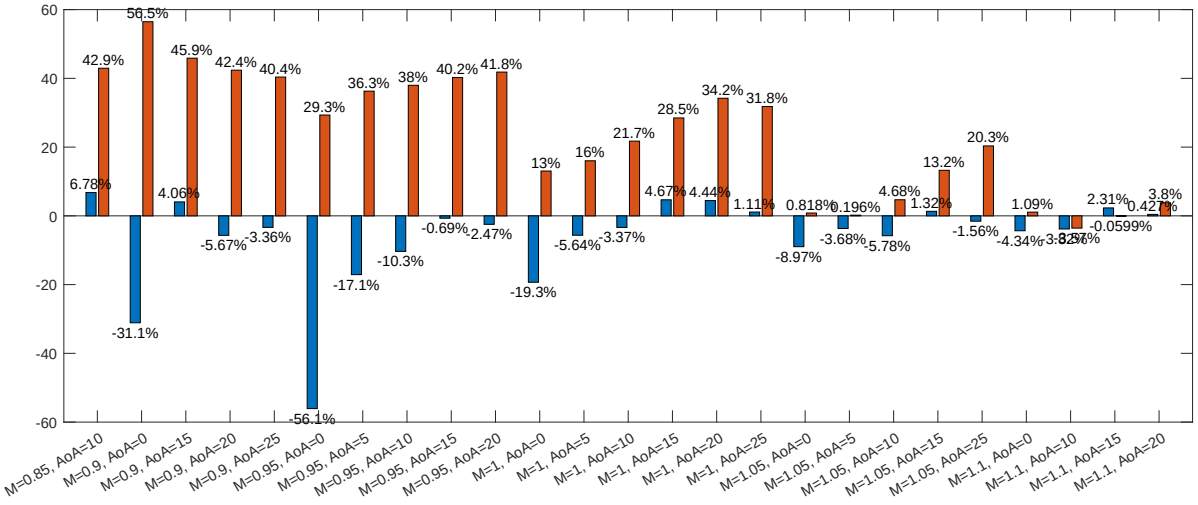


(c) Supersonic regime

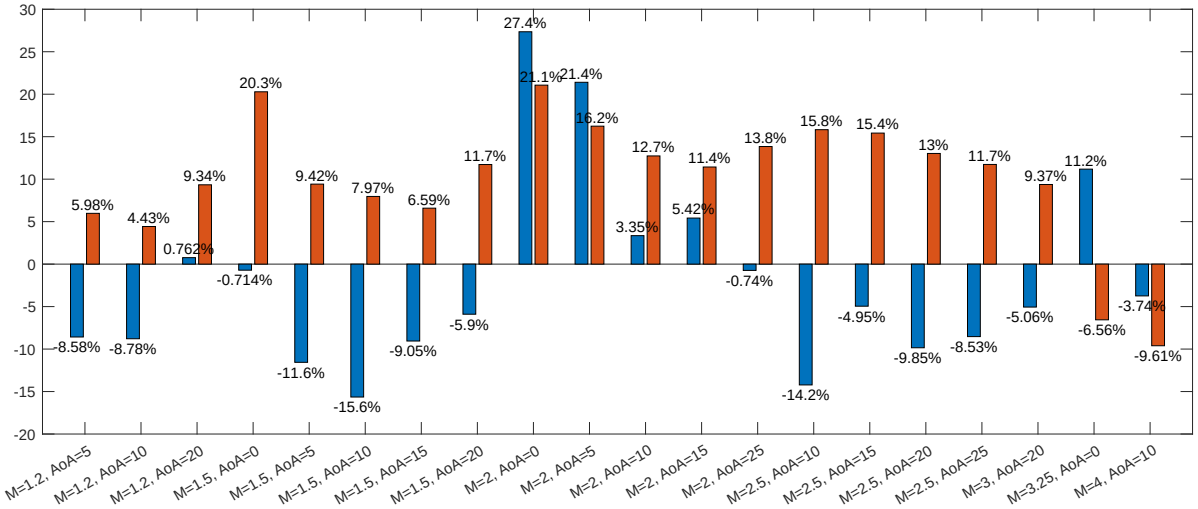
Figure C.16: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **lift coefficient**. 30 expensive points used for co-kriging; Run#15 from Table 6.9.



(a) Subsonic regime

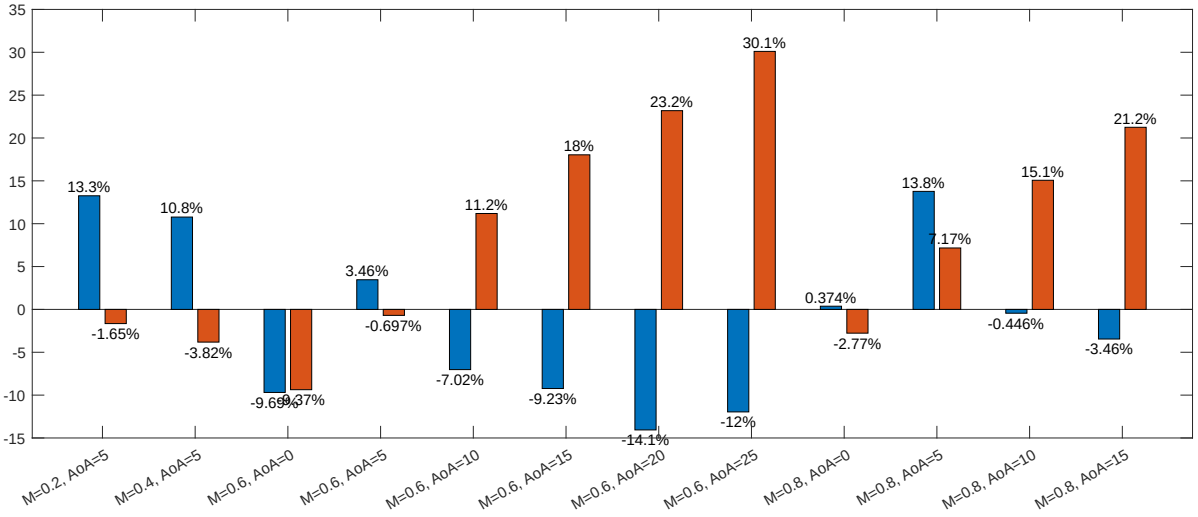


(b) Transonic regime

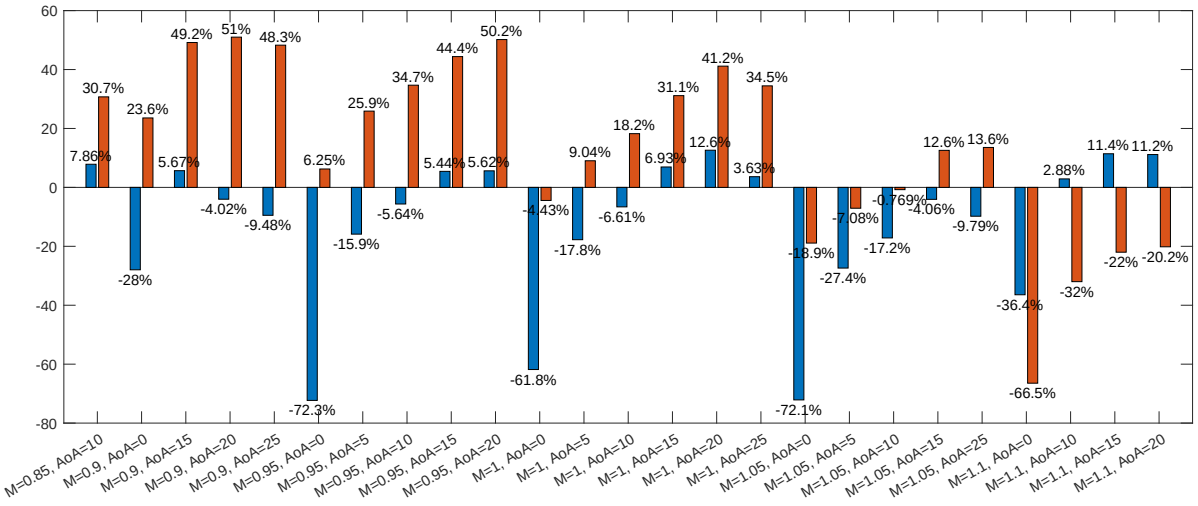


(c) Supersonic regime

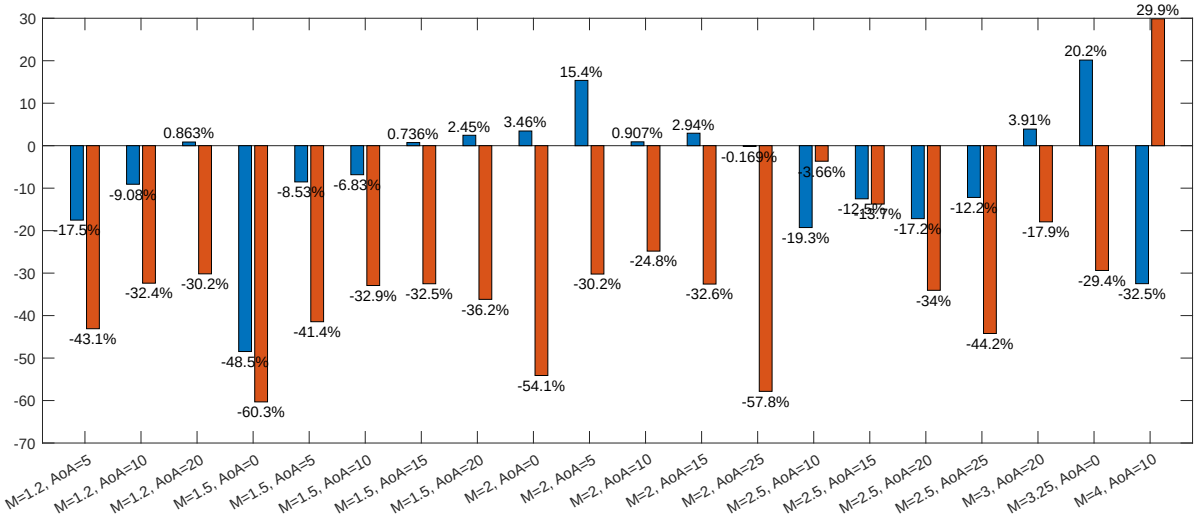
Figure C.17: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **drag coefficient**. 30 expensive points used for co-kriging; Run#15 from Table 6.9.



(a) Subsonic regime



(b) Transonic regime



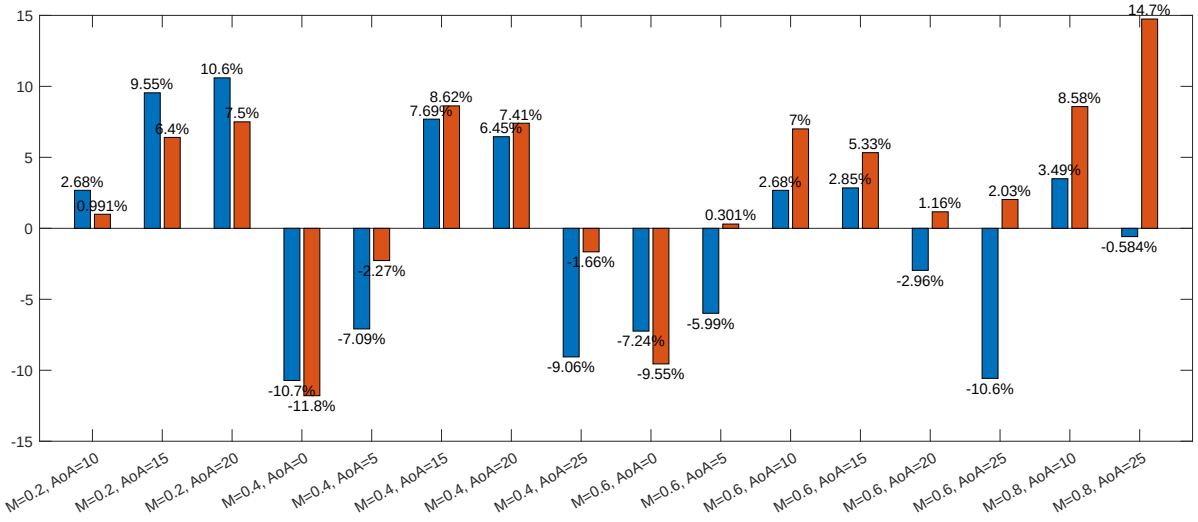
(c) Supersonic regime

Figure C.18: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **pitching moment coefficient**. 30 expensive points used for co-kriging; Run#15 from Table 6.9.

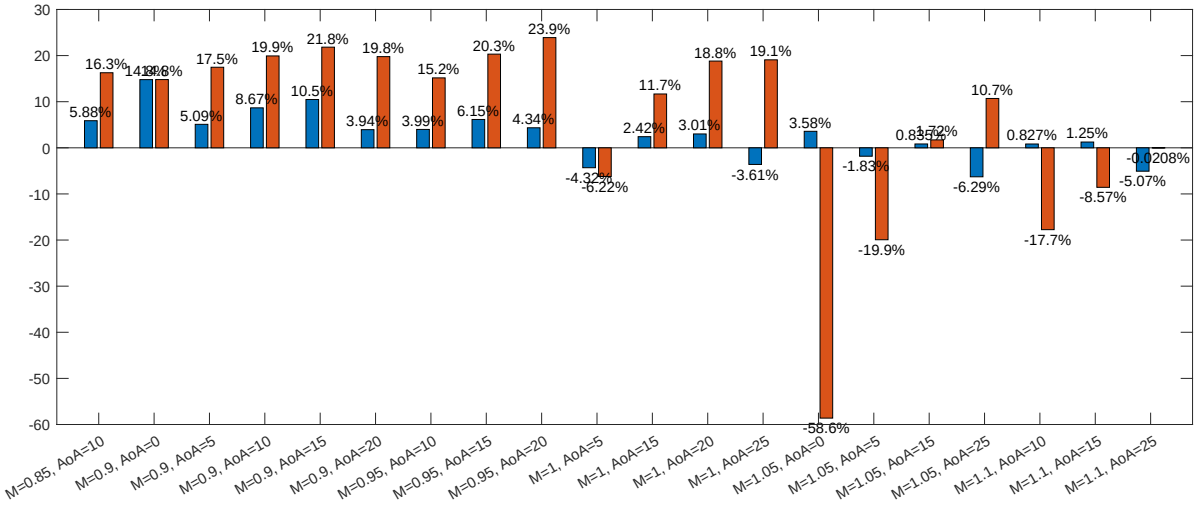
C.4. Case 4: 40 points

Table C.4: STDs of the thirty runs with 4 delimiting and 36 random points used for interpolation.

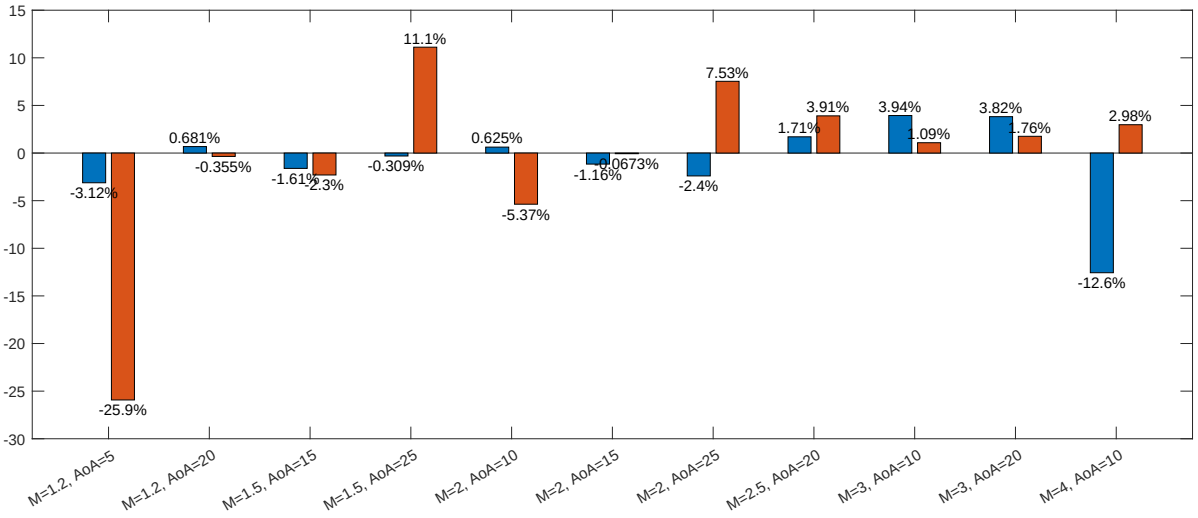
	Subsonic			Transonic			Supersonic		
Run	σ_{C_L}	σ_{C_D}	σ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}	σ_{C_L}	σ_{C_D}	σ_{C_M}
1	5.79	11.17	6.61	1.62	2.50	3.25	16.78	6.68	11.45
2	4.88	9.35	10.50	3.60	4.85	5.70	4.68	6.37	4.03
3	9.12	4.94	6.09	2.44	4.65	4.84	19.49	8.17	10.45
4	12.04	6.31	9.78	7.37	15.18	6.42	10.64	6.03	11.57
5	9.18	13.32	7.62	3.69	7.02	4.91	4.87	5.45	11.28
6	10.46	6.24	8.22	4.47	4.58	4.12	4.75	3.55	16.49
7	9.34	11.45	8.97	4.70	4.20	5.66	12.67	4.88	9.99
8	6.40	8.62	7.23	8.30	9.17	6.84	5.07	6.77	12.46
9	5.26	7.69	6.78	3.12	3.33	8.18	2.87	2.97	5.96
10	5.27	7.82	5.04	3.32	3.54	5.52	13.17	4.94	28.33
11	6.18	9.54	7.61	2.82	4.22	3.34	3.55	3.53	5.11
12	5.81	10.82	5.35	5.14	8.19	6.25	28.80	9.54	9.77
13	4.07	14.10	3.49	3.17	3.85	5.18	15.39	6.83	28.19
14	4.61	8.16	4.54	2.25	3.71	4.43	10.94	3.05	22.93
15	6.02	15.44	12.52	3.08	4.00	6.89	9.51	7.54	15.49
16	7.96	7.08	8.21	5.66	3.96	4.89	4.06	3.07	5.40
17	7.01	10.92	4.72	2.60	6.37	6.51	14.82	7.03	14.04
18	5.13	11.00	7.66	3.68	5.74	4.58	9.14	4.44	13.76
19	7.26	8.41	4.93	5.31	4.85	7.66	4.49	3.85	4.04
20	5.67	10.40	8.94	4.69	7.06	5.39	8.15	5.07	15.21
21	7.97	9.56	8.58	6.77	7.45	7.25	5.74	4.18	6.76
22	4.12	9.30	5.01	3.04	3.13	3.47	8.20	5.03	10.03
23	4.02	8.36	5.38	3.81	4.80	5.12	8.21	6.05	8.76
24	5.42	9.70	7.01	7.65	10.71	7.84	4.56	9.30	5.42
25	4.91	7.46	4.39	5.65	9.41	5.36	4.75	5.02	13.42
26	6.91	9.06	5.66	4.41	7.31	5.72	3.97	7.26	6.26
27	6.78	22.36	6.91	10.09	13.94	9.71	24.54	5.28	17.83
28	5.52	13.30	7.48	3.36	6.63	4.35	25.08	6.82	23.90
29	5.88	11.69	6.93	8.99	4.14	7.49	15.11	6.39	33.77
30	5.53	14.28	11.18	3.56	7.20	5.84	13.32	5.95	9.02
Mean	6.48	10.26	7.11	4.61	6.19	5.76	10.58	5.70	13.04
Kriging	6.66	7.34	15.59	25.87	18.60	31.01	29.21	14.28	22.20



(a) Subsonic regime

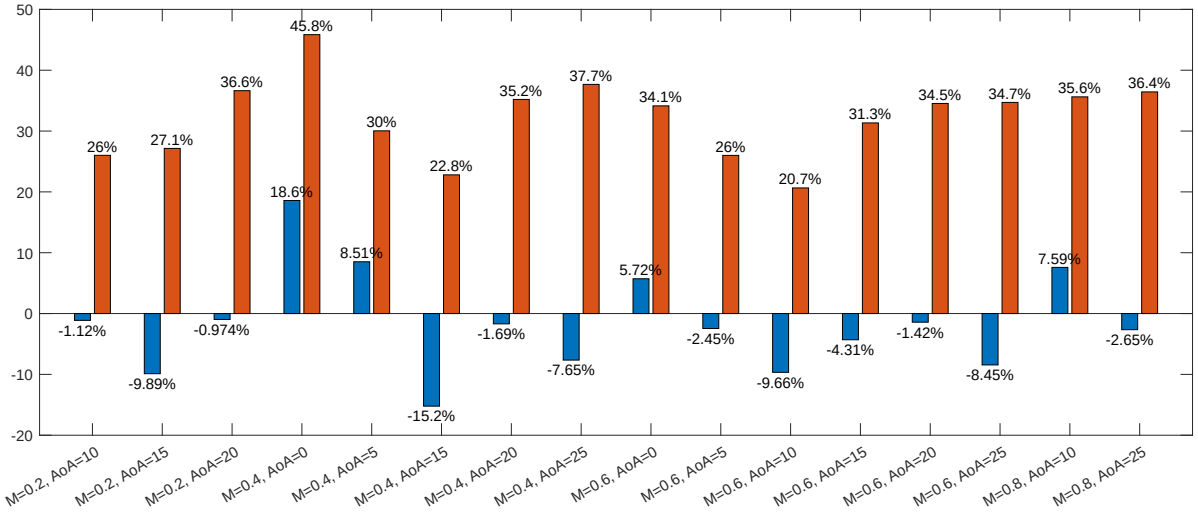


(b) Transonic regime

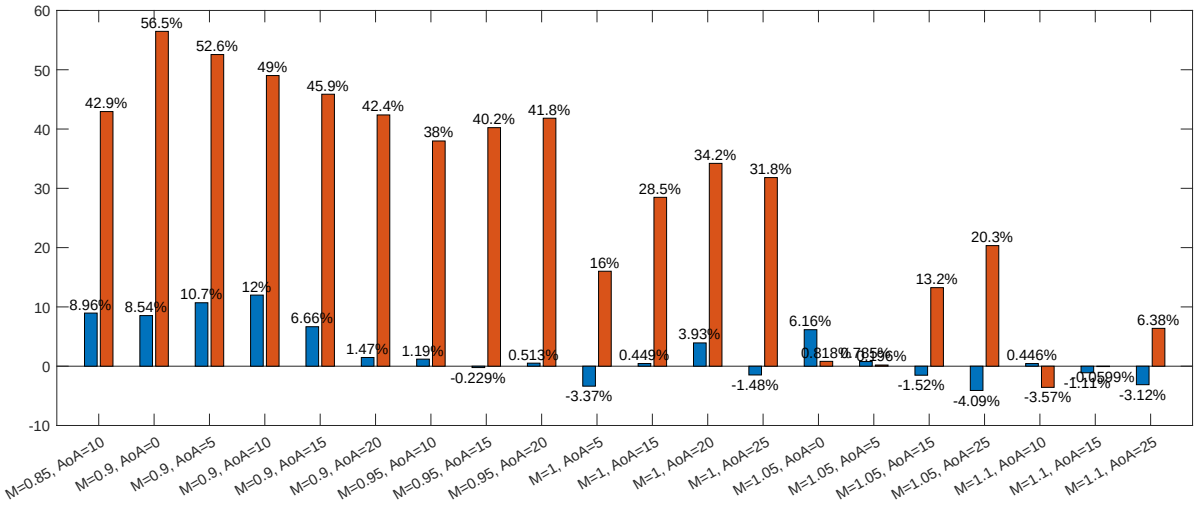


(c) Supersonic regime

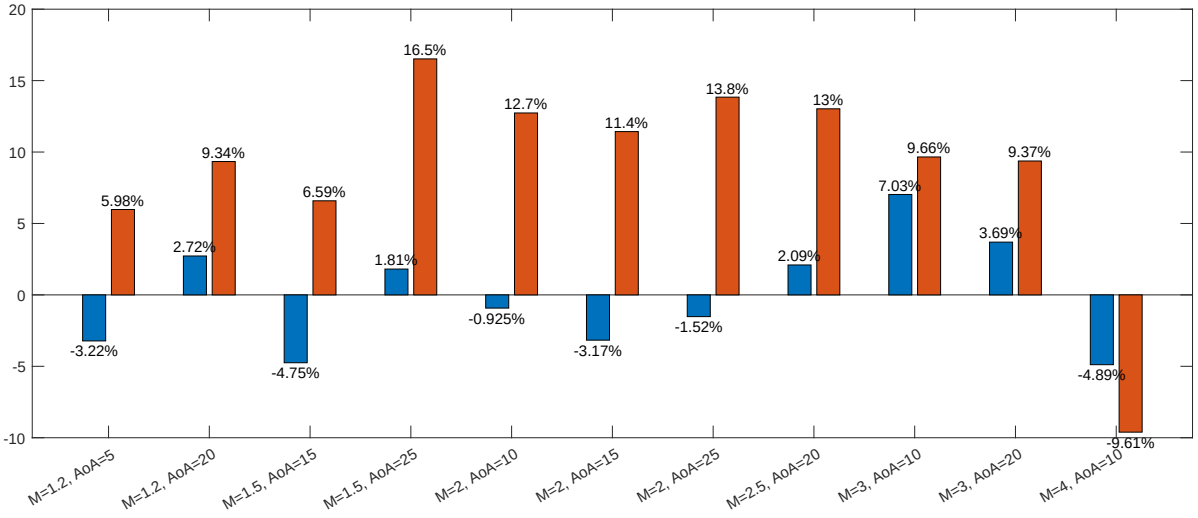
Figure C.19: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **lift coefficient**. 40 expensive points used for co-kriging; Run#19 from Table 6.10.



(a) Subsonic regime

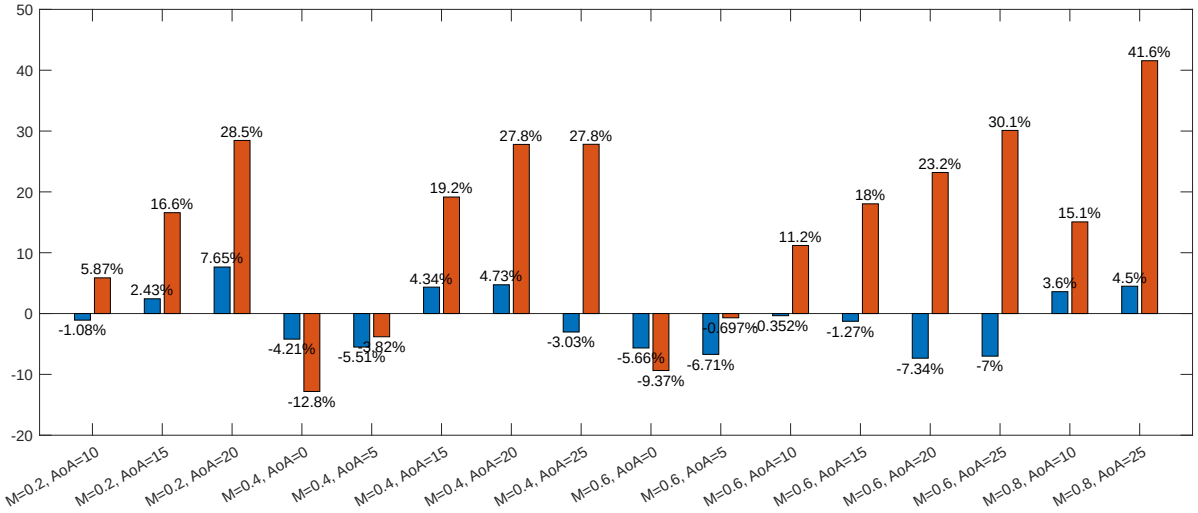


(b) Transonic regime

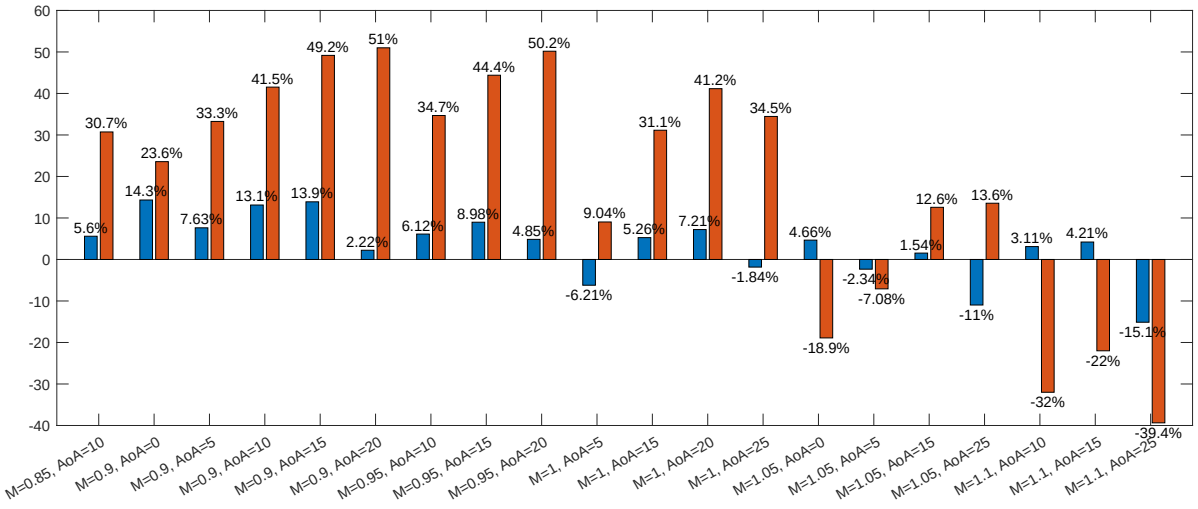


(c) Supersonic regime

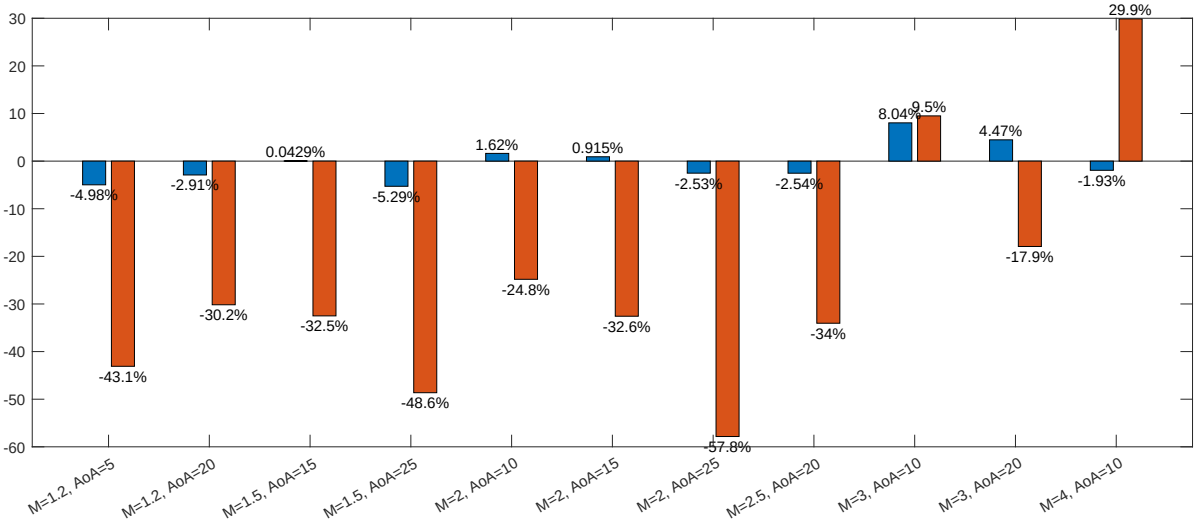
Figure C.20: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **drag coefficient**. 40 expensive points used for co-kriging; Run#19 from Table 6.10.



(a) Subsonic regime

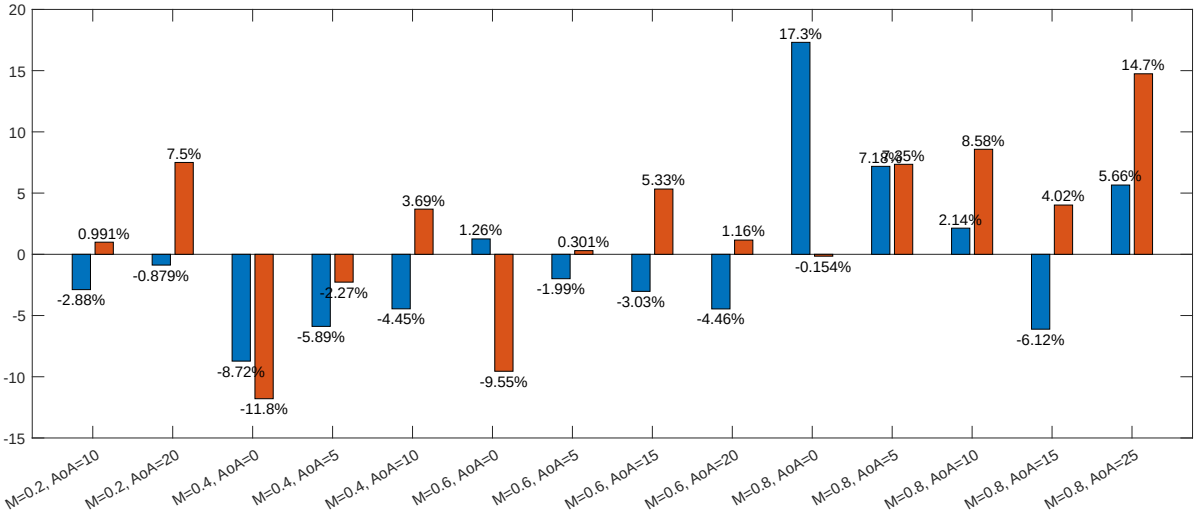


(b) Transonic regime

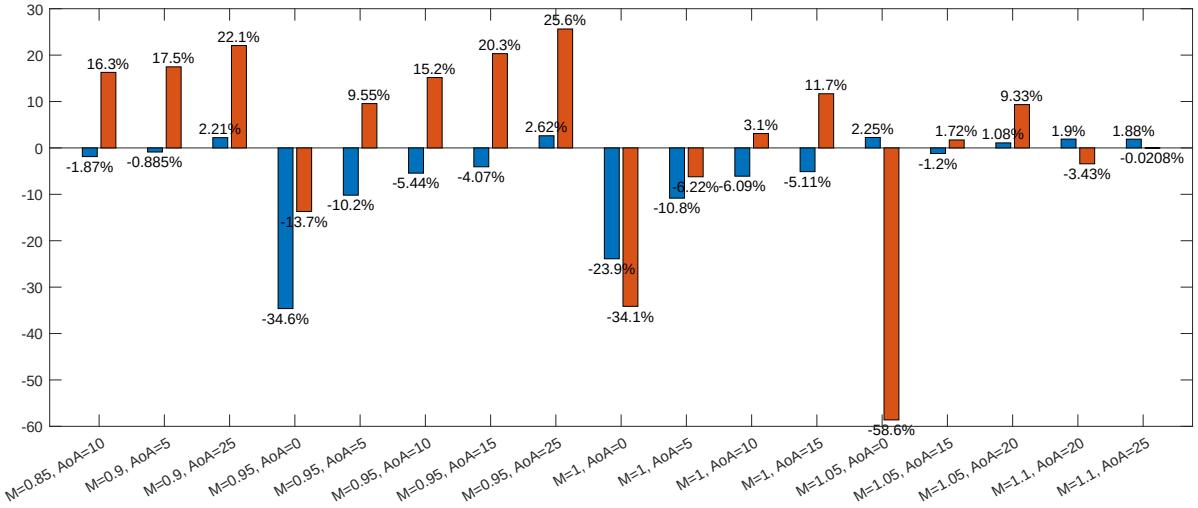


(c) Supersonic regime

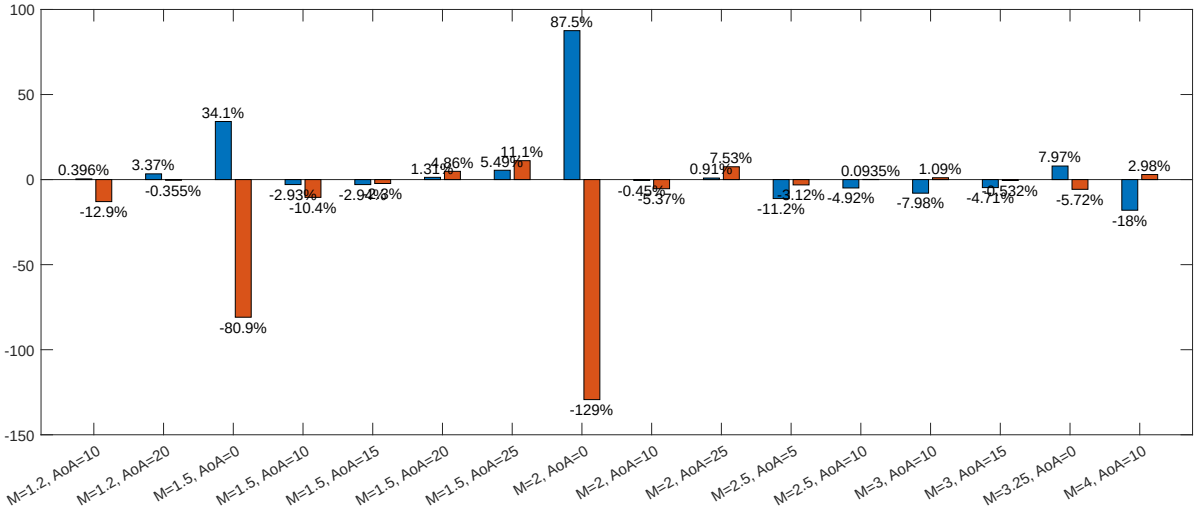
Figure C.21: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **pitching moment coefficient**. 40 expensive points used for co-kriging; Run#19 from Table 6.10.



(a) Subsonic regime

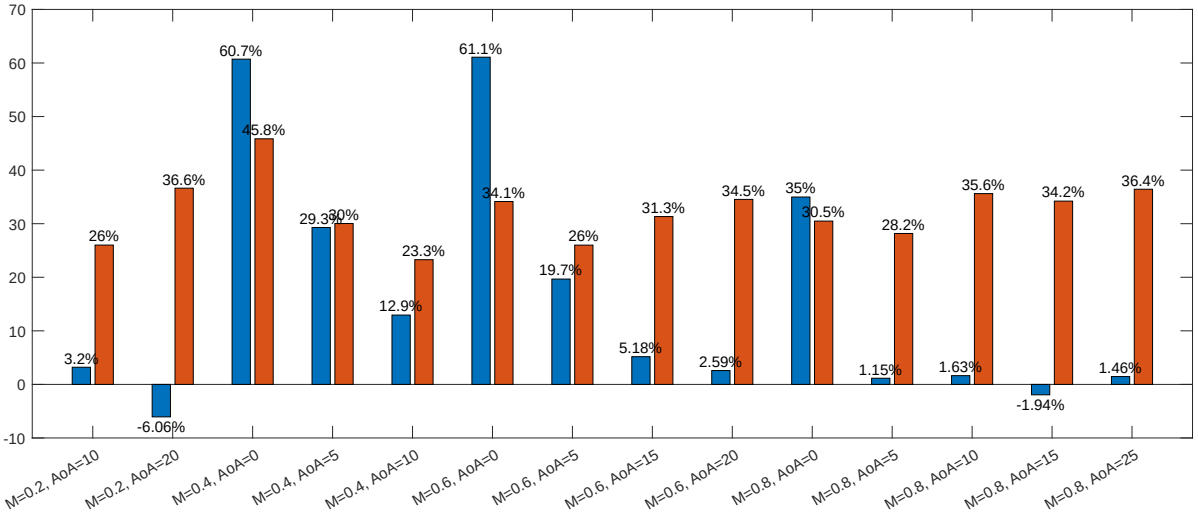


(b) Transonic regime

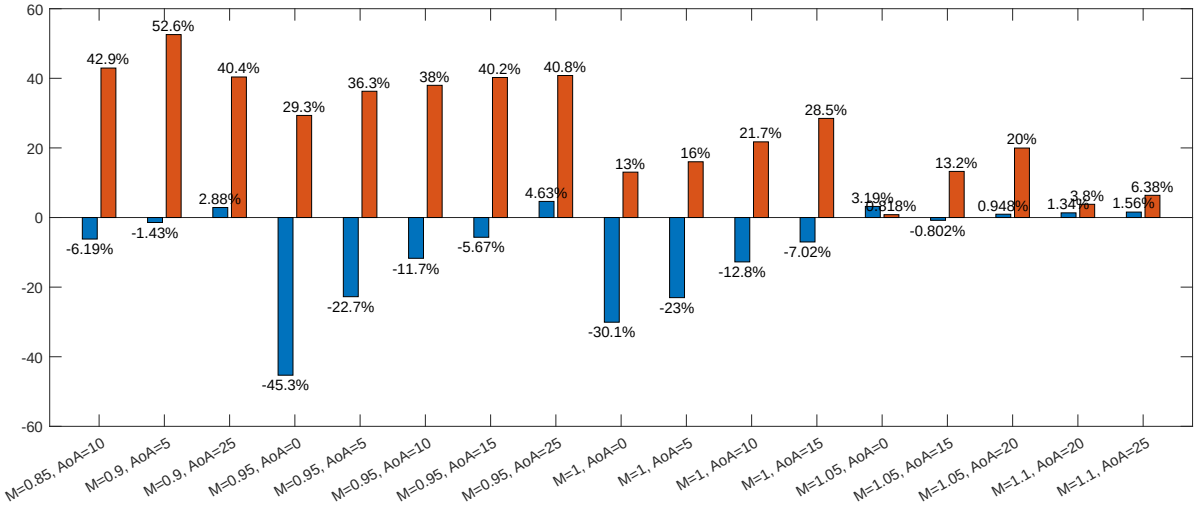


(c) Supersonic regime

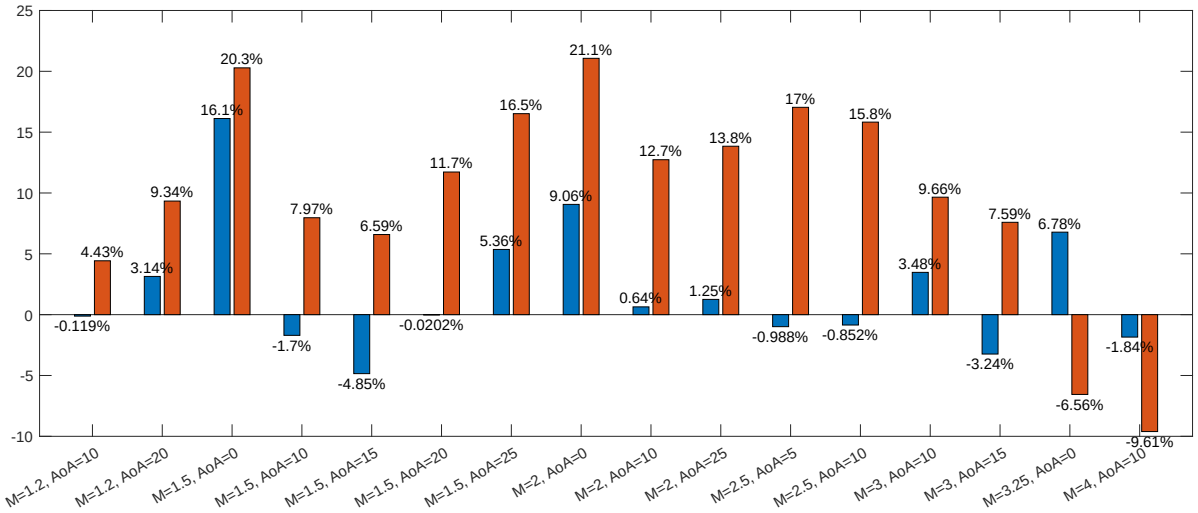
Figure C.22: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **lift coefficient**. 40 expensive points used for co-kriging; Run#27 from Table 6.10.



(a) Subsonic regime

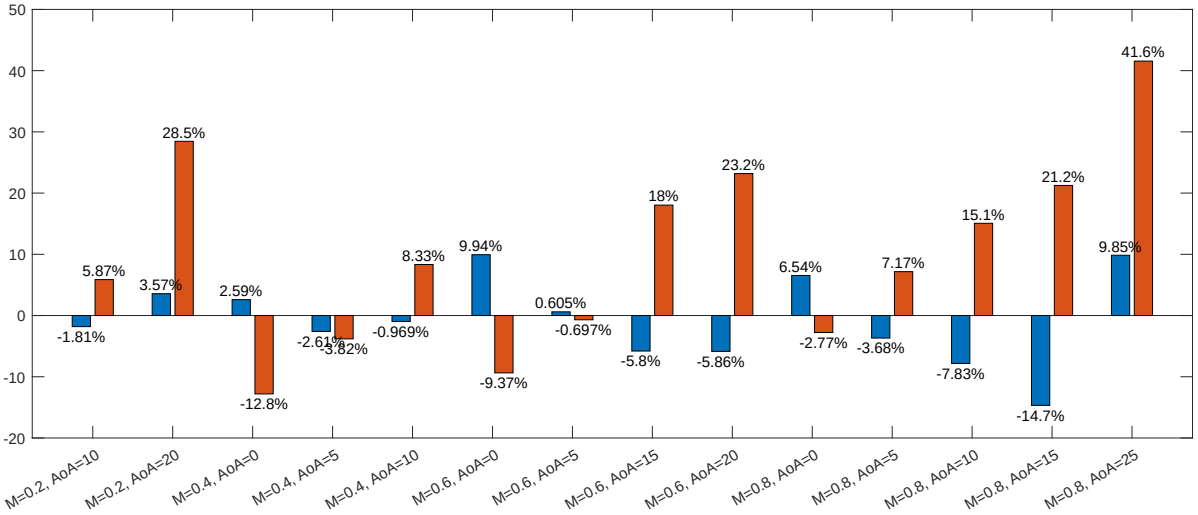


(b) Transonic regime

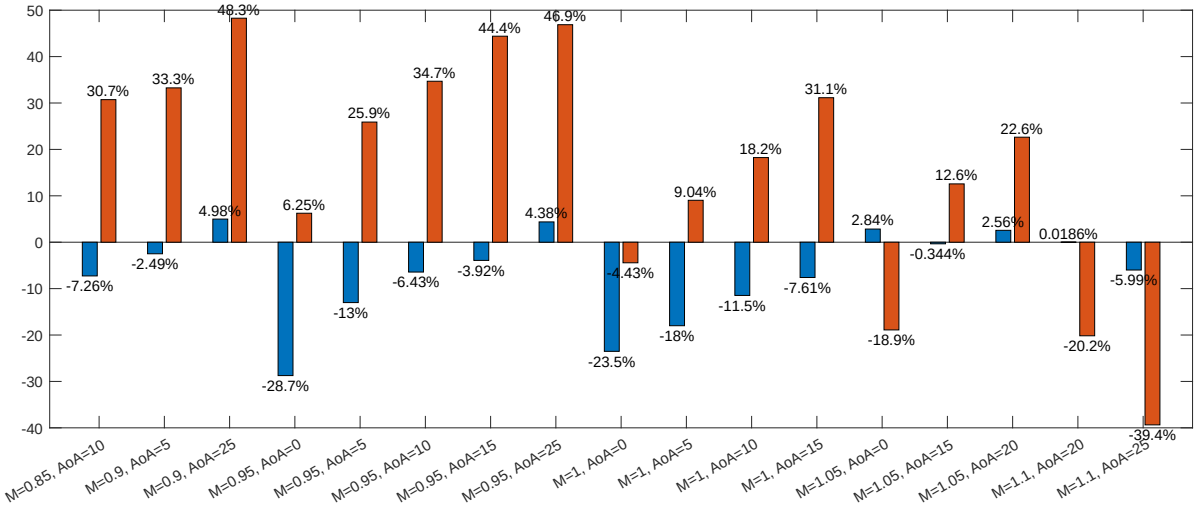


(c) Supersonic regime

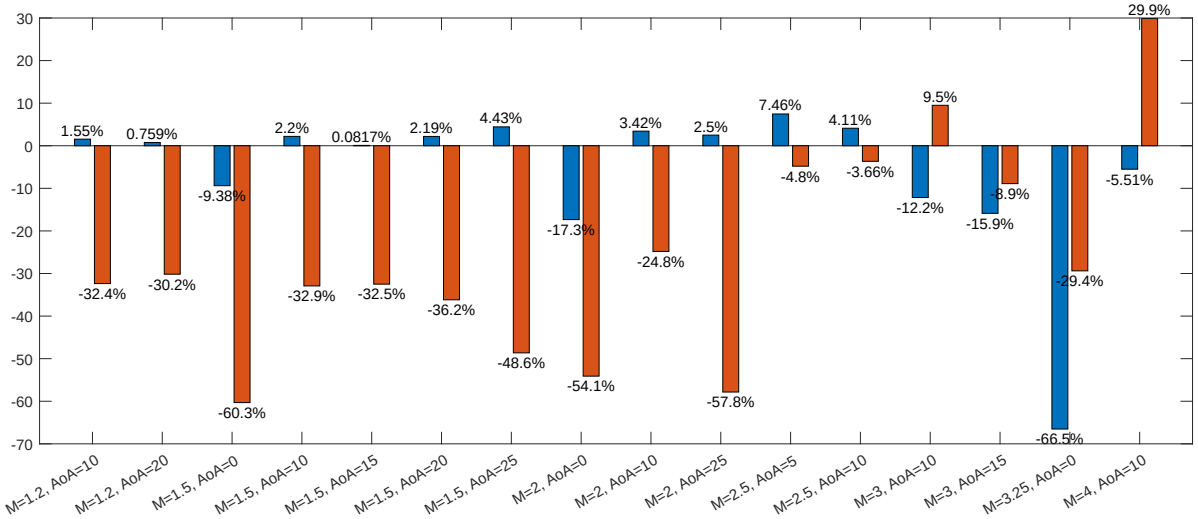
Figure C.23: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **drag coefficient**. 40 expensive points used for co-kriging; Run#27 from Table 6.10.



(a) Subsonic regime



(b) Transonic regime



(c) Supersonic regime

Figure C.24: Comparison of relative errors of the co-kriging (**blue**) and kriging (**red**) surrogate models of **pitching moment coefficient**. 40 expensive points used for co-kriging; Run#27 from Table 6.10.